# Structural Equation Models: An Open Textbook
# Edition 0.10

Jerry Brunner

Department of Statistical Sciences, University of Toronto
http://www.utstat.toronto.edu/brunner

January 3, 2023

# Contents

# Preface to Edition 0.10

## This book is free and open source

From the perspective of the student, possibly the most important thing about this textbook is that you don't have to pay for it. You can read it either online or in hard copy, and there are no restrictions on copying or printing. You may give a copy to anyone you wish; you may even sell it without paying royalties. The point is not so much that the book is free, but that *you* are free.

The plan for publishing this book is deliberately modeled on open source software. The source code is LaTeX (along with some modifiable graphics files in the OpenOffice drawing format), and the compiled binary is a PDF or DjVu file. Everthing is available at

http://www.utstat.toronto.edu/~brunner/openSEM.



**Reconstructed data sets** Structural equation modelling is a craft that is difficult to learn without having realistic data to analyze. But most good good data sets belong to somebody, and getting agreement to put them under copyleft protection can be a challenge. One solution is to make the data up, using a combination of random number generation and manual editing. Such a data set could be called *constructed*. I have done this in a few cases, and it can be quite tedious to make the sample statistics seem reasonable.

Another solution is to base the data upon the results of published studies. When I do this, I try to never use the original raw data set, even if I can get my hands on it. Instead, I start with a set of statistics derived directly or indirectly from the published source, and then simulate data that yield roughly (but not exactly) the same values of the statistics. I freely round the simulated data, change the sample size, and even add variables that the investigators probably would have measured, given sufficient resources.

Finally, I modify the data in any other way I can think of to make the example more instructive. I call such a data set *reconstructed*. I am not a lawyer, but it seems to me that (especially when the simulation is done using open source and copy-left protected software such as `R`) the specific raw data values generated in this manner can be protected under the GNU Free Documentation License. Another advantage is that the analysis of a reconstructed data set cannot necessarily be taken as a criticism of the way the data were originally treated, and it is easy to deny that conclusions based on a reconstructed data set have any clear scientific meaning. The purpose, of course, is to prepare the student to do statistical analyses that *do* have scientific meaning.

All the statistical analyses described in this book are based on constructed or reconstructed data sets. Appendix C contains a listing of the data sets used in examples and homework problems. A Zipped archive will also be available. As of this writing, it is not available yet.

**Software** Moderate familiarity with the R statistical computing environment [47] is assumed. Calculations on numerical data will use the `lavaan` (*la*tent *va*riable *an*alysis) package described by Rossel [48]. In this text, computing also extends to symbolic calculations that would ordinarily be done with paper and pencil. Symbolic calculations in this area (primarily, calculation of covariance matrices) are important for understanding particular models, but they are largely mechanical can get very tedious. The open source symbolic math program `SageMath` [53] is used extensively for pushing symbols around, starting with Chapter 1. Familiarity with the software is not assumed. An introduction is provided in Appendix B.

## This book is for Statistics students

This textbook is designed for third and fourth year undergraduate students in Statistics and Mathematics. It assumes the usual calculus-based second year sequence in Probability and Statistics and a basic course in linear algebra. Familiarity with linear regression is very helpful. Appendix A contains reference material and exercises that remind students of the necessary concepts. Some additional background material, especially on vector-valued random variables and the multivariate normal distribution, is needed but cannot be assumed. It is also covered in Appendix A.

This text is also appropriate (possibly as a supplemental text) for Masters level graduate students in Statistics. Since requests for structural equation models come up from time to time in consulting situations, it may also be useful to professional statisticians who need a quick introduction to the topic, in language they can understand.

But the main audience is undergraduate. For this reason, comments that are likely to be of interest to more advanced readers are often relegated to footnotes. These can be safely skipped by students who are primarily interested in learning the main ideas and getting a good mark.

# Message to the Instructor

It is common for textbook authors to claim that they decided to write a book because they could not locate an appropriate text, and I find myself in exactly this situation. Many introductions to structural equation models are available, but most of the ones I have seen are written for graduate students and researchers in the social sciences. Compared to most Statistics undergraduates, this audience has a very large English vocabulary and virtually no background in mathematical statistics. What works for them does not work as well for my students. So I have tried to write mainstream Statistics textboook on a topic that is somewhat out of the statistical mainstream.

Why bother? The main reason is that in addition to being standard statistical practice, ignoring measurement error is a disaster – and structural equation modeling is the simplest way I know to start addressing the problem. It helps that a well-prepared undergraduate is just a step away from having the necessary tools.

From a pedagogical viewpoint, structural equation modeling has another advantage. While the usual statistical methods we teach are like analytical devices purchased off the shelf, structural equation modeling methods are more like a kit which one can use to make a semi-customized analytical device. So, they help bridge the gap between applications of Statistics and genuine Applied Statistics. In particular, they force students to think at the interface between subject matter and technical statistical issues. And perhaps this is where the intellectual value of our discipline is most dense. Well, I *said* this was a message to the instructor.

# More details

The covariance review really is review and little else. Scalar covariance calculations are important throughout the course.

The maximum likelihood section has some warmup problems that are really review, but it also has some useful material on numerical maximum likelihood that students may find unfamiliar – for example the connection between the Hessian and the Fisher information. The main purpose is to build intuition about what can happen in more complicated situations.

# Chapter -1

# Overview

Structural equation models may be viewed as extensions of multiple regression. They generalize multiple regression in three main ways. First, there is usually more than one equation. Second, a response variable in one equation can be an explanatory variable in another equation. Third, structural equation models can include latent variables.

**Multiple equations** Structural equation models are usually based upon more than one regression-like equation. Having more than one equation is not really unique; multivariate regression already does that. But structural equation models are more flexible than the usual multivariate linear model.

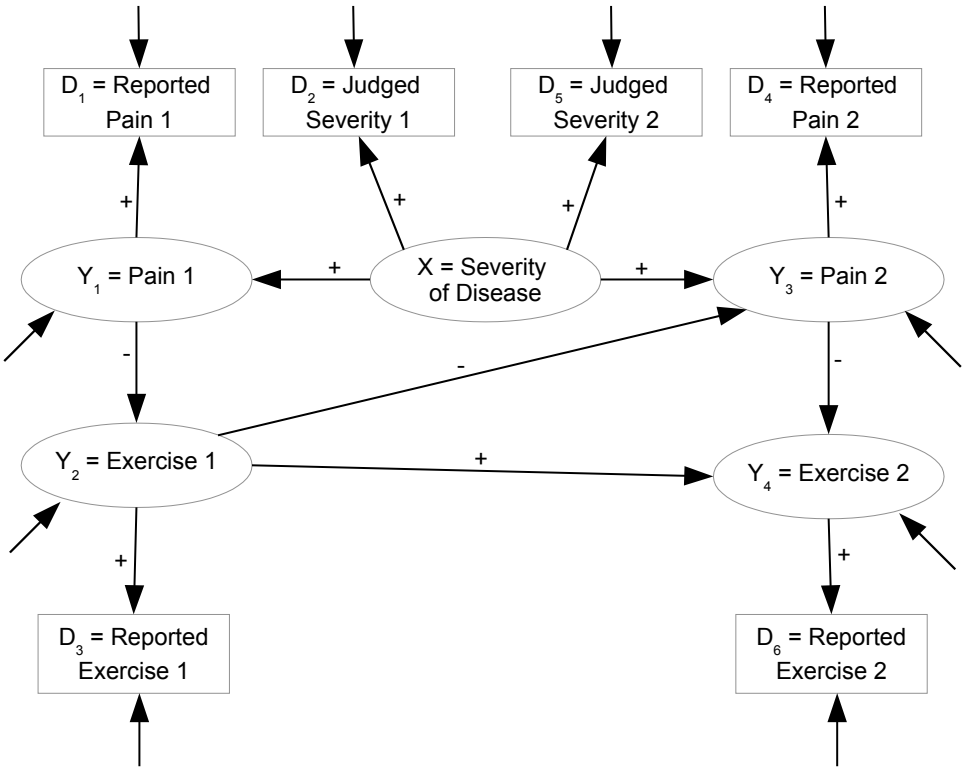
**Variables can be both explanatory and response** This is an attractive feature. Consider a political science study in which favourable information about a political party contributes to a favourable impression among potential voters at time one. But people often seek out information that supports their viewpoints, so that a favourable impression at time one contributes to exposure to favourable information at time two, which in turn contriutes to a favourable opinion at time two. Thus, opinion at time two is both a response variable and a response variable. Structural equation models are also capable of representing the back-and-forth nature of supply and demand in Economics. There are many other examples.

**Latent variables** To a degree that is often not acknowledged, the data you can see and record are not what you really are interested in. A *latent variable* is a random variable whose values cannot be directly observed – for example, true family income last year. Contrast this with an *observable variable* – for example, reported family income last year. Usually, interest is in relationships between latent variables, but the data set by definition includes only observable variables. Structural equation models may include latent as well as observable random variables, along with the connections between them. This capability (combined with relative simplicity) is their biggest advantage. It allows the statistican to admit that measurement error exists, and to incorporate it directly into the statistical model.

There are some ways that structural equation models are different from ordinary linear regression. These include random (rather than fixed) explanatory variable values, a bit of specialized vocabulary, and some modest changes in notation. Also, while structural equation models are definitely statistical models, they are also simple *scientific* models of the phenomena being investigated.

This last point is best conveyed by an example. Consider a study of arthritis patients, in which joint pain and exercise are assessed at more than one time point. Figure 1 is a path diagram that represents a structural equation model for the data.

Figure 1: Arthritis Pain




The notation is standard. Latent variables are in ovals, while observable variables are in boxes. Error terms seem to come from nowhere; in many path diagrams they are not shown at all. There is real modeling here, and plenty of theoretical input is required. The plusses and minuses on some of the straight arrows are a bit non-standard. The represent hypothesized positive and negative relationships.

As the directional arrows suggest, structural equation models are usually interpreted as *causal* models. That is, they are models of influence. $A \rightarrow B$ means $A$ has an influence on $B$. In the path diagram, reported pain at time one is influenced by true pain at time one. There are other influences on reported pain, including the patient's reading level, interpretation of the questions on the questionnaire, self-presentation strategy, and so on. These unmeasured influences are represented by an error term. The error term is not

shown explicitly, but the arrow that seems to come from nowhere is coming from the error term.

Structural equation models are causal models [9], but the data are usually observational. That is, explanatory variables are typically not manipulated or randomly assigned by the investigator, as they would be in an experimental study. Instead, they are simply measured or assessed. This brings up the classic *correlation versus causation* issue. The point is often summarized by saying "correlation does not imply causation." That is, if the variables $X$ and $Y$ are related to one another (not independent), it could be that $X$ is influencing $Y$, or that $Y$ is influencing $X$, or that a third variable, $Z$ is influencing both $X$ and $Y$. In the absence of other information, it's wise to be cautious. Practitioners of applied regression are often warned not to claim that the $x$ variables influence the $y$ variable unless the values of the $x$ variables are randomly assigned.

Structural equation modeling adresses the correlation-causation problem by constructing a model that is simultaneously a statistical model and a substantive theory of the data. In this way, a great many details are decided on theoretical or at least commonsense grounds, and the rest are left to statistical estimation and testing. In Figure 1, for example, it is obvious that the arrows should run from Time One to Time Two and not the other way around. Notice that in the path diagram, the severity of the disease is essentially the same at Time One and Time Two. This is a theoretical assertion based on the nature of the disease and the length of time involved. All such assertions are open to healthy debate.

Not everybody likes this. Some statisticians, particularly students, don't feel comfortable with theory construction in a scientific discipline outside their field. This is less a problem than it seems. While it's true that the ideal case is for the same person to be expert in both the statistics and the subject matter (as in econometrics), frequently the statistician works together with a scientist who wants to apply structural equation models to his or her data. Most scientists get the idea of path diagrams very fast, and the collaboration can go smoothly.

It must be admitted, though, that some scientists are uncomfortable with making theoretical commitments and incorporating them into the statistical analysis. To them, data analysis is where evidence is assessed and weighed. Building theory into the statistical model seems biased, like putting a finger on the scale[1]. One response to this is that the generic statistical models in common use also carry assumptions with theoretical implications. Getting involved in the assembly of the statistical model just serves to make the black box less mysterious, and that can only be a good thing.

Path diagrams correspond to systems of regression-like equations. Here are the equa-

[1]There is a distinctly Bayesian feel to the way structural equation models depend on prior information. The objection of bias is also raised against Bayesian methods, for exactly the same reason. It is possible to do structural equation modeling in a fully Bayesian way, but the approach in this book is strictly frequentist.

tions corresponding to Figure 1. Independently for $i = 1, \ldots, n$,

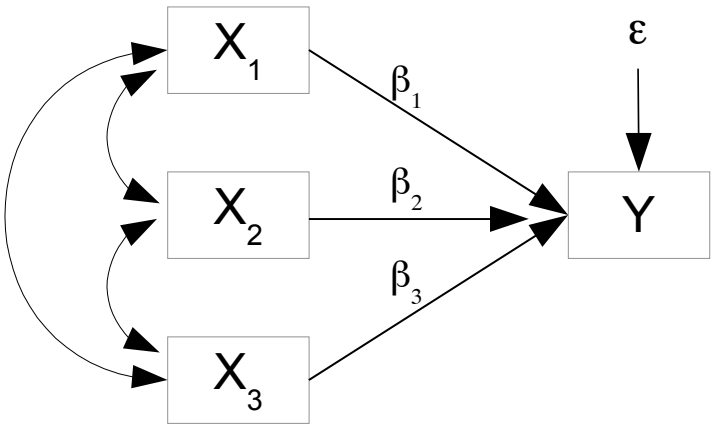
$$\begin{aligned}
Y_{i,1} &= \beta_{0,1} + \beta_1 X_i + \epsilon_{i,1} \\
Y_{i,2} &= \beta_{0,2} + \beta_2 Y_{i,1} + \epsilon_{i,2} \\
Y_{i,3} &= \beta_{0,3} + \beta_3 X_i + \beta_4 Y_{i,2} + \epsilon_{i,3} \\
Y_{i,4} &= \beta_{0,4} + \beta_5 Y_{i,2} + \beta_6 Y_{i,3} + \epsilon_{i,4} \\
D_{i,1} &= \lambda_{0,1} + \lambda_1 Y_{i,1} + e_{i,1} \\
D_{i,2} &= \lambda_{0,2} + \lambda_2 X_i + e_{i,2} \\
D_{i,3} &= \lambda_{0,3} + \lambda_3 Y_{i,2} + e_{i,3} \\
D_{i,4} &= \lambda_{0,4} + \lambda_4 Y_{i,3} + e_{i,4} \\
D_{i,5} &= \lambda_{0,5} + \lambda_2 X_i + e_{i,5} \\
D_{i,6} &= \lambda_{0,6} + \lambda_5 Y_{i,4} + e_{i,6}
\end{aligned} \tag{1}$$

Every variable that appears on the left side of an equation has at least one arrow pointing to it, and the arrows pointing to the left-side variable originate from the variables on the right side.

The path diagram contains some additional information. Note that there are no direct connections between the error terms, or between the error terms and underlying disease severity $X_i$. This represents an assertion that these quantities are independent. If they were not independent, covariances would be represented by curved, double-headed arrows. An example is given in Figure 2. Notice that all the variables are observable, the error term is shown this time, and the straight arrows from $x$ to $y$ are labelled with the regression coefficients. This is all within the range of standard notation for path diagrams.

Figure 2: Regression with Observable variables

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \beta_3 X_{i,3} + \epsilon_i$$



Returning to the example of Figure 1, the model as given is still not fully specified. It is common to assume that everything is normal. In most software, the default method of estimation is numerical maximum likelihood based on a multivariate normal distribution for the observable data. There is considerable robustness to this assumption so it does little harm. With the normal assumption and letting the expected values of the error terms equal zero, we have 12 more model parameters, including the expected value and variance of $X_i$, underlying disease severity. As usual in Statistics, the objective is to estimate and draw inferences about the unknown parameters, with the goal of casting light on the phenomena that gave rise to the data.

**Parameter identifiability** It is an uncomfortable truth that for the model given here, maximum likelihood estimation will fail. The maximum of the likelihood function would not be unique. Instead, infinitely many sets of parameter values would yield the same maximum. Geometrically, the likelihood function would have a flat surface at the top.

Here's why. Let $\boldsymbol{\theta}$ denote the vetor of parameters we are trying to estimate. $\boldsymbol{\theta}$ contains all the Greek-letter parameters in the model equations (1), plus ten error variances, and also the expected value and variance of $X_i$. Thus, $\boldsymbol{\theta}$ has 34 elements.

Assume that the model is completely correct, and that disease severity and all the error terms are normally distributed. This means the vector of six observable variables (there are six boxes in the path diagram) have a joint distribution that is multivariate normal — independently for $i = 1, \ldots, n$, of course. All one can ever learn from a data set is the joint distribution of the observable data, and a multivariate normal is completely characterized by its mean vector and variance covariance matrix. Thus, with increasing sample sizes, all you can ever know is a closer and closer approximations of the six expected values (call them $\mu_1, \ldots, \mu_6$) and the 21 unique values of the $6 \times 6$ covariance matrix (call them $\sigma_{ij}, i \leq j$). Suppose you knew the $\mu_j$ and $\sigma_{ij}$ values exactly (conceptually letting $n \to \infty$, if that is an idea that helps). Would this tell you the values of all the model parameters in $\boldsymbol{\theta}$?

The $\mu_j$ and $\sigma_{ij}$ are definitely functions of $\boldsymbol{\theta}$, and those functions may be obtained by direct calculation of the expected values, variances and covariances using the model equations (1). This yields 27 equations. To ask whether the 34 model parameters can be recovered from the $\mu_j$ and $\sigma_{ij}$ is to ask whether it's possible to solve the 27 equations for 34 unknowns. As one might expect, the answer is no. More precisely, it is impossible to solve uniquely. There are infinitely many solutions, so that infinitely many sets of parameter values are equally compatible with any data set. This corresponds to the flat place on the top of the likelihood surface.

In general, model parameters are said to be *identifiable* if their values can be recovered from the probability distribution of the observable data. In structural equation modeling, it is very easy to come up with reasonable models whose parameters are not identifiable — like the arthritis pain and exercise example we are considering. When parameters are not identifiable, estimation and inference can be a challenge, though in some cases the problems can be overcome. In structural equation modeling, almost everything is connected to the the issue of parameter identifiability, and on a technical level, this is

what sets structural equation modeling apart from other applied statistical methods based on large-sample maximum likelihood. One of the most important tools in the structural equation modeling toolkit is a set of rules (based on theorems about solving systems of equations) that often allow the identifiability of a model to be determined based on visual inspection of a path diagram, without any calculations. The story begins with an important special case: regression with measurement error.

# Chapter 0

# Regression with measurement error

## Introduction

This chapter seeks to accomplish two things. First, it is a self-contained introduction to linear regression with measurement error in the explanatory variables, suitable as a supplement to an ordinary regression course. Second, it is an introduction to the study of structural equation models. Without confronting the general formulation at first, the student will learn why structural equation models are important and see what can be done with them. Some of the ideas and definitions are repeated later in the book, so that the theoretical treatment of structural equation modeling does not depend much on this chapter. On the other hand, the material in this chapter will be used throughout the rest of the book as a source of examples. It should not be skipped by most readers.

## 0.1 Covariance and Relationship

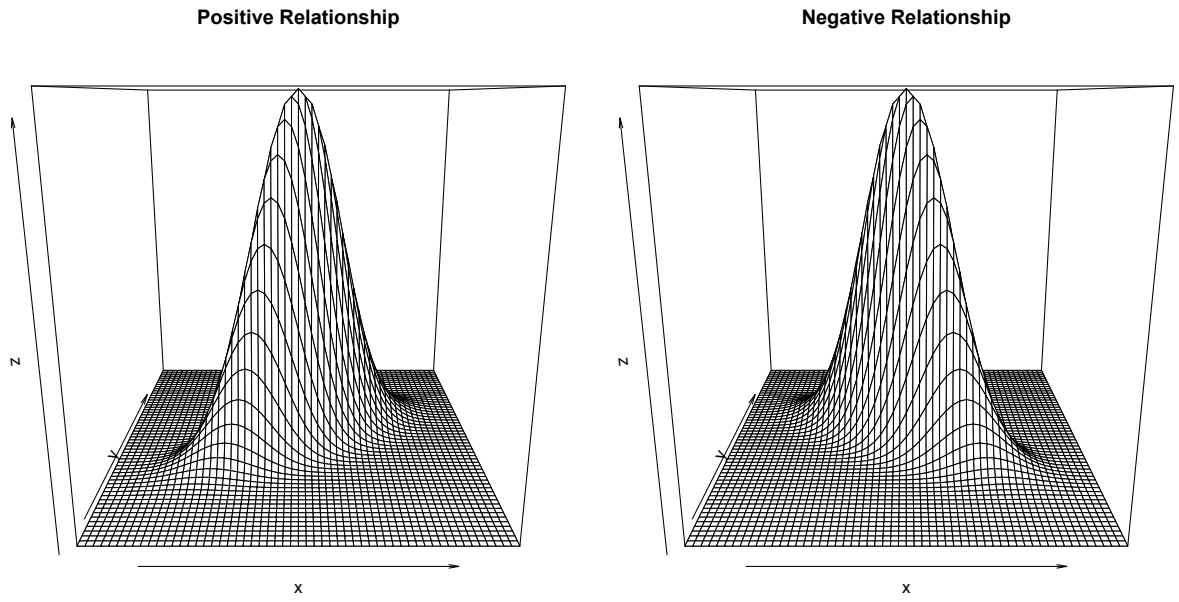
Most of the models we will consider are linear in the explanatory variables as well as the regression parameters, and so relationships between explanatory variables and response variables are represented by covariances. To clarify this fundamental point, first note that saying two random variables are "related" really just means that they are not independent. A non-zero covariance implies lack of independence, and therefore it implies a relationship of some kind between the variables. Furthermore, if the random variables in question are normally distributed (a common and very useful model), zero covariance is exactly the same thing as independence.

More generally, consider two random variables $X$ and $Y$ whose joint distribution might not be bivariate normal. Suppose there is a tendency for higher values of $X$ to go with higher values of $Y$, and for lower values of $X$ to go with lower values of $Y$. This idea of a "positive" relationship is pictured in the left panel of Figure 1. Since the probability of an $(x, y)$ pair is roughly proportional to the height of the surface, a large sample of points will be most dense where the surface is highest[1]. On a scatterplot, the best-fitting line

[1]Presumably this is why it's called a probability *density* function.

relating $X$ to $Y$ will have a positive slope. The right panel of Figure 1 shows a negative relationship. There, the best-fitting line will have a negative slope.

Figure 1: Relationship between $X$ and $Y$




The word "covariance" suggests that it is a measure of how $X$ and $Y$ vary together. To see that positive relationships yield positive covariances and negative relationships yield negative covariances, look at Figure 2.

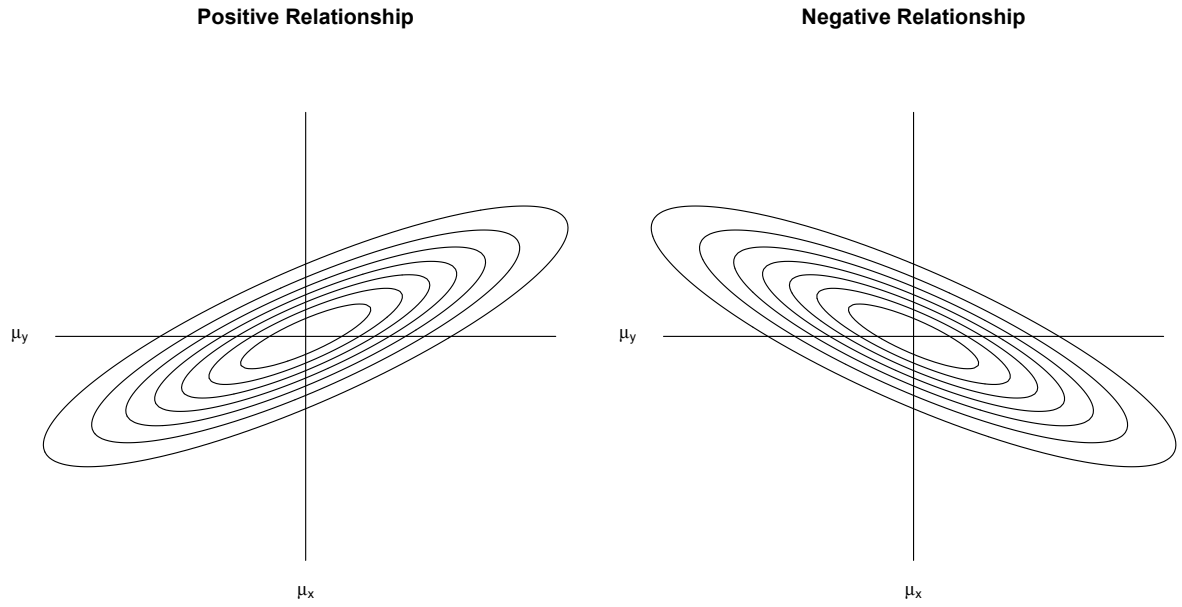
Figure 2 shows contour plots of the densities in Figure 1. Imagine you are looking down at a density from directly above, and that the density has been cut into slices that are parallel with the $x, y$ plane. The ellipses are the cut marks. The outer ellipse is lowest, the next one in is a bit higher, and so on. All the points on an ellipse (contour) are at the same height. It's like a topographic map of a mountainous region, except that the contours on maps are not so regular.

The definition of covariance is

$$Cov(X,Y) = E\left\{(X-\mu_x)(Y-\mu_y)\right\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\mu_x)(y-\mu_y)f(x,y)\,dx\,dy$$

In the left panel of Figure 2, more of the probability is in the upper right and lower left, and that is where $(x-\mu_x)(y-\mu_y)$ is positive. The positive volume in these regions is greater than the negative volume in the upper left and lower right, so that the integral is positive. In the right-hand panel the opposite situation occurs, and the covariance is negative. The pictures are just of one example, but the rule is general. Positive covariances reflect positive relationships and negative covariances reflect negative relationships.

Figure 2: Contour Plots




In the study of linear structural equation models, one frequently needs to calculate covariances and matrices of covariances. Covariances of linear combinations are frequently required. The following rules are so useful that they are repeated from Sections A.1 and A.3 of Appendix A.

Let $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ be scalar random variables, and define the linear combinations $L_1$ and $L_2$ by

$$\begin{aligned} L_1 &= a_1 X_1 + \cdots + a_{n_1} X_{n_1} = \sum_{i=1}^{n_1} a_i X_i, \text{ and} \\ L_2 &= b_1 Y_1 + \cdots + b_{n_2} Y_{n_2} = \sum_{i=1}^{n_2} b_i Y_i, \end{aligned}$$

where the $a_j$ and $b_j$ are constants. Then

$$cov(L_1, L_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} a_i b_j Cov(X_i, Y_j). \tag{1}$$

In the matrix version, let $\mathbf{x}_1, \ldots, \mathbf{x}_{n_1}$ and $\mathbf{y}_1, \ldots, \mathbf{y}_{n_2}$ be random vectors, and define

the linear combinations $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_2$ by

$$\begin{aligned} \boldsymbol{\ell}_1 &= \mathbf{A}_1\mathbf{x}_1 + \cdots + \mathbf{A}_{n_1}\mathbf{x}_{n_1} = \sum_{i=1}^{n_1} \mathbf{A}_i\mathbf{x}_i, \text{ and} \\ \boldsymbol{\ell}_2 &= \mathbf{B}_1\mathbf{y}_1 + \cdots + \mathbf{B}_{n_2}\mathbf{y}_{n_2} = \sum_{i=1}^{n_2} \mathbf{B}_i\mathbf{y}_i, \end{aligned}$$

where the $\mathbf{A}_j$ and $\mathbf{B}_j$ are matrices of constants. Then

$$cov(\boldsymbol{\ell}_1, \boldsymbol{\ell}_2) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \mathbf{A}_i \, cov(\mathbf{x}_i, \mathbf{y}_j) \, \mathbf{B}_j^\top. \tag{2}$$

Both these results say that to calculate the covariance of two linear combinations, just take the covariance of each term in the first linear combination with each term in the second linear combination, and add them up. When simplifying the results of calculations, it can be helpful to recall that $Cov(X, X) = Var(X)$ and $cov(\mathbf{x}, \mathbf{x}) = cov(\mathbf{x})$.

## 0.2 Regression: Conditional or Unconditional?

Consider the usual version of univariate multiple regression. For $i = 1, \ldots, n$,

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_{p-1} x_{i,p-1} + \epsilon_i,$$

where $\epsilon_1, \ldots \epsilon_n$ are independent random variables with expected value zero and common variance $\sigma^2$, and $x_{i,1}, \ldots x_{i,p-1}$ are fixed constants. For testing and constructing confidence intervals, $\epsilon_1, \ldots \epsilon_n$ are typically assumed normal.

Alternatively, the regression model may be written in matrix notation, as follows:

$$\mathbf{y} \;=\; \mathbf{X}\boldsymbol{\beta} \;+\; \boldsymbol{\epsilon}, \tag{3}$$

where $\mathbf{X}$ is an $n \times p$ matrix of known constants, $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown constants, and $\boldsymbol{\epsilon}$ is multivariate normal with mean zero and covariance matrix $\sigma^2\mathbf{I}_n$; the variance $\sigma^2 > 0$ is a constant.

Now please take a step back and think about this model, rather than just accepting it without question. In particular, think about why the $x$ variables should be constants. It's true that if they are constants then all the calculations are easier, but in the typical application of regression to observational[2] data, it makes more sense to view the explanatory variables as random variables rather than constants. Why? Because if you took repeated

[2] *Observational* data are just observed, rather than being controlled by the investigator. For example, the average number of minutes per day spent outside could be recorded for a sample of dogs. In contrast to observational data are *experimental* data, in which the values of the variable in question are controlled by the investigator. In an experimental study, dogs could be randomly assigned to several different values of the variable "time outside."

samples from the same population, the values of the explanatory variables would be different each time. Even for an experimental study with random assignment of cases (say dogs) to experimental conditions, suppose that the data are recorded in the order they were collected. Again, with high probability the values of the explanatory variables would be different each time.

So, why are the $x$ variables a set of constants in the formal model? One response is that the regression model is a conditional one, and all the conclusions hold conditionally upon the values of the explanatory variables. This is technically correct, but consider the reaction of a zoologist using multiple regression, assuming he or she really appreciated the point. She would be horrified at the idea that the conclusions of the study would be limited to this particular configuration of explanatory variable values. No! The sample was taken from a population, and the conclusions should apply to that population, not to the subset of the population with these particular values of the explanatory variables.

At this point you might be a bit puzzled and perhaps uneasy, realizing that you have accepted something uncritically from authorities you trusted, even though it seems to be full of holes. In fact, everything is okay this time. It is perfectly all right to apply a conditional regression model, even when the predictors are clearly random. But it's not so very obvious why it's all right, or in what sense it's all right. This section will give the missing details. These are skipped in every regression textbook I have seen; I'm not sure why.

**Unbiased Estimation** Under the standard conditional regression model (3), it is straightforward to show that the vector of least-squares regression coefficients $\widehat{\boldsymbol{\beta}}$ is unbiased for $\boldsymbol{\beta}$ (both of these are $p \times 1$ vectors). This means that it's unbiased *conditionally* upon $\mathbf{X} = \mathbf{x}$. In symbols,

$$E\{\widehat{\boldsymbol{\beta}}|\mathbf{X} = \mathbf{x}\} = \boldsymbol{\beta}.$$

This applies to every fixed $\mathbf{x}$ matrix with linearly independent columns, a condition that is necessary and sufficient for $\widehat{\boldsymbol{\beta}}$ to exist. Assume that the joint probability distribution of the random matrix $\mathbf{X}$ assigns zero probability to matrices with linearly dependent columns (which is the case for continuous distributions). Using the double expectation formula $E\{Y\} = E\{E\{Y|X\}\}$,

$$E\{\widehat{\boldsymbol{\beta}}\} = E\{E\{\widehat{\boldsymbol{\beta}}|\mathbf{X}\}\} = E\{\boldsymbol{\beta}\} = \boldsymbol{\beta},$$

since the expected value of a constant is just the constant. This means that *estimates of the regression coefficients from the conditional model are still unbiased, even when the explanatory variables are random.*

The following calculation might make the double expectation a bit clearer. The outer expected value is with respect to the joint probability distribution of the explanatory variable values – all $n$ vectors of them; think of the $n \times p$ matrix $\mathbf{X}$. To avoid unfamiliar

notation, suppose they are all continuous, with joint density $f(\mathbf{x})$. Then

$$\begin{aligned} E\{\widehat{\boldsymbol{\beta}}\} &= E\{E\{\widehat{\boldsymbol{\beta}}|\mathbf{X}\}\} \\ &= \int\cdots\int E\{\widehat{\boldsymbol{\beta}}|\mathbf{X}=\mathbf{x}\}\, f(\mathbf{x})\, d\mathbf{x} \\ &= \int\cdots\int \boldsymbol{\beta}\, f(\mathbf{x})\, d\mathbf{x} \\ &= \boldsymbol{\beta}\int\cdots\int f(\mathbf{x})\, d\mathbf{x} \\ &= \boldsymbol{\beta}\cdot 1 = \boldsymbol{\beta}. \end{aligned}$$

**Consistent Estimation** It will now be shown that when the explanatory variable values are random, $\widehat{\boldsymbol{\beta}}_n \xrightarrow{p} \boldsymbol{\beta}$; see Section A.5 in Appendix A for a brief discussion of consistency. The demonstation is a bit lengthy, but the details are shown because one of the intermediate results will be very useful later. The argument begins by establishing an alternative formula for the ordinary least-squares estimates. The explanatory variable values are fixed for now, but in the end, the formula will be applied to random $X$ values.

A regression model can be "centered" by subtracting sample means from the values of the explanatory variables. Geometrically, what this does is to shift the cloud of points in a high-dimensional scatterplot left or right along each $x$ axis – or equivalently, to adopt a shifted set of co-ordinate axes. Clearly, this will not affect the tilt (slopes) of the best-fitting hyperplane, but it will affect the intercept. Writing the regression model in scalar form and then centering, ...

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} + \epsilon_i \\ &= \beta_0 + \beta_1\overline{x}_1 + \cdots + \beta_p\overline{x}_p \\ &\quad +\beta_1(x_{i,1} - \overline{x}_1) + \cdots + \beta_p(x_{i,p} - \overline{x}_p) + \epsilon_i \\ &= \alpha_0 + \alpha_1(x_{i,1} - \overline{x}_1) + \cdots + \alpha_p(x_{i,p} - \overline{x}_p) + \epsilon_i, \end{aligned}$$

where the $\alpha$ parameters are the regression coefficients of the centered model. We have $\alpha_0 = \beta_0 + \beta_1\overline{x}_1 + \cdots + \beta_k\overline{x}_p$, and $\alpha_j = \beta_j$ for $j = 1, \ldots, p$. This re-parameterization is one-to-one. Since the least-squares and maximum likelihood estimates coincide for multiple regression with normal errors, the invariance principle of maximum likelihood estimation (See Section A.6.3 in Appendix A) says that $\widehat{\alpha}_j = \widehat{\beta}_j$ for $j = 1, \ldots, p$. That is, centering does not change the estimated slopes. In addition, the MLE of the intercept for the centered model is $\widehat{\alpha}_0 = \widehat{\beta}_0 + \widehat{\beta}_1\overline{x}_1 + \cdots + \widehat{\beta}_p\overline{x}_p$. Invoking once again the identity of least-squares and maximum likelihood estimates for this case, we see that the $\widehat{\alpha}_j$ quantities are also the least-squares estimates for the centered model[3].

[3]This argument uses the invariance principle for maximum likelihood estimation, but that's not really necessary. There is also an invariance principle for least-squares, which is proved in exactly the same way as the invariance principle for maximum likelihood.

For any regression model with an intercept, the sum of residuals is zero. Thus,

$$\begin{aligned}\bar{y} &= \frac{1}{n}\sum_{i=1}^{n}\widehat{y}_i \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{\beta}_0 + \widehat{\beta}_1 x_{i,1} + \cdots + \widehat{\beta}_p x_{i,p}\right) \\ &= \widehat{\beta}_0 + \widehat{\beta}_1\overline{x}_1 + \cdots + \widehat{\beta}_p\overline{x}_p \\ &= \widehat{\alpha}_0\end{aligned}$$

That is, the least-squares estimate of the intercept is $\bar{y}$ for any centered regression model, regardless of the data.

We already know how to calculate the $\widehat{\beta}_j$, but we are working toward another formula for them. Suppose we start with the centered model

$$y_i = \alpha_0 + \beta_1(x_{i,1} - \overline{x}_1) + \cdots + \beta_p(x_{i,p} - \overline{x}_p) + \epsilon_i.$$

Because this is a centered model, we know that $\widehat{\alpha}_0 = \overline{y}$. To find the $\widehat{\beta}_j$, first substitute $\widehat{\alpha}_0 = \overline{y}$ and then minimize

$$Q(\boldsymbol{\beta}) = \sum_{i=1}^{n}\left(y_i - \overline{y}\ - \beta_1(x_{i,1} - \overline{x}_1) - \cdots - \beta_p(x_{i,p} - \overline{x}_p)\ \right)^2$$

over all $\boldsymbol{\beta}$. This is the same as centering $y$ as well as $x$, and then fitting a regression through the origin. The usual formula $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ applies. We just need to remember that the columns of the $n \times p$ matrix $\mathbf{X}$ are centered, and so is the $n \times 1$ vector $\mathbf{y}$. For $p = 3$, the $\mathbf{X}$ matrix looks like this:

$$\begin{pmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & x_{13} - \bar{x}_3 \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & x_{23} - \bar{x}_3 \\ x_{31} - \bar{x}_1 & x_{32} - \bar{x}_2 & x_{33} - \bar{x}_3 \\ \vdots & \vdots & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & x_{n3} - \bar{x}_3 \end{pmatrix}.$$

The $\mathbf{X}^\top\mathbf{X}$ matrix, the so-called the "sums of squares and cross products" matrix, is

$$\begin{aligned}\mathbf{X}^\top\mathbf{X} &= \begin{pmatrix} x_{11} - \bar{x}_1 & x_{21} - \bar{x}_1 & x_{31} - \bar{x}_1 & \cdots & x_{n1} - \bar{x}_1 \\ x_{12} - \bar{x}_2 & x_{22} - \bar{x}_2 & x_{32} - \bar{x}_2 & \cdots & x_{n2} - \bar{x}_2 \\ x_{13} - \bar{x}_3 & x_{23} - \bar{x}_3 & x_{33} - \bar{x}_3 & \cdots & x_{n3} - \bar{x}_3 \end{pmatrix} \begin{pmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & x_{13} - \bar{x}_3 \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & x_{23} - \bar{x}_3 \\ x_{31} - \bar{x}_1 & x_{32} - \bar{x}_2 & x_{33} - \bar{x}_3 \\ \vdots & \vdots & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & x_{n3} - \bar{x}_3 \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^n (x_{i1} - \overline{x}_1)^2 & \sum_{i=1}^n (x_{i1} - \overline{x}_1)(x_{i2} - \overline{x}_2) & \sum_{i=1}^n (x_{i1} - \overline{x}_1)(x_{i3} - \overline{x}_3) \\ \sum_{i=1}^n (x_{i2} - \overline{x}_2)(x_{i1} - \overline{x}_1) & \sum_{i=1}^n (x_{i2} - \overline{x}_2)^2 & \sum_{i=1}^n (x_{i2} - \overline{x}_2)(x_{i3} - \overline{x}_3) \\ \sum_{i=1}^n (x_{i3} - \overline{x}_3)(x_{i1} - \overline{x}_1) & \sum_{i=1}^n (x_{i3} - \overline{x}_3)(x_{i2} - \overline{x}_2) & \sum_{i=1}^n (x_{i3} - \overline{x}_3)^2 \end{pmatrix}.\end{aligned}$$

It's clear that larger examples would follow this same pattern. The entries in the matrix look like sample variances and covariances, except that they are not divided by $n$. Dividing and multiplying by $n$, we have $\mathbf{X}^\top\mathbf{X} = n\widehat{\boldsymbol{\Sigma}}_x$, where $\widehat{\boldsymbol{\Sigma}}_x$ is the sample variance-covariance matrix of the explanatory variables.

Still looking at the $p = 3$ case for simplicity,

$$\begin{aligned}
\mathbf{X}^\top\mathbf{y} &= \begin{pmatrix} x_{11}-\bar{x}_1 & x_{21}-\bar{x}_1 & x_{31}-\bar{x}_1 & \cdots & x_{n1}-\bar{x}_1 \\ x_{12}-\bar{x}_2 & x_{22}-\bar{x}_2 & x_{32}-\bar{x}_2 & \cdots & x_{n2}-\bar{x}_2 \\ x_{13}-\bar{x}_3 & x_{23}-\bar{x}_3 & x_{33}-\bar{x}_3 & \cdots & x_{n3}-\bar{x}_3 \end{pmatrix} \begin{pmatrix} y_1-\bar{y} \\ y_2-\bar{y} \\ y_3-\bar{y} \\ \vdots \\ y_n-\bar{y} \end{pmatrix} \\
&= \begin{pmatrix} \sum_{i=1}^n (x_{i1}-\overline{x}_1)(y_i-\overline{y}) \\ \sum_{i=1}^n (x_{i1}-\overline{x}_1)(y_i-\overline{y}) \\ \sum_{i=1}^n (x_{i1}-\overline{x}_1)(y_i-\overline{y}) \end{pmatrix} \\
&= n \begin{pmatrix} \frac{1}{n}\sum_{i=1}^n (x_{i1}-\overline{x}_1)(y_i-\overline{y}) \\ \frac{1}{n}\sum_{i=1}^n (x_{i1}-\overline{x}_1)(y_i-\overline{y}) \\ \frac{1}{n}\sum_{i=1}^n (x_{i1}-\overline{x}_1)(y_i-\overline{y}) \end{pmatrix} \\
&= n\widehat{\boldsymbol{\Sigma}}_{xy},
\end{aligned}$$

where $\widehat{\boldsymbol{\Sigma}}_{xy}$ is the $k \times 1$ vector of sample covariances between the explanatory variables and the response variable.

Putting the pieces together, the least squares estimator of $\boldsymbol{\beta}$ is

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}_n &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \\
&= (n\widehat{\boldsymbol{\Sigma}}_x)^{-1} n\widehat{\boldsymbol{\Sigma}}_{xy} \\
&= \frac{1}{n}(\widehat{\boldsymbol{\Sigma}}_x)^{-1} n\widehat{\boldsymbol{\Sigma}}_{xy} \\
&= \widehat{\boldsymbol{\Sigma}}_x^{-1}\widehat{\boldsymbol{\Sigma}}_{xy}. \qquad (4)
\end{aligned}$$

Several comments are in order. First, recall that $\widehat{\boldsymbol{\beta}}_n$ is a vector of least-squares slopes only. It does not include the intercept. However, the intercept for a centered model is $\bar{y}$, and is easily computed. Second, because the slopes are the same for the centered model and the uncentered model, formula (4) applies equally to uncentered models. Third, in spite of the suggestive $\widehat{\boldsymbol{\Sigma}}$ notation, expression (4) is just a computational formula. It applies whether the explanatory variable values are random or fixed. Only when the variables are random do $\widehat{\boldsymbol{\Sigma}}_x$ and $\widehat{\boldsymbol{\Sigma}}_{xy}$ actually estimate variances and covariances.

When the explanatory variables are random, the Strong Law of Large Numbers and continuous mapping yield

$$\widehat{\boldsymbol{\beta}}_n \stackrel{a.s.}{\to} \boldsymbol{\Sigma}_x^{-1}\boldsymbol{\Sigma}_{xy}. \tag{5}$$

The only requirement for convergence is that $\boldsymbol{\Sigma}_x^{-1}$ exist, which is equivalent to $\boldsymbol{\Sigma}_x$ being positive definite.

The convergence (5) applies whether the regression model is correct or not. For this reason, it can be a valuable tool for studying *mis-specified* regression models — that is, models that are assumed, but are not actually correct. If you can calculate $\widehat{\boldsymbol{\Sigma}}_x$ and $\widehat{\boldsymbol{\Sigma}}_{xy}$ under the true model, you can determine where the estimated regression coefficients are going as the sample size increases. This will often indicate whether the mis-specification is likely to cause mistaken conclusions.

For the present, suppose that the usual uncentered regression model is correct. Independently for $i = 1, \ldots, n$, let

$$y_i = \beta_0 + \boldsymbol{\beta}^\top \mathbf{X}_i + \epsilon_i$$

where

$\beta_0$ (the intercept) is an unknown scalar constant.

$\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown slope parameters.

$\mathbf{x}_i$ is a $p \times 1$ random vector with expected value $\boldsymbol{\mu}$ and positive definite covariance matrix $\boldsymbol{\Sigma}_x$.

$\epsilon_i$ is a scalar random variable with $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2$.

$cov(\mathbf{x}_i, \epsilon_i) = \mathbf{0}$.

So,

$$\begin{aligned}
\boldsymbol{\Sigma}_{xy} &= cov(\mathbf{x}_i, y_i) \\
&= cov(\mathbf{x}_i, \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i + \epsilon_i) \\
&= cov(\mathbf{x}_i, \boldsymbol{\beta}^\top \mathbf{x}_i + \epsilon_i) \\
&= cov(\mathbf{x}_i, \boldsymbol{\beta}^\top \mathbf{x}_i) + cov(\mathbf{x}_i, \epsilon_i) \\
&= cov(\mathbf{x}_i, \mathbf{x}_i)\boldsymbol{\beta} + \mathbf{0} \\
&= \boldsymbol{\Sigma}_x\boldsymbol{\beta}.
\end{aligned}$$

Then by (5)

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}_n &\stackrel{a.s.}{\to} \boldsymbol{\Sigma}_x^{-1}\boldsymbol{\Sigma}_{xy} \\
&= \boldsymbol{\Sigma}_x^{-1}\boldsymbol{\Sigma}_x\boldsymbol{\beta} \\
&= \boldsymbol{\beta}.
\end{aligned}$$

Since almost sure convergence implies convergence in probability (see Section A.5 in Appendix A), we have $\widehat{\boldsymbol{\beta}}_n \stackrel{p}{\to} \boldsymbol{\beta}$. This is the standard definition of (weak) consistency. The

meaning is that as the sample size increases, the probability that the usual least-squares estimate $\widehat{\boldsymbol{\beta}}_n$ is arbitrarily close to $\boldsymbol{\beta}$ approaches one. This holds even though the explanatory variable values are random variables, and $\widehat{\boldsymbol{\beta}}_n$ was derived under the assumption that they are fixed constants.

**Size $\alpha$ Tests** Suppose Model (3) is conditionally correct, and we plan to use an $F$ test. Conditionally upon the $x$ values, the $F$ statistic has an $F$ distribution when the null hypothesis is true, but unconditionally it does not. Rather, its probability distribution is a *mixture* of $F$ distributions, with

$$Pr\{F \in A\} = \int \cdots \int Pr\{F \in A|\mathbf{X} = \mathbf{x}\} f(\mathbf{x})\, d\mathbf{x}.$$

If the null hypothesis is true and the set $A$ is the critical region for an exact size $\alpha$ $F$-test, then $Pr\{F \in A|\mathbf{X} = \mathbf{x}\} = \alpha$ for every fixed set of explanatory variable values $\mathbf{x}$. In that case,

$$\begin{aligned} Pr\{F \in A\} &= \int \cdots \int \alpha f(\mathbf{x})\, d\mathbf{x} \\ &= \alpha \int \cdots \int f(\mathbf{x})\, d\mathbf{x} \\ &= \alpha. \end{aligned} \tag{6}$$

Thus, the so-called $F$-test has the correct Type I error rate when the explanatory variables are random (assuming the model is conditionally correct), even though the test statistic does not have an $F$ distribution.

It might be suspected that if the explanatory variables are random and we assume they are fixed, the resulting estimators and tests might be of generally low quality, even though the estimators are unbiased and the tests have the right Type I error probability. Now we will see that given a fairly reasonable set of assumptions, this fear is unfounded.

Denoting the explanatory variable values by $\mathbf{X}$ and the response variable values by $\mathbf{Y}$, suppose the joint distribution of $\mathbf{X}$ and $\mathbf{Y}$ has the following structure. The distribution of $\mathbf{X}$ depends on a parameter vector $\boldsymbol{\theta}_1$. Conditionally on $\mathbf{X} = \mathbf{x}$, the distribution of $\mathbf{Y}$ depends on a parameter vector $\boldsymbol{\theta}_2$, and $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are *not functionally related*. For a standard regression model this means that the distribution of the explanatory variables does not depend upon the values of $\boldsymbol{\beta}$ or $\sigma^2$ in any way. This is surely not too hard to believe.

Please notice that the model just described is not at all limited to linear regression. It is very general, covering almost any conceivable regression-like method including logistic regression and other forms of non-linear regression, generalized linear models and the like.

Because likelihoods are just joint densities or probability mass functions viewed as functions of the parameter, the notation of Appendix A.6.8 may be stretched just a little bit to write the likelihood function for the unconditional model (with $\mathbf{X}$ random) in terms

of conditional densities as

$$\begin{aligned} L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \mathbf{x}, \mathbf{y}) &= f_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2}(\mathbf{x}, \mathbf{y}) \\ &= f_{\boldsymbol{\theta}_2}(\mathbf{y}|\mathbf{x})\, f_{\boldsymbol{\theta}_1}(\mathbf{x}) \\ &= L_2(\boldsymbol{\theta}_2, \mathbf{x}, \mathbf{y})\, L_1(\boldsymbol{\theta}_1, \mathbf{x}) \end{aligned} \tag{7}$$

Now, take the log and partially differentiate with respect to the elements of $\boldsymbol{\theta}_2$. The marginal likelihood $L_1(\boldsymbol{\theta}_1, \mathbf{x})$ disappears, and $\widehat{\boldsymbol{\theta}}_2$ is exactly what it would have been for a conditional model.

In this setting, likelihood ratio tests are also identical under conditional and unconditional models. Suppose the null hypothesis concerns $\boldsymbol{\theta}_2$, which is most natural. Note that the structure of (7) guarantees that the MLE of $\boldsymbol{\theta}_1$ is the same under the null and alternative hypotheses. Letting $\widehat{\boldsymbol{\theta}}_{0,2}$ denote the restricted MLE of $\boldsymbol{\theta}_2$ under $H_0$, the likelihood ratio for the unconditional model is

$$\begin{aligned} \lambda &= \frac{L_2(\widehat{\boldsymbol{\theta}}_{0,2}, \mathbf{x}, \mathbf{y})\, L_1(\widehat{\boldsymbol{\theta}}_1, \mathbf{x})}{L_2(\widehat{\boldsymbol{\theta}}_2, \mathbf{x}, \mathbf{y})\, L_1(\widehat{\boldsymbol{\theta}}_1, \mathbf{x})} \\ &= \frac{L_2(\widehat{\boldsymbol{\theta}}_{0,2}, \mathbf{x}, \mathbf{y})}{L_2(\widehat{\boldsymbol{\theta}}_2, \mathbf{x}, \mathbf{y})}, \end{aligned}$$

which again is exactly what it would have been under a conditional model. While this holds only because the likelihood has the nice structure in (7), it's a fairly reasonable set of assumptions.

Thus in terms of both estimation and hypothesis testing, the fact that explanatory variables are usually random variables presents no difficulty, regardless of what the distribution of those explanatory variables may be. In fact, the conditional nature of the usual regression model is a strength. In all the calculations above, the joint distribution of the explanatory variables is written in a very general way. It really doesn't matter what it is, because it disappears. So one might say that with respect to the explanatory variables, the usual linear regression model is distribution free.

In spite of the virtues of the conditional regression model, in this book we will focus on *unconditional* regression models, in which the explanatory variables are random. The reason is that ultimately, the explanatory variables themselves may be influenced by other variables. The easiest way to represent this is to admit from the outset that they are random variables.

## 0.3 Unconditional regression with observed variables

**Example 0.3.1** *Simple Regression*

Suppose that the covariance between two random variables arises from a regression. Independently for $i = 1, \ldots, n$, let

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{8}$$

where

- $X_i$ is has expected value $\mu_x$ and variance $\phi > 0$
- $\epsilon_i$ has expected value zero and variance $\sigma^2 > 0$
- $X_i$ and $\epsilon_i$ are independent.

The pairs $(X_i, Y_i)$ have a joint distribution that is unspecified, except for the expected value

$$E\begin{pmatrix} X_i \\ Y_i \end{pmatrix} = \boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} \mu_x \\ \beta_0 + \beta_1\mu_x \end{pmatrix},$$

and variance-covariance matrix

$$cov\begin{pmatrix} X_i \\ Y_i \end{pmatrix} = \boldsymbol{\Sigma} = [\sigma_{i,j}] = \begin{pmatrix} \phi & \beta_1\phi \\ \beta_1\phi & \beta_1^2\phi + \sigma^2 \end{pmatrix}.$$

The linear property of the covariance (Expression 1 on page 9) is useful for calculating the covariance between the explanatory and response variables.

$$\begin{aligned} Cov(X_i, Y_i) &= Cov(X_i, \beta_0 + \beta_1 X_i + \epsilon_i) \\ &= Cov(X_i, \beta_1 X_i + \epsilon_i) \\ &= \beta_1 Cov(X_i, X_i) + Cov(X_i, \epsilon_i) \\ &= \beta_1 Var(X_i) + 0 \\ &= \beta_1\phi \end{aligned}$$

Since $\phi$ is a variance, it is greater than zero. Thus the sign of the covariance is the sign of the regression coefficient. Positive regression coefficients produce positive relationships, negative regression coefficients produce negative relationships, and zero corresponds to no relationship as measured by the covariance.

While the sign of the covariance (and hence the direction of the relationship) is determined by $\beta_1$, the magnitude of the covariance is jointly determined by the magnitude of $\beta_1$ and the magnitude of $\phi$, the variance of $X_i$. Consequently the covariance of $X_i$ and $Y_i$ depends on the scale of measurement of $X_i$. If $X_i$ is measured in centimeters instead of meters, its variance is $100^2 = 10,000$ times as great, and $Cov(X_i, Y_i)$ is ten thousand times as great, as well. This makes raw covariances difficult to interpret, except for the sign.

A solution is to put the variables on a standard common scale by looking at correlations instead of covariances. Denoting the correlation of any two random variables $X$ and $Y$ by Greek letter "rho," which is a common notation,

$$\begin{aligned} \rho_{xy} &= \frac{Cov(X,Y)}{SD(X)SD(Y)} \\ &= \frac{E\left\{(X-\mu_x)(Y-\mu_y)\right\}}{\sqrt{Var(X)}\sqrt{Var(Y)}} \\ &= E\left\{\left(\frac{X-\mu_x}{\sigma_x}\right)\left(\frac{Y-\mu_y}{\sigma_y}\right)\right\}. \end{aligned} \tag{9}$$

That is, the correlation between two random variables is the covariance between versions of the variables that have been standardized to have mean zero and variance one. Using (9), the correlation for Example 0.3.1 is

$$\begin{aligned} \rho &= \frac{\beta_1\phi}{\sqrt{\phi}\sqrt{\beta_1^2\phi+\sigma^2}} \\ &= \frac{\beta_1\sqrt{\phi}}{\sqrt{\beta_1^2\phi+\sigma^2}}. \end{aligned} \tag{10}$$

This may not look like much, but consider the following. In any regression, the response variable is likely to represent the phenomenon of primary interest, and explaining why it varies from unit to unit is an important scientific goal. For example, if $Y_i$ is academic performance, we want to know why some students do better than others. If $Y_i$ is the crime rate in neighbourhood $i$, we want to know why there is more crime in some neighbourhoods than in others. If there were no variation in some phenomenon (the sum rises in the East) there might still be something to explain, but it would not be a statistical question. Because $X_i$ and $\epsilon_i$ are independent,

$$\begin{aligned} Var(Y_i) &= Var(\beta_1 X_i + \epsilon_i) \\ &= \beta_1^2 Var(X_i) + Var(\epsilon_i) \\ &= \beta_1^2\phi + \sigma^2. \end{aligned}$$

Thus the variance of $Y_i$ is separated into two parts[4], the part that comes from $X_i$ and the part that comes from $\epsilon_i$. The part that comes from $X_i$ is $\beta_1^2\phi$, and the part that comes from $\epsilon_i$ (that is, everything else) is $\sigma^2$. From (10) the *squared* correlation between $X_i$ and $Y_i$ is

$$\rho^2 = \frac{\beta_1^2\phi}{\beta_1^2\phi+\sigma^2}, \tag{11}$$

the proportion of the variance in $Y_i$ that comes from $X_i$. This quantity does not depend on the scale of $X_i$ or the scale of $Y_i$, because both variables are standardized.

**Example 0.3.2** *Multiple Regression*

Now consider multiple regression. In ordinary multiple regression (the conditional model), one speaks of the relationship between and explanatory variable and the response variable "controlling" for other variables in the model[5]. This really refers to the conditional expectation of $Y$ as a function of $x_j$ for fixed values of the other $x$ variables, say in the sense of a partial derivative. In unconditional regression with random explanatory variables one talks about it in the same way, but the technical version is a bit different and perhaps easier to understand. Here is an example with two explanatory variables.

[4]The word "analysis" means splitting into parts, so this is literally analysis of variance.

[5]One can also speak of "correcting" for the other variables, or "holding them constant," or "allowing" for them, or "taking them into account." These are all ways of saying exactly the same thing.

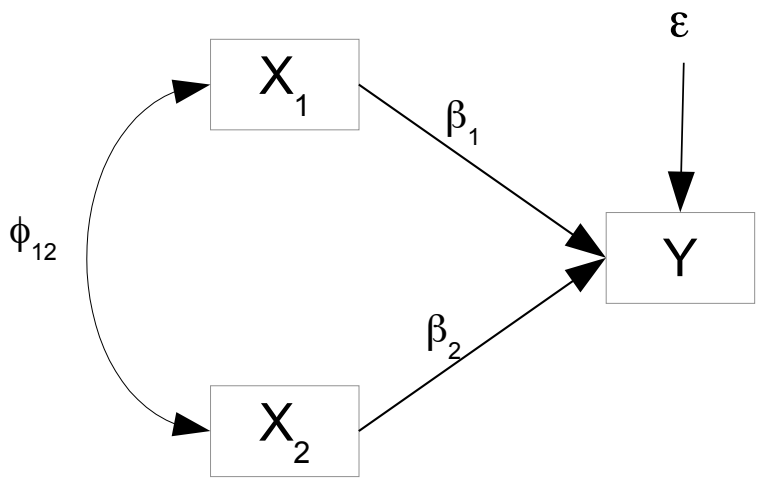
Independently for $i = 1, \ldots, n$, let $Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i$, where $E(X_{i,1}) = \mu_1$, $E(X_{i,2}) = \mu_2$, $E(\epsilon_i) = 0$, $Var(\epsilon_i) = \sigma^2$, $\epsilon_i$ is independent of both $X_{i,1}$ and $X_{i,2}$, and

$$cov \begin{pmatrix} X_{i,1} \\ X_{i,2} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}.$$

Figure 3 shows a path diagram for this model. The explanatory and response variables are all observed, so they are enclosed in boxes. The double-headed curved arrow between the explanatory variables represents a possibly non-zero covariance. This covariance might arises from interesting and important processes including common influences on the $X$ variables, but those processes are not part of the model. Curved double-headed arrows represent *unanalyzed* covariances between explanatory variables.

The straight arrows from the explanatory to response variables represent direct influence, or at least that we are interested in predicting $y$ from $x$ rather than the other way around. There is a regression coefficient $\beta$ on each straight arrow, and a covariance $\phi_{12}$ on the curved double-headed arrow.

Figure 3: Unconditional multiple regression




For this model, the covariance of $X_{i,1}$ and $Y_i$ is

$$\begin{aligned} Cov(X_{i,1}, Y_i) &= Cov(X_{i,1}, \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i) \\ &= Cov(X_{i,1}, \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i) \\ &= \beta_1 Cov(X_{i,1}, X_{i,1}) + \beta_2 Cov(X_{i,1}, X_{i,2}) + Cov(X_{i,1}, \epsilon_i) \\ &= \beta_1 Var(X_{i,1}) + \beta_2 Cov(X_{i,1}, X_{i,2}) + 0 \\ &= \beta_1 \phi_{11} + \beta_2 \phi_{12} \end{aligned}$$

This means that the relationship between $X_1$ and $Y$ has two sources. One is the direct link from $X_1$ to $Y$ through the straight arrow represented by $\beta_1$, and the other is through

the curved arrow between $X_1$ and $X_2$ and then through the straight arrow linking $X_2$ to $Y$. Even if $\beta_1 = 0$, there still will be a relationship provided that $X_1$ is related to $X_2$ and $X_2$ is related to $Y$[6]. Furthermore, $\beta_2\phi_{12}$ may overwhelm $\beta_1\phi_{11}$, so that the covariance between $X_1$ and $Y$ may be positive even though $\beta_1$ is negative.

All this is true of the unconditional relationship between $X_1$ and $Y$, but what if you "control" for $X_2$ by holding it constant at some fixed value? When the explanatory variables are all random, the relationship between $X_1$ and $Y$ controlling for $X_2$ simply refers to a conditional distribution — the joint distribution of $X_1$ and $Y$ given $X_2 = x_2$. In this case the regression equation is

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_{i,1} + \beta_2 x_{i,2} + \epsilon_i \\ &= (\beta_0 + \beta_2 x_{i,2}) + \beta_1 X_{i,1} + \epsilon_i \\ &= \beta_0' + \beta_1 X_{i,1} + \epsilon_i \end{aligned}$$

The constant is simply absorbed into the intercept. It's a little strange in that the intercept is potentially different for $i = 1, \ldots, n$, but that doesn't affect the covariance. Following the calculations in Example 0.3.1, the conditional covariance between $X_{i,1}$ and $Y_i$ is $\beta_1\phi_{11}$. Thus to test whether $X_1$ is connected to $Y$ controlling for $X_2$ (or correcting for it, or allowing for it or some such term), it is appropriate to test $H_0 : \beta_1 = 0$. If the null hypothesis is rejected, the sign of the estimated regression coefficient guides your conclusion as to whether the conditional relationship is positive or negative. These considerations extend immediately to multiple regression.

In terms of interpreting the regression coefficients, it is helpful to decompose (analyze) the variance of $Y_i$.

$$\begin{aligned} Var(Y_i) &= Var(\beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i) \\ &= \beta_1^2\phi_{11} + \beta_2^2\phi_{22} + 2\beta_1\beta_2\phi_{12} + \sigma^2 \end{aligned}$$

The explanatory variables contribute to the variance of the response individually through their variances and squared regression coefficients, and also jointly through their regression coefficients and their covariance. This joint effect is not an interaction in the ordinary sense of the term; the model of Example 0.3.2 has no product term. The null hypothesis $H_0 : \beta_1 = 0$ means that $X_1$ does not contribute at all to the variance of $Y$, either directly or through its covariance with $X_2$.

## Estimation

Here is some useful terminology, repeated from Appendix A.

**Definition 0.1** Moments *of a distribution are quantities such* $E(X)$, $E(Y^2)$, $Var(X)$, $E(X^2Y^2)$, $Cov(X, Y)$, *and so on.*

[6] Yes, body weight may be positively related to income because men are bigger on average and they tend to make more money for the same work.

**Definition 0.2** Moment structure equations *are a set of equations expressing moments of the distribution of the data in terms of the model parameters. If the moments involved are limited to variances and covariances, the moment structure equations are called* covariance structure equations.

For the simple (one explanatory variable) regression model of Example 0.3.1, the moments are the elements of the mean vector $\boldsymbol{\mu} = E\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$, and the unique elements of the covariance matrix $\boldsymbol{\Sigma} = cov\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$. The moments structure equations are

$$\begin{aligned} \mu_1 &= \mu_x \\ \mu_2 &= \beta_0 + \beta_1\mu_x \\ \sigma_{1,1} &= \phi \\ \sigma_{1,2} &= \beta_1\phi \\ \sigma_{2,2} &= \beta_1^2\phi + \psi. \end{aligned} \tag{12}$$

In this model, the parameters are $\mu_x$, $\phi$, $\beta_0$, $\beta_1$, $\psi$, and also the unknown distribution functions of $X_i$ and $\epsilon_i$. Our interest is in the Greek-letter parameters, especially $\beta_0$ and $\beta_1$. Method of Moments estimates (See Section A.6.2 in Appendix A) can be obtained by solving the moment structure equations (12) for the unknown parameters and putting hats on the result. The moment structure equations form a system of 5 equations in five unknowns, and may be readily be solved to yield

$$\begin{aligned} \beta_0 &= \mu_2 - \frac{\sigma_{1,2}}{\sigma_{1,1}}\mu_1 \\ \mu_x &= \mu_1 \\ \phi &= \sigma_{1,1} \\ \beta_1 &= \frac{\sigma_{1,2}}{\sigma_{1,1}} \\ \psi &= \sigma_{2,2} - \frac{\sigma_{1,2}^2}{\sigma_{1,1}}. \end{aligned} \tag{13}$$

Thus, even though the distributions of $X_i$ and $\epsilon_i$ are unknown, we have nice consistent[7] estimators of the interesting part of the unknown parameter. Putting hats on the param-

[7] By the Law of Large Numbers and continuous mapping

eters in Expression 13,

$$\begin{aligned}
\widehat{\beta}_0 &= \overline{y} - \frac{\widehat{\sigma}_{1,2}}{\widehat{\sigma}_{1,1}}\overline{x} \\
\widehat{\mu}_x &= \widehat{\mu}_1 = \overline{x} \\
\widehat{\phi} &= \widehat{\sigma}_{1,1} \\
\widehat{\beta}_1 &= \frac{\widehat{\sigma}_{1,2}}{\widehat{\sigma}_{1,1}} \\
\widehat{\psi} &= \widehat{\sigma}_{2,2} - \frac{\widehat{\sigma}_{1,2}^2}{\widehat{\sigma}_{1,1}}.
\end{aligned}$$

It is very standard to assume that $X_i$ and $\epsilon_i$ are normally distributed. In this case, the existence of the solution (13) tells us that the parameters of the normal version of this regression model stand in a one-to-one-relationship with the mean and covariance matrix of the bivariate normal distribution posessed by the observable data. In fact, the two sets of parameter values are 100% equivalent; they are just different ways of expressing the same thing. For some purposes, the parameterization represented by the regression model may be more informative.

Furthermore, the Invariance Principle of maximum likelihood estimation (see Section A.6.5 in Appendix A) says that the MLE of a one-to-one function is just that function of the MLE. So, the Method of Moments estimates are also the Maximum Likelihood estimates in this case. Recognizing the formula for $\widehat{\beta}_1$ as a special case of Expression 4 on Page 14 (from the centered multiple regression model), we see that $\widehat{\beta}_1$ is also the ordinary least-squares estimate.

The calculations just shown are important, because they are an easy, clear example of something that will be necessary again and again throughout the course. Here is the process:

- Calculate the moments of the distribution (usually means, variances and covariances) in terms of the model parameters, obtaining a system of moment structure equations.
- Solve the moment structure equations for the parameters, expressing the parameters in terms of the moments.

When the second step is successful, putting hats on all the parameters in the solution yields Method of Moments estimators, even when these do not correspond to the MLEs[8].

It turns out that for many reasonable models that go beyond ordinary multiple regression, a unique solution for the parameters is mathematically impossible. In such cases, successful parameter estimation by any method is impossible as well. It is vitally important to verify the *possibility* of successful parameter estimation before trying it for a

[8]When there are the same number of moment structure equations and a unique sulution for the parametrers exists, the Mothod of Moments estimators and MLEs coincide. When there are more equations than parameters they no longer coincide in general, but still the process of "putting hats on everything" yields Method of Moments estimators.

given data set (say, by maximum likelihood), and verification consists of a process like the one you have just seen. Of course it is no surprise that estimating the parameters of a regression model is technically possible.

Because the process is so important, let us take a look at the extension to multivariate multiple regression — that is, to linear regression with multiple explanatory variables and multiple response variables. This will illustrate the matrix versions of the calculations.

**Example 0.3.3** *Multivariate Regression*

Independently for $i = 1, \ldots, n$, let

$$\mathbf{y}_i = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1^\top \mathbf{x}_i + \boldsymbol{\epsilon}_i \tag{14}$$

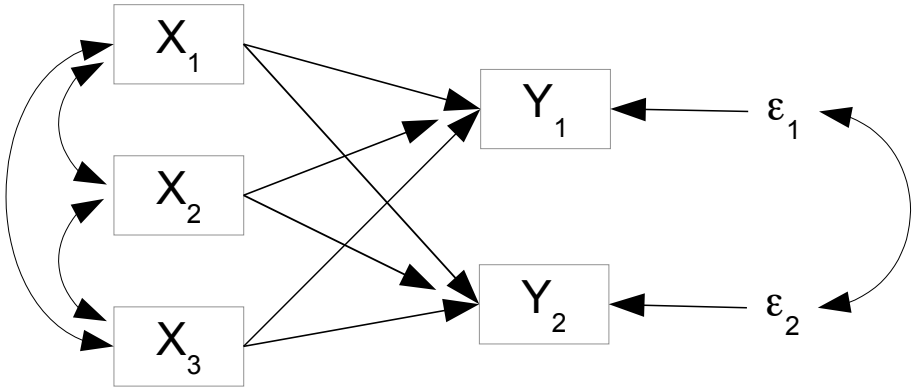
where

> $\mathbf{y}_i$ is a $q \times 1$ random vector of observable response variables, so the regression can be multivariate; there are $q$ response variables.
>
> $\boldsymbol{\beta}_0$ is a $q \times 1$ vector of unknown constants, the intercepts for the $q$ regression equations. There is one for each response variable.
>
> $\mathbf{x}_i$ is a $p \times 1$ observable random vector; there are $p$ explanatory variables. $\mathbf{x}_i$ has expected value $\boldsymbol{\mu}_x$ and variance-covariance matrix $\boldsymbol{\Phi}$, a $p \times p$ symmetric and positive definite matrix of unknown constants.
>
> $\boldsymbol{\beta}_1$ is a $p \times q$ matrix of unknown constants. These are the regression coefficients, with one row for each explanatory variable and one column for each response variable.
>
> $\boldsymbol{\epsilon}_i$ is the error term of the latent regression. It is an $q \times 1$ multivariate normal random vector with expected value zero and variance-covariance matrix $\boldsymbol{\Psi}$, a $q \times q$ symmetric and positive definite matrix of unknown constants. $\boldsymbol{\epsilon}_i$ is independent of $\mathbf{x}_i$.

The parameter vector for this model could be written $\boldsymbol{\theta} = (\boldsymbol{\beta}_0, \boldsymbol{\mu}_x, \boldsymbol{\Phi}, \boldsymbol{\beta}_1, \boldsymbol{\Psi}, F_{\mathbf{x}}, F_{\boldsymbol{\epsilon}})$, where it is understood that the symbols for the matrices refer to their unique elements.

Figure 4 depicts a model with three explanatory variables and two response variables. The explanatory and response variables are all observable, so they are enclosed in boxes. Double-headed curved arrows between the explanatory variable represent possible non-zero covariances. The straight arrows from the explanatory to response variables represent direct influence, or at least that we are interested in predicting $y$ from $x$ rather than the other way around. There is a regression coefficient $\beta_{j,k}$ on each arrow. The error terms $\epsilon_1$ and $\epsilon_2$ represent all other influences on $Y_1$ and $Y_2$. Since there could be common influences (omitted variables that affect both $Y_1$ and $Y_2$), the error terms are assumed to be correlated. This is the reason for the curved double-headed arrow joining $\epsilon_1$ and $\epsilon_2$.

Figure 4: Multivariate multiple regression




There is one regression equation for each response variable. In scalar form, the model equations are

$$\begin{aligned} Y_{i,1} &= \beta_{0,1} + \beta_{1,1}X_{i,1} + \beta_{2,1}X_{i,2} + \beta_{3,1}X_{i,3} + \epsilon_{i,1} \\ Y_{i,2} &= \beta_{0.2} + \beta_{1,2}X_{i,1} + \beta_{2,2}X_{i,2} + \beta_{3,2}X_{i,3} + \epsilon_{i,2}. \end{aligned}$$

In matrix form,

$$\begin{array}{ccccccc} \mathbf{y}_i & = & \boldsymbol{\beta}_0 & + & \boldsymbol{\beta}_1^\top & \mathbf{x}_i & + & \boldsymbol{\epsilon}_i \end{array}$$

$$\begin{pmatrix} Y_{i,1} \\ Y_{i,2} \end{pmatrix} = \begin{pmatrix} \beta_{1,0} \\ \beta_{2,0} \end{pmatrix} + \begin{pmatrix} \beta_{1,1} & \beta_{2,1} & \beta_{3,1} \\ \beta_{1,2} & \beta_{2,2} & \beta_{3,2} \end{pmatrix} \begin{pmatrix} X_{i,1} \\ X_{i,2} \\ X_{i,3} \end{pmatrix} + \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \end{pmatrix}$$

Returning to the general case of Example 0.3.3, the observable data are the random vectors $\mathbf{D}_i = \begin{pmatrix} \mathbf{x}_i \\ \hline \mathbf{y}_i \end{pmatrix}$, for $i = 1, \ldots, n$. The notation indicates that $\mathbf{D}_i$ is a partitioned random vector, with $\mathbf{x}_i$ stacked directly on top of $\mathbf{y}_i$. Using the notation $E(\mathbf{D}_i) = \boldsymbol{\mu}$ and $cov(\mathbf{D}_i) = \boldsymbol{\Sigma}$, one may write $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as partitioned matrices (matrices of matrices).

$$\boldsymbol{\mu} = \begin{pmatrix} E(\mathbf{x}_i) \\ \hline E(\mathbf{y}_i) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \hline \boldsymbol{\mu}_2 \end{pmatrix}$$

and

$$\boldsymbol{\Sigma} = cov \begin{pmatrix} \mathbf{x}_i \\ \hline \mathbf{y}_i \end{pmatrix} = \left(\begin{array}{c|c} cov(\mathbf{x}_i) & cov(\mathbf{x}_i, \mathbf{y}_i) \\ \hline cov(\mathbf{x}_i, \mathbf{y}_i)^\top & cov(\mathbf{y}_i) \end{array}\right) = \left(\begin{array}{c|c} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \hline \boldsymbol{\Sigma}_{12}^\top & \boldsymbol{\Sigma}_{22} \end{array}\right)$$

As in the univariate case, the maximum likelihood estimators may be obtained by solving the moment structure equations for the unknown parameters. The moment structure

equations are obtained by calculating expected values and covariances in terms of the model parameters. All the calculations are immediate except possibly

$$\begin{aligned}
\boldsymbol{\Sigma}_{12} &= cov(\mathbf{x}_i, \mathbf{y}_i) \\
&= cov\left(\mathbf{x}_i \ , \ \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1^\top \mathbf{x}_i + \boldsymbol{\epsilon}_i\right) \\
&= cov\left(\mathbf{x}_i \ , \ \boldsymbol{\beta}_1^\top \mathbf{x}_i + \boldsymbol{\epsilon}_i\right) \\
&= cov\left(\mathbf{x}_i, \mathbf{x}_i\right)\boldsymbol{\beta}_1 + cov(\mathbf{x}_i, \boldsymbol{\epsilon}_i) \\
&= cov(\mathbf{x}_i)\boldsymbol{\beta}_1 + \mathbf{0} \\
&= \boldsymbol{\Phi}\boldsymbol{\beta}_1
\end{aligned}$$

Thus, the moment structure equations are

$$\begin{aligned}
\boldsymbol{\mu}_1 &= \boldsymbol{\mu}_x \\
\boldsymbol{\mu}_2 &= \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1^\top \boldsymbol{\mu}_x \\
\boldsymbol{\Sigma}_{11} &= \boldsymbol{\Phi} \\
\boldsymbol{\Sigma}_{12} &= \boldsymbol{\Phi}\boldsymbol{\beta}_1 \\
\boldsymbol{\Sigma}_{22} &= \boldsymbol{\beta}_1^\top \boldsymbol{\Phi}\boldsymbol{\beta}_1 + \boldsymbol{\Psi}.
\end{aligned} \tag{15}$$

Solving for the parameter matrices is routine.

$$\begin{aligned}
\boldsymbol{\beta}_0 &= \boldsymbol{\mu}_2 - \boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\mu}_1 \\
\boldsymbol{\mu}_x &= \boldsymbol{\mu}_1 \\
\boldsymbol{\Phi} &= \boldsymbol{\Sigma}_{11} \\
\boldsymbol{\beta}_1 &= \boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12} \\
\boldsymbol{\Psi} &= \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{12}^\top\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}
\end{aligned} \tag{16}$$

As in the univariate case, the Method of Moments estimates are obtained by putting hats on all the parameters in Expression (16). If the distributions of $\mathbf{x}_i$ and $\boldsymbol{\epsilon}_i$ are multivariate normal, the Invariance Principle implies that these Method of Moments estimates are also the maximum likelihood estimates.

**Least Squares** Recall that in the proof of consistency for ordinary least squares with random explanatory variables, we centered the explanatory variables and obtained Formula (4) on Page 14: $\widehat{\boldsymbol{\beta}}_n = \widehat{\boldsymbol{\Sigma}}_x^{-1}\widehat{\boldsymbol{\Sigma}}_{xy}$. Compare this to the estimate of the slopes obtained from the solution (16) above: $\widehat{\boldsymbol{\beta}}_1 = \widehat{\boldsymbol{\Sigma}}_{11}^{-1}\widehat{\boldsymbol{\Sigma}}_{12}$. The formulas are almost the same. $\widehat{\boldsymbol{\Sigma}}_{11} = \widehat{\boldsymbol{\Sigma}}_x$, the sample variance-covariance matrix of the explanatory variables. $\widehat{\boldsymbol{\Sigma}}_{12}$ and $\widehat{\boldsymbol{\Sigma}}_{xy}$ are both matrices of sample covariances between explanatory and response variables, except that $\widehat{\boldsymbol{\Sigma}}_{12}$ is $p \times q$ while $\widehat{\boldsymbol{\Sigma}}_{xy}$ is $p \times 1$. $\widehat{\boldsymbol{\Sigma}}_{12}$ has one column for each response variable. So, in addition to being a method of moments estimate and a maximum likelihood estimate under normality $\widehat{\boldsymbol{\beta}}_1$ is a $p \times q$ matrix of least-squares estimates,

## 0.4 Omitted Variables

Some very serious problems can arise when standard regression methods are applied to non-experimental data. Note that regression methods are applied to non-experimental data *all the time*, and we teach students how to do it in almost every statistics class where regression is mentioned. Without an understanding of the technical issues involved, the typical applications can be misleading.

The trouble is not the explanatory variables are random. As we saw in Section 0.2, that's fine. But when the random explanatory variables have non-zero correlations with other explanatory variables that are missing from the regression equation and are related to the response variable, things can get ugly. In this section, we will see how omitting important explanatory variables from a regression equation can cause the error term to be correlated with the explanatory variables that remain, and how that can produce incorrect results.

To appreciate the issue, it is necessary to understand what the error term in a regression equation really represents. When we write something like

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \epsilon_i, \tag{17}$$

we are saying that $X_{i,1}$ contributes to $Y_i$, but there are also other, unspecified influences. Those other influences are all rolled together into $\epsilon_i$.

The words "contributes" and "influences" are used deliberately. They should be setting off alarm bells, because they imply a causal connection between $X_i$ and $Y_i$. Regression models with random explanatory variables are applied mostly to observational data, in which explanatory variables are merely recorded rather than being manipulated by the investigator. The correlation-causation issue applies. That is, if $X$ and $Y$ are related, there is in general no way to tell whether $X$ is influencing $Y$, or $Y$ is influencing $X$, or if other variables are influencing both $X$ and $Y$.

It could be argued that a *conditional* regression model (the usual model in which the explanatory variable values are fixed constants) is just a convenient way to represent dependence between $X$ and $Y$ by specifying a generic, more or less reasonable conditional distribution for $Y$ given $X = x$. In this case, the correlation-causation issue can be set aside, and taken up when it is time to interpret the results. But if the explanatory variables are explicitly random, it is harder to avoid the obvious. In the simple regression model (17), the random variable $Y_i$ is a function of the random variables $X_i$ and $\epsilon_i$. It is being directly produced by them. If this is taken seriously as a *scientific* model as well as a statistical model[9], it is inescapably causal; it is a model of what affects what. That's why the straight arrows in path diagrams are directional. The issue of whether $X$ is influencing $Y$, or $Y$ is influencing $X$ or both is a modelling issue that will mostly be decided based on subject-matter theory.

It is natural to ask whether the data can be used to decide which way the arrows should be pointing. The answer is usually no, but it can be yes with certain other restrictions

[9]In structural equation modelling, the models are both statistical models and primitive scientific models of the data. Once the general linear structural model is introduced, you will see that regression is a special case.

on the model. We will return to this issue later in the book. In the meantime, regression models with random explanatory variables, like the general structural equation models that are their extensions, will be recognized as causal models.

Again, Equation (17) says that $X_i$ is influencing $Y_i$. All other influences are represented by $\epsilon_i$. It is common practice to assume that $X_{i,1}$ and $\epsilon_i$ are independent, or at least uncorrelated. But that does not mean the assumption can be justified in practice. Prepare yourself for a dose of reality.

**Example 0.4.1** *Omitted Explanatory Variables*

Suppose that the variables $X_2$ and $X_3$ have an impact on $Y$ and are correlated with $X_1$, but they are not part of the data set. The values of the response variable are generated as follows:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \beta_3 X_{i,3} + \epsilon_i, \tag{18}$$

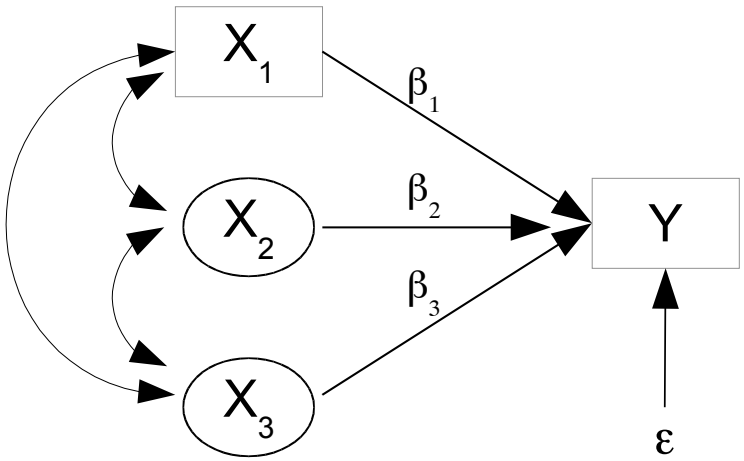
independently for $i = 1, \ldots, n$, where $\epsilon_i \sim N(0, \sigma^2)$. The explanatory variables are random, with expected value and variance-covariance matrix

$$E\begin{pmatrix} X_{i,1} \\ X_{i,2} \\ X_{i,3} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} \quad \text{and} \quad cov\begin{pmatrix} X_{i,1} \\ X_{i,2} \\ X_{i,3} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ & \phi_{22} & \phi_{23} \\ & & \phi_{33} \end{pmatrix},$$

where $\epsilon_i$ is independent of $X_{i,1}$, $X_{i,2}$ and $X_{i,3}$. Values of the variables $X_{i,2}$ and $X_{i,3}$ are latent, and are not included in the data set.

Figure 5 shows a path diagram of this model. Because the explanatory variables $X_{i,2}$ and $X_{i,3}$ are not observable, they are *latent* variables, and so they are encolsed by ovals in the path diagram. Their covariances with $X_{i,1}$ and each other are represented by two-headed curved arrows.

Figure 5: Omitted explanatory variables



Since $X_2$ and $X_3$ are not available, we use what we have, and consider a model with $X_1$ only. In this case $X_2$ and $X_3$ are absorbed by the intercept and error term, as follows.

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \beta_3 X_{i,3} + \epsilon_i \\ &= (\beta_0 + \beta_2\mu_2 + \beta_3\mu_3) + \beta_1 X_{i,1} + (\beta_2 X_{i,2} + \beta_3 X_{i,3} - \beta_2\mu_2 - \beta_3\mu_3 + \epsilon_i) \\ &= \beta_0' + \beta_1 X_{i,1} + \epsilon_i'. \end{aligned}$$

The primes just denote a new $\beta_0$ and a new $\epsilon$; the addition and subtraction of $\beta_2\mu_2 + \beta_3\mu_3$ serve to make $E(\epsilon_i') = 0$. And of course there could be any number of omitted variables. They would all get swallowed by the intercept and error term, the garbage bins of regression analysis.

Notice that although the original error term $\epsilon_i$ is independent of $X_{i,1}$, the new error term $\epsilon_i'$ is not.

$$\begin{aligned} Cov(X_{i,1}, \epsilon_i') &= Cov(X_{i,1}, \beta_2 X_{i,2} + \beta_3 X_{i,3} - \beta_2\mu_2 - \beta_3\mu_3 + \epsilon_i) \\ &= \beta_1 Cov(X_{i,1}, X_{i,2}) + \beta_3 Cov(X_{i,1}, X_{i,3}) + 0 \\ &= \beta_2\phi_{12} + \beta_3\phi_{13} \end{aligned} \tag{19}$$

So, when explanatory variables are omitted from the regression equation and those explanatory variables have non-zero covariance with variables that *are* in the equation, the result is non-zero covariance between the error term and the explanatory variables in the equation[10].

Response variables are almost always affected by more than one explanatory variable, and in observational data, explanatory variables usually have non-zero covariances with one another. So, the most realistic model for a regression with just one explanatory variable should include a covariance between the error term and the explanatory variable. The covariance comes from the regression coefficients and covariances of some unknown number of omitted variables; it will be represented by a single quantity because there is no hope of estimating all those parameters individually. We don't even know how many there are.

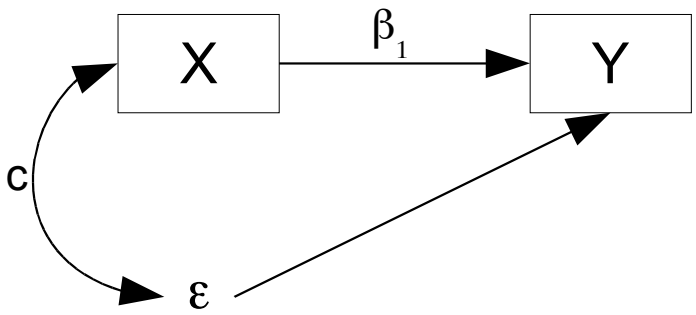
We have arrived at the following model, which will be called the *true model* in the discussion that follows. It may not be the ultimate truth of course, but for observational data it is almost always closer to the truth than the usual model. Independently for $i = 1, \ldots, n$,

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \tag{20}$$

where $E(X_i) = \mu_x$, $Var(X_i) = \sigma_x^2$, $E(\epsilon_i) = 0$, $Var(\epsilon_i) = \sigma_\epsilon^2$, and $Cov(X_i, \epsilon_i) = c$. A path diagram of the true model is given in Figure 6. The covariance $c$ is indicated on the curved arrow connecting the explanatory variable and the error term. Consider a data set consisting of pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$ *coming from the true model*, and the interest is in the regression coefficent $\beta_1$. Who will try to estimate the parameters of the true model? Almost no one. Practically everyone will use ordinary least squares, as

[10]The effects of the omitted variables could offset each other. In this example, it is possible that $\beta_2\phi_{12} + \beta_3\phi_{13} = 0$, but that is really too much to hope.

Figure 6: Omitted explanatory variables have been swallowed by $\epsilon$




described in countless textbooks and implemented in countless computer programs and even statistical calculators.

The model underlying ordinary least squares is $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, where $x_1, \ldots, x_n$ are fixed constants, and conditionally on $x_1, \ldots, x_n$, the error terms $\epsilon_1, \ldots, \epsilon_n$ are independent normal random variables with mean zero and variance $\sigma^2$. It may not be immediately obvious, but this model implies independence of the explanatory variable and the error term. It is a conditional model, and the distribution of the error terms is *the same* for every fixed set of values $x_1, \ldots, x_n$. Using a loose but understandable notation for densities and conditional densities,

$$\begin{aligned} & f(\epsilon_i|x_i) = f(\epsilon_i) \\ \Leftrightarrow \quad & \frac{f(\epsilon_i, x_i)}{f(x_i)} = f(\epsilon_i) \\ \Leftrightarrow \quad & f(\epsilon_i, x_i) = f(\epsilon_i) f(x_i), \end{aligned}$$

which is the definition of independence. So, the usual regression model makes a hidden assumption. It assumes that *any explanatory variable that is omitted from the equation has zero covariance with the variables that are in the equation.*

Surprisingly, this does not depend on the assumption of any particular distribution for the error terms. All you need is the stipulation $E(\epsilon_i) = 0$ in a fixed-$x$ regression model. It's worth doing this in generality, so consider the multivariate multiple regression model of Example 0.3.3 on page 24:

$$\mathbf{Y}_i = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1^\top \mathbf{X}_i + \boldsymbol{\epsilon}_i.$$

If the $\mathbf{X}_i$ values are considered fixed constants, the statement $E(\boldsymbol{\epsilon}_i) = \mathbf{0}$ actually means $E(\boldsymbol{\epsilon}_i|\mathbf{X}_i = \mathbf{x}_i) = \mathbf{0}$ for all $p \times 1$ constant vectors $\mathbf{x}_i$ in the support of $\mathbf{X}_i$. Then,

$$E(\boldsymbol{\epsilon}_i) = E\{E(\boldsymbol{\epsilon}_i|\mathbf{X}_i)\} = E\{\mathbf{0}\} = \mathbf{0},$$

and

$$\begin{aligned} cov(\mathbf{X}_i, \boldsymbol{\epsilon}_i) &= E(\mathbf{X}_i \boldsymbol{\epsilon}_i^\top) - E(\mathbf{X}_i) E(\boldsymbol{\epsilon}_i)^\top \\ &= E(\mathbf{X}_i \boldsymbol{\epsilon}_i^\top) - \mathbf{0} \\ &= E\{E(\mathbf{X}_i \boldsymbol{\epsilon}_i^\top | \mathbf{X}_i)\}. \end{aligned}$$

The inner expected value is a multiple integral or sum with respect to the conditional distribution of $\boldsymbol{\epsilon}_i$ given $\mathbf{X}_i$, so $\mathbf{X}_i$ may be moved through the inner expected value sign. To see this, it may help to write the double expectation in terms of integrals of a general kind[11]. Continuing the calculation,

$$\begin{aligned} E\{E(\mathbf{X}_i \boldsymbol{\epsilon}_i^\top | \mathbf{X}_i)\} &= \int \left( \int \mathbf{x} \boldsymbol{\epsilon}^\top dP_{\boldsymbol{\epsilon}|\mathbf{X}}(\boldsymbol{\epsilon}) \right) dP_{\mathbf{X}}(\mathbf{x}) \\ &= \int \mathbf{x} \left( \int \boldsymbol{\epsilon}^\top dP_{\boldsymbol{\epsilon}|\mathbf{X}}(\boldsymbol{\epsilon}) \right) dP_{\mathbf{X}}(\mathbf{x}) \\ &= E\{\mathbf{X}_i E(\boldsymbol{\epsilon}_i^\top | \mathbf{X}_i)\} \\ &= E\{\mathbf{X}_i \mathbf{0}^\top\} \\ &= E\{\mathbf{0}\} \\ &= \mathbf{0} \end{aligned}$$

Unconditional (random $\mathbf{X}$) regression models typically assume zero covariance between error terms and explanatory variables. It is now clear that conditional (fixed $\mathbf{x}$) regression models smuggle this same assumption in by making the seemingly reasonable and harmless assertion that $E(\boldsymbol{\epsilon}_i) = \mathbf{0}$.

Zero covariance between error terms and explanatory variables means that *any potential explanatory variable not in the model must have zero covariance with the explanatory variables that are in the model*. Of course this is almost never realistic without random assignment to experimental conditions, so that almost every application of regression methods to non-experimental data makes an assumption that cannot be justified. Now we will see the consequences.

For a simple regression, both ordinary least squares and an unconditional regression model like the true model on Page 29 with $c = 0$ lead to the same standard formula:

$$\begin{aligned} \widehat{\beta}_1 &= \frac{\sum_{i=1}^n (X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^n (X_i - \overline{X})^2} \\ &= \frac{\frac{1}{n}\sum_{i=1}^n (X_i - \overline{X})(Y_i - \overline{Y})}{\frac{1}{n}\sum_{i=1}^n (X_i - \overline{X})^2} \\ &= \frac{\widehat{\sigma}_{x,y}}{\widehat{\sigma}_x^2}, \end{aligned}$$

[11] These are Lebesgue integrals with respect to probability measures and conditional probability measures. They include multiple sums and ordinary Reimann integrals as special cases.

where $\widehat{\sigma}_{x,y}$ is the sample covariance between $X$ and $Y$, and $\widehat{\sigma}_x^2$ is the sample variance of $X$. These are maximum likelihood estimates of $Cov(X,Y)$ and $Var(X)$ respectively under the assumption of normality. If the denominators were $n-1$ instead of $n$, they would be unbiased.

By the strong consistency of the sample variance and covariance (see Section A.5 in Appendix A), $\widehat{\sigma}_{x,y}$ converges almost surely to $Cov(X,Y)$ and $\widehat{\sigma}_x^2$ converges almost surely to $Var(X)$ as $n \to \infty$. Under the true model,

$$\begin{aligned} Cov(X,Y) &= Cov(X_i, \beta_0 + \beta_1 X_i + \epsilon_i) \\ &= \beta_1 Cov(X_i, X_i) + Cov(X_i, \epsilon_i) \\ &= \beta_1 \sigma_x^2 + c \end{aligned}$$

So by continuity,

$$\widehat{\beta}_1 = \frac{\widehat{\sigma}_{x,y}}{\widehat{\sigma}_x^2} \xrightarrow{a.s.} \beta_1 + \frac{c}{\sigma_x^2}. \tag{21}$$

Since the estimator is converging to quantity that is off by a fixed amount, it may be called *asymptotically biased*. Thus, while the usual teaching is that sample regression coefficients are unbiased estimators, we see here that $\widehat{\beta}_1$ is biased as $n \to \infty$. Regardless of the true value $\beta_1$, the estimate $\widehat{\beta}_1$ could be absolutely anything, depending on the value of $c$, the covariance between $X_i$ and $\epsilon_i$. The only time $\widehat{\beta}_1$ behaves properly is when $c = 0$.

What's going on here is that the calculation of $\widehat{\beta}_1$ is based on a model that is *misspecified*. That is, it's not the right model. The right model is what we've been calling the *true model*. And to repeat, the true model is the most reasonable model for simple regression, at least for most non-experimental data.

The lesson is this. *When a regression model fails to include all the explanatory variables that contribute to the response variable, and those omitted explanatory variables have non-zero covariance with variables that are in the model, the regression coefficients are inconsistent.* In other words, with more and more data they do not approach the right answer. Instead, they get closer and closer to a specific wrong answer.

If you think about it, this fits with what happens frequently in practical regrssion analysis. When you add a new explanatory variable to a regression equation, the coefficients of the variables that are already in the equation do not remain the same. Almost anything can happen. Positive coefficients can turn negative, negative ones can turn positive, statistical significance can appear where it was previously absent or disappear where it was previously present. Now you know why.

Notice that if the values of one or more explanatory variables are randomly assigned, the random assignment guarantees that these variables are independent of any and all variables that are omitted from the regression equation. Thus, the variables in the equation have zero covariance with those that are omitted, and all the trouble disappears. So, *well-controlled experimental studies are not subject to the kind of problems described here.*

Actually, the calculations in this section support a familiar point, the *correlation-causation* issue, which is often stated more or less as follows. If $A$ and $B$ are related to one another, one cannot necessarily infer that $A$ affects $B$. It could be that $B$ affects $A$,

or that some third variable $C$ is affecting both $A$ and $B$. To this we can now add the possibility that the third variable $C$ affects $B$ and is merely correlated with $A$.

Variables like $C$ are often called *confounding variables*, or more rarely, *lurking variables*. The usual advice is that the only way to completely rule out their action is to randomly assign subjects in the study to the various values of $A$, and then assess the relationship of $A$ to $B$. Again, now you know why.

It should be pointed out that while the correlation-causation issue presents grave obstacles to interpreting the results of observational studies, there is no problem with pure prediction. If you have a data set with $x$ and $y$ values and your interest is predicting $y$ from the $x$ values for a new set of data, a regression equation will be useful, provided that there is a reasonably strong relationship between $x$ and $y$. From the standpoint of prediction, it does not really matter whether $y$ is related to $x$ directly, or indirectly through unmeasured variables that are related to $x$. You have $x$ and not the unmeasured variables, so use it. An example would be an insurance company that seeks to predict the amount of money that you will claim next year (so they can increase your premiums accordingly now). If it turns out that this is predictable from the type of music you download, they will cheerfully use the information, and not care why it works.

Also, the convergence of $\widehat{\beta}_1$ to the wrong answer in (21) may be misleading, but it does not necessarily yield the wrong conclusion. In much of the social and biological sciences, the theories are not detailed and sophisticated enough to make predictions about the actual values of regression coefficients, just whether they should be positive, negative or zero. So, if the variable being tested and the omitted variables are pulling in the same direction (that is, if $\beta_1$ and $c$ in Model (20) on Page 29 are either both positive or both negative), the study will come to the "right" conclusion. The trouble is that you can't tell, because you don't even know what the omitted variables are. All you can do is hope, and that's not a recipe for good science.

**Trying to fit the true model** We have seen that serious trouble arises from adopting a mis-specified model with $c = Cov(X_i, \epsilon_i) = 0$, when in fact because of omitted variables, $c \neq 0$. It is natural, therefore, to attempt estimation and inference for the true model $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ (see Page 29) in the case where $c = Cov(X_i, \epsilon_i)$ need not equal zero. For simplicity, assume that $X_i$ and $\epsilon_i$ have a bivariate normal distribution, so that the observable data pairs $(X_i, Y_i)$ for $i = 1, \ldots, n$ are a random sample from a bivariate normal distribution with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$.

It is straightforward to calculate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from the equation and assumptions of the true model (20). The result is

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = E \begin{pmatrix} X_i \\ Y_i \end{pmatrix} = \begin{pmatrix} \mu_x \\ \beta_0 + \beta_1 \mu_x \end{pmatrix} \tag{22}$$

and

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} = cov \begin{pmatrix} X_i \\ Y_i \end{pmatrix} = \begin{pmatrix} \sigma_x^2 & \beta_1\sigma_x^2 + c \\ \beta_1\sigma_x^2 + c & \beta_1^2\sigma_x^2 + 2\beta_1 c + \sigma_\epsilon^2 \end{pmatrix}. \tag{23}$$

This shows the way in which the parameter vector $\boldsymbol{\theta} = (\mu_x, \sigma_x^2, \beta_0, \beta_1, \sigma_\epsilon^2, c)$ determines $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and hence the probability distribution of the data.

Our primary interest is in $\beta_1$. Because the data pairs $(X_i, Y_i)$ come from a bivariate normal distribution, all you can ever learn from the data are the approximate values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. With larger and larger samples, all you get is better and better approximations of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. That's all there is to know. But even if you knew $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ exactly, could you know $\beta_1$? Formulas (22) and (23) yield a system of five equations in six unknown parameters.

$$\begin{aligned} \mu_1 &= \mu_x \\ \mu_2 &= \beta_0 + \beta_1\mu_x \\ \sigma_{11} &= \sigma_x^2 \\ \sigma_{12} &= \beta_1\sigma_x^2 + c \\ \sigma_{22} &= \beta_1^2\sigma_x^2 + 2\beta_1 c + \sigma_\epsilon^2 \end{aligned} \tag{24}$$

The problem of recovering the parameter values from $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is exactly the problem of solving these five equations in six unknowns. $\mu_x = \mu_1$ and $\sigma_x^2 = \sigma_{11}$ are easy. The remaining 3 equations in 4 unknowns have infinitely many solutions. That is, infinitely many sets of parameter values yield *exactly the same distribution of the sample data*. Distinguishing among them based on sample data is impossible in principle.

To see this in detail, substitute $\mu_1$ for $\mu_x$ and $\sigma_{11}$ for $\sigma_x^2$ in (24), obtaining

$$\begin{aligned} \mu_2 &= \beta_0 + \beta_1\mu_1 \\ \sigma_{12} &= \beta_1\sigma_{11} + c \\ \sigma_{22} &= \beta_1^2\sigma_{11} + 2\beta_1 c + \sigma_\epsilon^2 \end{aligned} \tag{25}$$

Letting the moments $\mu_j$ and $\sigma_{ij}$ remain fixed, we will now write the other parameters as functions of $c$, the covariance between $X_i$ and $\epsilon_i$. Then, moving $c$ will move the other parameters (except for $\mu_x = \mu_1$ and $\sigma_x^2 = \sigma_{11}$), tracing out a one-dimensional subset of the 6-dimensional parameter space where

- All the equations in (24) are satisfied,
- The values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ remain constant, and
- The distribution of $(X_i, Y_i)^\top$ is $N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

First solve for $\beta_1$ in the second equation, obtaining $\beta_1 = \frac{\sigma_{12}-c}{\sigma_{11}}$. Substituting this expression for $\beta_1$ and simplifying, we are able to write all the other model parameters in terms of $c$, as follows.

$$\begin{aligned} \mu_x &= \mu_1 \\ \sigma_x^2 &= \sigma_{11} \\ \beta_0 &= \mu_2 - \mu_1\left(\frac{\sigma_{12}-c}{\sigma_{11}}\right) \\ \beta_1 &= \frac{\sigma_{12}-c}{\sigma_{11}} \\ \sigma_\epsilon^2 &= \sigma_{22} + \frac{c^2-\sigma_{12}^2}{\sigma_{11}} \end{aligned} \tag{26}$$

The parameters $\mu_x$ and $\sigma_x^2$ are constant functions of $c$, while $\beta_0$ and $\beta_1$ are linear functions, and $\sigma_\epsilon^2$ is a quadratic function. The equations (26) define a one-dimensional surface in the six-dimensional parameter space, a kind of curved thread in $\mathbb{R}^6$. Moving $c$ from $-\infty$ to $\infty$ traces out the points on the thread. Importantly, as $c$ ranges from $-\infty$ to $+\infty$ the regression coefficient $\beta_1$ ranges from $+\infty$ to $-\infty$. This means that $\beta_1$ might be positive, it might be negative, or it might be zero. But you really can't tell, because all real values of $\beta_1$ on the surface yield the same population mean and population variance-covariance matrix, and hence the same distribution of the sample data. There is no way to distinguish between the possible values of $\beta_1$ based on sample data.

One technical detail needs to be resolved. Can $c$ really range from $-\infty$ to $\infty$? If not, the possible values of $\beta_1$ would be restricted as well. Two conditions need to be checked. First, the covariance matrix of $(X_i, \epsilon_i)^\top$, like all covariance matrices, has a non-negative determinant. For the bivariate normal density to exist (not a bad assumption), the determinant must be non-zero, and hence it must be strictly positive. Second, $\sigma_\epsilon^2$ must be greater than zero. For points on the thread, the first condition is

$$\begin{aligned} 0 &< \begin{vmatrix} \sigma_x^2 & c \\ c & \sigma_\epsilon^2 \end{vmatrix} \\ &= \sigma_x^2\sigma_\epsilon^2 - c^2 \\ &= \sigma_{11}\left(\sigma_{22} + \frac{c^2 - \sigma_{12}^2}{\sigma_{11}}\right) - c^2 \\ &= \sigma_{11}\sigma_{22} + c^2 - \sigma_{12}^2 - c^2 \\ &= \sigma_{11}\sigma_{22} - \sigma_{12}^2 \\ &= |\boldsymbol{\Sigma}|. \end{aligned}$$

This imposes no restriction on $c$ at all. We also need to check whether $\sigma_\epsilon^2 > 0$ places any restriction on $c$ — for points on the thread, of course.

$$\begin{aligned} & \sigma_\epsilon^2 > 0 \\ \Leftrightarrow\ & \sigma_{22} + \frac{c^2 - \sigma_{12}^2}{\sigma_{11}} > 0 \\ \Leftrightarrow\ & \sigma_{11}\sigma_{22} + c^2 - \sigma_{12}^2 > 0 \\ \Leftrightarrow\ & |\boldsymbol{\Sigma}| + c^2 > 0 \end{aligned}$$

which is true since $|\boldsymbol{\Sigma}| > 0$. Again, the inequality places no restriction on $c$.

Let me beat this point into the ground a bit, because it is important. Since the data are bivariate normal, their probability distribution corresponds uniquely to the pair $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. All you can *ever* learn from *any* set of sample data is the probability distribution from which they come. So all you can ever get from bivariate normal data, no matter what the sample size, is a closer and closer approximation of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. If you cannot find out whether $\beta_1$ is positive, negative or zero from $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, you will *never* be able to make reasonable estimates or inferences about $\beta_1$ from any set of sample data.

What would happen if you tried to estimate the parameters by maximum likelihood? For every $\boldsymbol{\mu} \in \mathbb{R}^2$ and every $2 \times 2$ symmetric positive definite $\boldsymbol{\Sigma}$, there is a surface (thread)

in $\mathbb{R}^6$ defined by (26). This includes $(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$. On that particular thread, the likelihood is highest. Picture a surface with a curvy ridge at the top. The surface has infinitely many maxima, all at the same height, forming a connected set. If you take partial derivatives of the log likelihood and set all six of them equal to zero, there will be infinitely many solutions. If you do numerical maximum likelihood, good software will find a point on the ridge, stop, detect that the surface is not fully concave down there, and complain. Less sophisticated software will just find a point on the ridge, and stop. The stopping place, that is, the maximum likelihood estimate, will depend entirely on where the numerical search starts.

To summarize, if explanatory variables are omitted from a regression equation and those variables have non-zero covariance $c$ with explanatory variables that are *not* omitted, the result is non-zero covariance between explanatory variables and the error term. And, if there is a non-zero covariance between the error term an an explanatory variable in a regression equation, the false assumption that $c = 0$ can easily lead to false results. But allowing $c$ to be non-zero means that infinitely many parameter estimates will be equally plausible, given any set of sample data. In particular, no set of data will be able to provide a basis for deciding whether regression coefficients are positive, negative or zero. The problem is fatal if all you have is $X_i$ and $Y_i$.

The trouble here is lack of parameter identifiability. If a parameter is a function of the distribution of the observable data, it is said to be *identifiable*. The idea is that the parameter is potentially knowable if you knew the distribution of the observable data. If the parameter is not knowable based on the data, they naturally there will be trouble with estimation and inference. Parameter identifiability is a central theme of this book, and will be taken up again in Section 0.9 on Page 58.

## 0.5 Instrumental Variables

The method of instrumental variables was introduced by the economist Phillip Wright in the appendix a 1928 book *The Tariff on Animal and Vegetable Oils* [71]. Phillip Wright was the father of Sewell Wright, the biologist whose work on path analysis led to modern structural equation modeling as well as much of Econometrics. The story is told in a 2003 paper by Stock and Trebbi [62].

An instrumental variable for an explanatory is a variable that is correlated with that explanatory variable, but is not correlated with any error terms or other explanatory variables, and has no direct connection to the response variable. In Econometrics, the instrumental variable usually *influences* the explanatory variable. An instrumental variable is usually not the main focus of attention; it's just a tool.

**Example 0.5.1** *Credit Card Debt*

Suppose we want to know the contribution of income to credit card debt. Because of omitted variables, the model

$$Y_i = \alpha + \beta X_i + \epsilon_i,$$

is guaranteed to fail. Many things influence both income and credit card debt, such as personal style of money management, education, number of children, expenses caused by illness .... The list goes on. As a result, $X_i$ and $\epsilon_i$ have non-zero covariance. The least squares estimate of $\beta$ is inconsistent, and so is every other possible estimate[12]. We can't possibly measure all the variables that affect both income and debt; we don't even know what they all are. Instead, let's add an instrumental variable.

**Definition 0.3** *An instrumental variable for an explanatory variable is another random variable that has non-zero covariance with the explanatory variable, and* no direct connection with any other variable in the model.

Focus the study on real estate agents in many cities, and include median price of resale home for each agent along with income and credit card debt. Median price of resale home qualifies an an instrumental variable according to the definition. Since real estate agents typically receive a percentage of the selling price, it is definitely related to income. Also, housing prices are determined by external economic forces that have little to do with all the personal, individual-level variables that affect the income and debt of individual real estate agents. So, we have the following:

- $W_i$ is median price of resale home in agent $i$'s district.
- $X_i$ is annual income of real estate agent $i$.
- $Y_i$ is agent $i$'s credit card debt.

The model equations are

$$\begin{aligned} X_i &= \alpha_1 + \beta_1 W_i + \epsilon_{i1} \\ Y_i &= \alpha_2 + \beta_2 X_i + \epsilon_{i2}, \end{aligned}$$

and Figure 7 shows the path diagram. The main interest is in $\beta_2$, the link between income and credit card debt. The covariance between $\epsilon_1$ and $\epsilon_2$ represents all the omitted variables that affect both income and credit card debt.

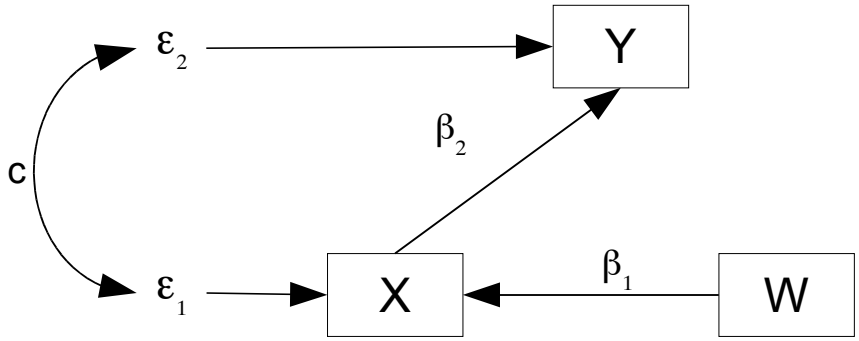
Denoting the expected value of the data vector $\mathbf{D}_i = (W_i, X_i, Y_i)^\top$ by $\boldsymbol{\mu} = [\mu_j]$ and its covariance matrix by $\boldsymbol{\Sigma} = [\sigma_{ij}]$, we have

$$\boldsymbol{\Sigma} = \begin{array}{|c|ccc|} \hline & W & X & Y \\ \hline W & \sigma_w^2 & \beta_1\sigma_w^2 & \beta_1\beta_2\sigma_w^2 \\ X & \cdot & \beta_1^2\sigma_w^2 + \sigma_1^2 & \beta_2(\beta_1^2\sigma_w^2 + \sigma_1^2) + c \\ Y & \cdot & \cdot & \beta_1^2\beta_2^2\sigma_w^2 + \beta_2^2\sigma_1^2 + 2\beta_2 c + \sigma_2^2 \\ \hline \end{array} \tag{27}$$

The lower triangle of the covariance matrix is omitted to make it less cluttered. The notation in (27) is self-explanatory except possibly for $Var(\epsilon_{i1}) = \sigma_1^2$ and $Var(\epsilon_{i2}) = \sigma_2^2$.

[12]This is strictly true if the data are normal. For non-normal data consistent estimation might be possible, but one would have to know the specific non-normal distribution(s).

Figure 7: $W$ is median price of resale home, $X$ is income, $Y$ is credit card debt




It is immediately apparent that the critical parameter $\beta_2$ can be recovered from $\boldsymbol{\Sigma}$ by $\beta_2 = \frac{\sigma_{13}}{\sigma_{12}}$, provided $\beta_1 \neq 0$. A nice Method of Moments estimator in terms of the sample covariances is $\widehat{\beta}_2 = \frac{\widehat{\sigma}_{13}}{\widehat{\sigma}_{12}}$.

The requirement that $\beta_1 \neq 0$ can be verified, by testing $H_0 : \sigma_{12} = 0$ with an elementary test of the correlation between housing prices and income. We expect no problem, because $W$ is a good instrumental variable. Median resale price certainly should be related to the income of real estate agents, and furthermore the relationship is practically guaranteed to be positive. This is a feature of good a instrumental variable. Its relationship to the explanatory variable should be clear, and so obvious that it is hardly worth investigating. The usefulness of the instrumental variable is in the light it casts on relationships that are not so obvious.

In this example, the instrumental variable works beautifully. All the model parameters that appear in $\boldsymbol{\Sigma}$ can be recovered by simple substitution, $\mu_z = \mu_1$, and then $\alpha_1$ and $\alpha_2$ can be recovered from $\mu_2 = E(X_i)$ and $\mu_3 = E(Y_i)$ respectively. The function from $(\alpha_1, \alpha_2, \beta_1, \beta_2, \mu_w, \sigma_w^2, \sigma_1^2, \sigma_2^2, c)$ to $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is one-to one. Method of Moments estimates are readily available, and they are consistent by the continuity of the functions involved. Under the additional assumption of multivariate normality, the Method of Moments estimates are also maximum likelihood by the invariance principle.

To test the central null hypothesis $H_0 : \beta_2 = 0$, fancy software is not required. Since we have concluded with high confidence that $\beta_1 > 0$, the covariance $\sigma_{13}$ equals zero if and only if $\beta_2 = 0$, and the sign of $\sigma_{13}$ is the same as the sign of $\beta_2$. So it is necessary only to test the correlation between housing price and real estate agents' credit card debt. Under the normal assumption, the usual test is exact, and a large sample is not required. If the normal assumption is worrisome, the non-parametric test associated with the Spearman rank correlation coefficient is a permutation test carried out on ranks, and an exact small-sample $p$-value is available even though some software produces a large-sample approximation by default.

The instrumental variable method saved the day in this example, but it does not solve

the problem of omitted variables in every case, or even in most cases. This is because good instrumental variables are not easy to find. They will not just happen to be in the data set, except by a miracle. They really have to come from another universe, and still have a strong, clear connection to the explanatory variable. Data collection has to be *planned*, with a model that admits the existence of omitted variables explicitly in mind.

**Measurement Error** All models are inexact representations of reality, but I must admit that the model in Figure 7 is seriously wrong. Our interest is in how *true* income affects *true* credit card debt. But these variables are not observed. What we have in the data file are *reported* income and *reported* credit card debt. For various reasons that the reader can easily supply, what people report about financial details is not the same thing as the truth. When we record median price of a resale home, that's unlikely to be perfectly accurate either. As we will see later in this chapter, measurement error in the explanatory variables presents serious problems for regression analysis in general. We will also see that instrumental variables can help with measurement error as well as with omitted variables, but first it is helpful to introduce the topic of measurement error in an organized way.

## 0.6 The Idea of Measurement Error

In a survey, suppose that a respondent's annual income is "measured" by simply asking how much he or she earned last year. Will this measurement be completely accurate? Of course not. Some people will lie, some will forget and give a reasonable guess, and still others will suffer from legitimate confusion about what constitutes income. Even physical variables like height, weight and blood pressure are subject to some inexactness of measurement, no matter how skilled the personnel doing the measuring. In fact, very few of the variables in the typical data set are measured completely without error.

One might think that for experimentally manipulated variables like the amount of drug administered in a biological experiment, laboratory procedures would guarantee that for all practical purposes, the amount of drug a subject receives is exactly what you think it is. But Alison Fleming (University of Toronto Psychology department) pointed out to me that when hormones are injected into a laboratory rat, the amount injected is exactly right, but due to tiny variations in needle placement, the amount actually reaching the animal's bloodstream can vary quite a bit. The same thing applies to clinical trials of drugs with humans. We will see later, though, that the statistical consequences of measurement error are not nearly as severe with experimentally manipulated variables, assuming the study is well-controlled in other respects.

Random variables that cannot be directly observed are called *latent variables*. The ones we can observe are sometimes called "manifest," but here they will be called "observed" or "observable," which is also a common usage. Upon reflection, it is clear that most of the time, we are interested in relationships among latent variables, but at best our data consist only of their imperfect, observable counterparts. One is reminded of the allegory of the cave in Plato's *Republic* [46], where human beings are compared to prisoners in a

cave, with their heads chained so that they can only look at a wall. Behind them is a fire, which casts flickering shadows on the wall. They cannot observe reality directly; all they can see are the shadows.

## A simple additive model for measurement error

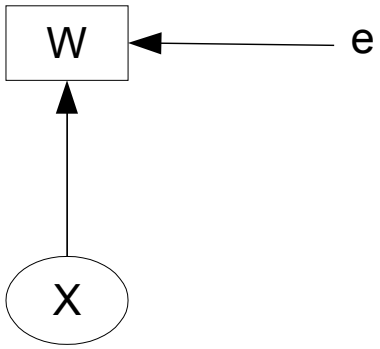
Measurement error can take many forms. For categorical variables, there is *classification error*. Suppose a data file indicates whether or not each subject in a study has ever had a heart attack. Clearly, the latent Yes-No variable (whether the person has *truly* had a heart attack) does not correspond perfectly to what is in the data file, no matter how careful the assessment is. Mis-classification can and does occur, in both directions.

Here, we will put classification error aside for now because it is technically difficult, and focus on a very simple form of measurement error that applies to continuous variables. There is a latent random variable $X$ that cannot be observed, and a little random shock $e$ that pushes $X$ up or down, producing an observable random variable $W$. That is,

$$W = X + e \tag{28}$$

Let's say $E(X) = \mu_x$, $E(e) = 0$, $Var(X) = \sigma_x^2$, $Var(e) = \sigma_e^2$, and $Cov(X, e) = 0$. Figure 8 is a path diagram of this model.

Figure 8: Additive Measurement Error




Because $X$ and $e$ are uncorrelated,

$$Var(W) = Var(X) + Var(e) = \sigma_x^2 + \sigma_e^2.$$

Variance is an index of unit-to unit variation in a measurement. The simple calculation above reveals that variation in the observable variable has two sources: variation in the actual quantity of interest, and variation in the magnitude of the random shocks that create error in measurement. To judge the quality of a measurement $W$, it is important to assess how much of its variance comes from variation in the true quantity of interest, and how much comes from random noise.

In psychometric theory[13], the *reliability*[14] of a measurement is defined as the squared correlation of the true score with the observed score. Here the "true score" is $X$ and the "observed score" is $W$. The reliability of the measurement $W$ is

$$\begin{aligned}\rho^2 &= \left(\frac{Cov(X,W)}{SD(X)SD(W)}\right)^2 \\ &= \left(\frac{\sigma_x^2}{\sqrt{\sigma_x^2}\sqrt{\sigma_x^2+\sigma_e^2}}\right)^2 \\ &= \frac{\sigma_x^4}{\sigma_x^2(\sigma_x^2+\sigma_e^2)} \\ &= \frac{\sigma_x^2}{\sigma_x^2+\sigma_e^2}. \end{aligned} \tag{29}$$

That is, the reliability of a measurement is the proportion of the measurement's variance that comes from the true quantity being measured, rather than from measurement error[15].

A reliability of one means there is no measurement error at all, while a reliability of zero means the measurement is pure noise. In the social sciences, reliabilities above 0.9 could be called excellent, from 0.8 to 0.9 good, and from 0.7 to 0.8 acceptable. Frequently, responses to single questions have reliabilities that are much less than this. To see why reliability depends on the number of questions that measure the latent variable, see Exercise 6 at the end of this section.

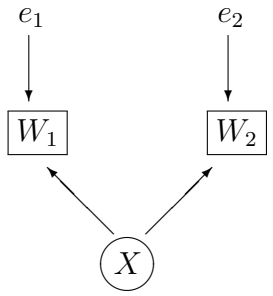
Since reliability represents quality of measurement, estimating it is an important goal. Using the definition directly is seldom possible. Reliability is the squared correlation between a latent variable and its observable counterpart, but by definition, values of the latent variable cannot be observed. On rare occasions and perhaps with great expense, it may be possible to obtain perfect or near-perfect measurements on a subset of the sample; the term *gold standard* is sometimes applied to such measurements. In that case, the reliability of the usual measurement can be estimated by a squared sample correlation between the usual measurement and the gold standard measurement. But even measurements that are called gold standard are seldom truly free of measurement error. Consequently, reliabilities that are estimated by correlating imperfect gold standards and ordinary measurements are biased downward: See Exercise 4 at the end of this section. It is clear that another approach is needed.

[13]Psychometric theory is the statistical theory of psychological measurement. The bible of psychometric theory is Lord and Novick's (1968) classic *Statistical theories of mental test scores* [44]. It is not too surprising that measurement error would be acknowledged and studied by psychologists. A large sector of psychological research employs "measures" of hypothetical constructs like neuroticism or intelligence (mostly paper-and-pencil tests), but no sensible person would claim that true value of such a trait is exactly the score on the test. It's true there is a famous quote "Intelligence is whatever an intelligence test measures." I have tried unsuccessfully to track down the source of this quote, and I now suspect that it is just an illustration of a philosophic viewpoint called Logical Positivism (which is how I first heard it), and not a serious statement about intelligence measurement.

[14]In survival analysis and statistical quality control, reliability means something entirely different.

[15]It's like the proportion of variance in the response variable explained by a regression, except that here the explanatory variable is the latent true score. Compare Expression (11) on Page 19.

Figure 9: Two independent measurements of a latent variable




**Test-retest reliability** Suppose that it is possible to make the measurement of $W$ twice, in such a way that the errors of measurement are independent on the two occasions. We have

$$\begin{aligned} W_1 &= X + e_1 \\ W_2 &= X + e_2, \end{aligned}$$

where $E(X) = \mu_x$, $Var(X) = \sigma_x^2$, $E(e_1) = E(e_2) = 0$, $Var(e_1) = Var(e_2) = \sigma_e^2$, and $X$, $e_1$ and $e_2$ are all independent. Because $Var(e_1) = Var(e_2)$, $W_1$ and $W_2$ are called *equivalent measurements*. That is, they are contaminated by error to the same degree. Figure 9 is a path diagram of this model.

It turns out that the correlation between $W_1$ and $W_2$ is exactly equal to the reliability, and this opens the door to reasonable methods of estimation. The calculation (like many in this book) is greatly simplified by using the rule for covariances of linear combinations (1) on Page 9.

$$\begin{aligned} Corr(W_1, W_2) &= \frac{Cov(W_1, W_2)}{SD(W_1)SD(W_2)} \\ &= \frac{Cov(X + e_1, X + e_2)}{\sqrt{\sigma_x^2 + \sigma_e^2}\sqrt{\sigma_x^2 + \sigma_e^2}} \\ &= \frac{Cov(X, X) + Cov(X, e_2) + Cov(e_1, X) + Cov(E_1, e_2))}{\sigma_x^2 + \sigma_e^2} \\ &= \frac{Var(X) + 0 + 0 + 0}{\sigma_x^2 + \sigma_e^2} \\ &= \frac{\sigma_x^2}{\sigma_x^2 + \sigma_e^2}, \end{aligned} \tag{30}$$

which is the reliability.

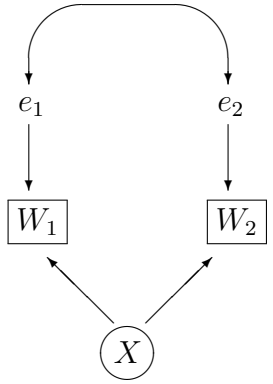
The calculation above is the basis of *test-retest reliability*[16], in which the reliability of a measurement such as an educational or psychological test is estimated by the sample

[16]Closely related to test-retest reliability is *alternate forms reliability*, in which you correlate two

correlation between two independent administrations of the test. That is, the test is given twice to the same sample of individuals, ideally with a short enough time between tests so that the trait does not really change, but long enough apart so they forget how they answered the first time.

**Correlated measurement error** Suppose participants remembered their wrong answers or lucky guesses from the first time they took a test, and mostly gave the same answer the second time. The result would be a positive correlation between the measurement errors $e_1$ and $e_2$. Omitted variables (see Section 0.4) like level of test anxiety for educational tests or desire to make a favourable impression for attitude questionnaires can also produce a positive covariance between errors of measurement. Whatever the source, positive covariance between $e_1$ and $e_2$ is an additional source of positive covariance between $W_1$ and $W_2$ that does *not* come from the latent variable $X$ being measured. The result is an inflated estimate of reliability and an unduly rosy picture of the quality of measurement. Figure 10 shows this situation.

Figure 10: Correlated Measurement Error




We will return more than once to the issue of correlated errors of measurement. For now, just notice how careful planning of the data collection (in this case, the time lag between the two administrations of the test) can eliminate or at least reduce the correlation between errors of measurement. In general, the best way to take care of correlated measurement error is with good research design[17].

**Sample Test-retest Reliability** Again, suppose it is possible to measure a variable of interest twice, in such a way that the errors of measurement are uncorrelated and have

equivalent versions of the test. In *split-half reliability*, you split the items of the test into two equivalent subsets and correlate them. There are also *internal consistency* estimates of reliability based on correlations among items. Assuming independent errors of measurement for split half reliability and internal consistency reliability is largely a fantasy, because both measurements are affected in the same way by short-term situational influences like mood, amount of sleep the night before, noise level, behaviour of the person administering the test, and so on.

[17]Indeed, one could argue that most principles of good research design are methods for minimizing the variance and covariance of measurement errors.

equal variance. Then the reliability may be estimated by doing this for a random sample of individuals. Let $X_1, \ldots, X_n$ be a random sample of latent variables (true scores), with $E(X_i) = \mu$ and $Var(X_i) = \sigma_x^2$. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} W_{i,1} &= X_i + e_{i,1} \\ W_{i,2} &= X_i + e_{i,2}, \end{aligned}$$

where $E(e_{i,1}) = E(e_{i,2}) = 0$, $Var(e_{i,1}) = Var(e_{i,2}) = \sigma_e^2$, and $X_i$, $e_{i,1}$ and $e_{i,2}$ are all independent for $i = 1, \ldots, n$. Then the sample correlation between the pairs of measurements is

$$\begin{aligned} R_n &= \frac{\sum_{i=1}^n (W_{i,1} - \overline{W}_1)(W_{i,2} - \overline{W}_2)}{\sqrt{\sum_{i=1}^n (W_{i,1} - \overline{W}_1)^2}\sqrt{\sum_{i=1}^n (W_{i,2} - \overline{W}_2)^2}} \\ &= \frac{\frac{1}{n}\sum_{i=1}^n (W_{i,1} - \overline{W}_1)(W_{i,2} - \overline{W}_2)}{\sqrt{\frac{1}{n}\sum_{i=1}^n (W_{i,1} - \overline{W}_1)^2}\sqrt{\frac{1}{n}\sum_{i=1}^n (W_{i,2} - \overline{W}_2)^2}} \\ &\stackrel{a.s.}{\to} \frac{Cov(W_{i,1}, W_{i,2})}{\sqrt{Var(W_{i,1})}\sqrt{Var(W_{i,2})}} \\ &= \frac{\sigma_x^2}{\sigma_x^2 + \sigma_e^2} \\ &= \rho^2, \end{aligned}$$

where the convergence follows from continuous mapping and the fact that sample variances and covariances are strongly consistent estimators of the corresponding population quantities; see Section A.5.2 in Appendix A. The conclusion is that $R_n$ is a strongly consistent estimator of the reliability. That is, for a large enough sample size, $R_n$ will get arbitrarily close to the true reliability, and this happens with probability one.

## 0.7 Ignoring measurement error

Standard regression models make no provision at all for measurement error, so when we apply such models to real data, we are effectively ignoring any measurement error that may be present; we are pretending it's not there. This section will show that the result can be a real disaster, featuring incorrect estimates of regression parameters and Type I error probabilities approaching one as the sample size increases. Much of this material, including the history of the topic (warnings go back to at least 1936) can be found in a 2009 paper by Brunner and Austin [14].

### Measurement error in the response variable

While ignoring measurement error in the explanatory variables can have very bad consequences, it turns out that under some conditions, measurement error in the response variable is a less serious problem.

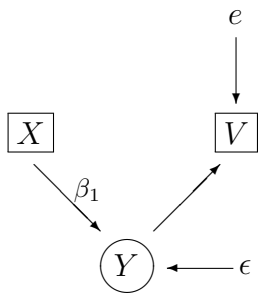
**Example 0.7.1** *Measurement Error in Y Only*

Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_i + \epsilon_i \\ V_i &= \nu + Y_i + e_i, \end{aligned}$$

where $Var(X_i) = \sigma_x^2$, $Var(e_i) = \sigma_e^2$, $Var(\epsilon_i) = \sigma_\epsilon^2$, and $X_i, e_i, \epsilon_i$ are all independent. Figure 11 is a path diagram of this model.

Figure 11: Measurement error in the response variable




In Example 0.7.1, the explanatory variable $X_i$ is observable, but the response variable $Y_i$ is latent. Instead of $Y_i$, we can see $V_i$, which is $Y_i$ plus a piece of random noise, and also plus a constant $\nu$ that represents the difference between the expected value of the latent random variable and the expected value of its observable counterpart. This constant term could be called *measurement bias*. For example, if $Y$ is true amount of exercise in minutes and $V$ is reported exercise, the measurement bias $\nu$ is population mean exaggeration, in minutes.

Since $Y_i$ cannot be observed, $V_i$ is used in its place, and the data analyst fits the *naive* model

$$V_i = \beta_0 + \beta_1 X_i + \epsilon_i.$$

**Studying Mis-specified Models** The "naive model" above is an example of a model that is *mis-specified*. That is, the model says that the data are being generated in a particular way, but this is not how the data are actually being produced. Generally speaking, correct models will usually yield better results than incorrect models, but it's not that simple. In reality, most statistical models are imperfect. The real question is how much any given imperfection really matters. As Box and Draper (1987, p. 424) put it, "Essentially all models are wrong, but some are useful." [11]

So, it is not enough to complain that a statistical model is incorrect, or unrealistic. To make the point convincingly, one must show that being wrong in a particular way causes the model to yield misleading results. To do this, it is necessary to have a specific *true model* in mind; typically the so-called true model is one that is obviously more believable than the model being challenged. Then, one can examine estimators or test

statistics based on the mis-specified model, and see how they behave when the true model holds. We have already done this in Section 0.4 in connection with omitted variables; see Example 0.4.1 starting on Page 28.

Under the true model of Example 0.7.1 (measurement error in the response variable only), we have $Cov(X_i, V_i) = \beta_1\sigma_x^2$ and $Var(X_i) = \sigma_x^2$. Then,

$$\begin{aligned}
\widehat{\beta}_1 &= \frac{\sum_{i=1}^{n}(X_i - \overline{X})(V_i - \overline{V})}{\sum_{i=1}^{n}(X_i - \overline{X})^2} \\
&= \frac{\widehat{\sigma}_{x,v}}{\widehat{\sigma}_x^2} \\
&\stackrel{a.s.}{\to} \frac{Cov(X_i, V_i)}{Var(X_i)} \\
&= \frac{\beta_1\sigma_x^2}{\sigma_x^2} \\
&= \beta_1.
\end{aligned} \tag{31}$$

Even when the model is mis-specified by assuming that the response variable is measured without error, the ordinary least squares estimate of the slope is consistent. There is a general lesson here about mis-specified models. Mis-specification (using the wrong model) is not always a problem; sometimes everything works out fine.

Let's see why the naive model works so well here. The response variable under the true model may be re-written

$$\begin{aligned}
V_i &= \nu + Y_i + e_i \\
&= \nu + (\beta_0 + \beta_1 X_i + \epsilon_i) + e_i \\
&= (\nu + \beta_0) + \beta_1 X_i + (\epsilon_i + e_i) \\
&= \beta_0' + \beta_1 X_i + \epsilon_i'
\end{aligned} \tag{32}$$

What has happened here is a *re-parameterization* (not a one-to-one re-parameterization), in which the pair $(\nu, \beta_0)$ is absorbed into $\beta_0'$, and $Var(\epsilon_i + e_i) = \sigma_\epsilon^2 + \sigma_e^2$ is absorbed into a single unknown variance that will probably be called $\sigma^2$. It is true that $\nu$ and $\beta_0$ will never be knowable separately, and also $\sigma_\epsilon^2$ and $\sigma_e^2$ will never be knowable separately. But that really doesn't matter, because the true interest is in $\beta_1$.

In this book and in standard statistical practice, there are many models where the response variable appears to be measured without error. But error-free measurement is a rarity at best, so these models should be viewed as re-parameterized versions of models that do acknowledge the reality of measurement error in the response variable. A critical feature of these re-parameterized models is that the measurement error is assumed independent of everything else in the model. When this fails, there is usually trouble.

## Measurement error in the explanatory variables

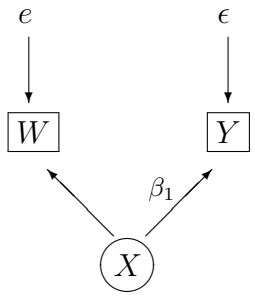
**Example 0.7.2** *Measurement error in a single explanatory variable*

Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_i + \epsilon_i \\ W_i &= X_i + e_i, \end{aligned}$$

where $Var(X_i) = \sigma_x^2$, $Var(e_i) = \sigma_e^2$, $Var(\epsilon_i) = \sigma_\epsilon^2$, and $X_i, e_i, \epsilon_i$ are all independent. Figure 12 is a path diagram of the model.

Figure 12: Measurement error in the explanatory variable




Unfortunately, the explanatory variable $X_i$ cannot be observed; it is a latent variable. So instead $W_i$ is used in its place, and the data analyst fits the naive model

$$Y_i = \beta_0 + \beta_1 W_i + \epsilon_i.$$

Under the naive model of Example 0.7.2, the ordinary least squares estimate of $\beta_1$ is

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (W_i - \overline{W})(Y_i - \overline{Y})}{\sum_{i=1}^n (W_i - \overline{W})^2} = \frac{\widehat{\sigma}_{w,y}}{\widehat{\sigma}_w^2}.$$

Regardless of what model is correct, $\widehat{\sigma}_{w,y} \stackrel{a.s.}{\to} Cov(W,Y)$ and $\widehat{\sigma}_w^2 \stackrel{a.s.}{\to} Var(W)$[18], so that by the continuous mapping property of ordinary limits[19], $\widehat{\beta}_1 \stackrel{a.s.}{\to} \frac{Cov(W,Y)}{Var(W)}$.

Let us assume that the true model holds. In that case,

$$Cov(W,Y) = \beta_1 \sigma_x^2 \quad \text{and} \quad Var(W) = \sigma_x^2 + \sigma_e^2.$$

[18]This is true because sample variances and covariances are strongly consistent estimators of the corresponding population quantities; see Section A.5.2 in Appendix A.

[19]Almost sure convergence acts like an ordinary limit, applying to all points in the underlying sample space, except possibly a set of probability zero. If you wanted to do this problem strictly in terms of convergence in probability, you could use the Weak Law of Large Numbers and then use Slutsky Lemma 7a of Appendix A.5.

Consequently,

$$\begin{aligned}
\widehat{\beta}_1 &= \frac{\sum_{i=1}^{n}(W_i - \overline{W})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(W_i - \overline{W})^2} \\
&= \frac{\widehat{\sigma}_{w,y}}{\widehat{\sigma}_w^2} \\
&\stackrel{a.s.}{\rightarrow} \frac{Cov(W,Y)}{Var(W)} \\
&= \beta_1 \left( \frac{\sigma_x^2}{\sigma_x^2 + \sigma_e^2} \right). \qquad (33)
\end{aligned}$$

So when the fuzzy explanatory variable $W_i$ is used instead of the real thing, $\widehat{\beta}_1$ converges not to the true regression coefficient, but to the true regression coefficient multiplied by the reliability of $W_i$. That is, it's biased, even as the sample size approaches infinity. It is biased toward zero, because reliability is between zero and one. The worse the measurement of $X$, the more the asymptotic bias.

What happens to $\widehat{\beta}_1$ in (33) is sometimes called *attenuation*, or weakening, and in this case that's what happens. The measurement error weakens the apparent relationship between $X_1$ and $Y$. If the reliability of $W$ can be estimated from other data (and psychologists are always trying to estimate reliability), then the sample regression coefficient can be "corrected for attentuation." Sample correlation coefficients are sometimes corrected for attenuation too.

Now typically, social and biological sientists are not really interested in point estimates of regression coefficients. They only need to know whether the coefficients are positive, negative or zero. So the idea of attenuation sometimes leads to a false sense of security. It's natural to over-generalize from the case of one explanatory variables, and think that measurement error just weakens what's really there. Therefore, the reasoning goes, if you can reject the null hypothesis and conclude that a relationship is present even with measurement error, you would have reached the same conclusion if the explanatory variables had not been measured with error.

Unfortunately, it's not so simple. With two or more explanatory variables the effects of measurement error are far more serious and potentially misleading.

## Measurement error in more than one explanatory variable

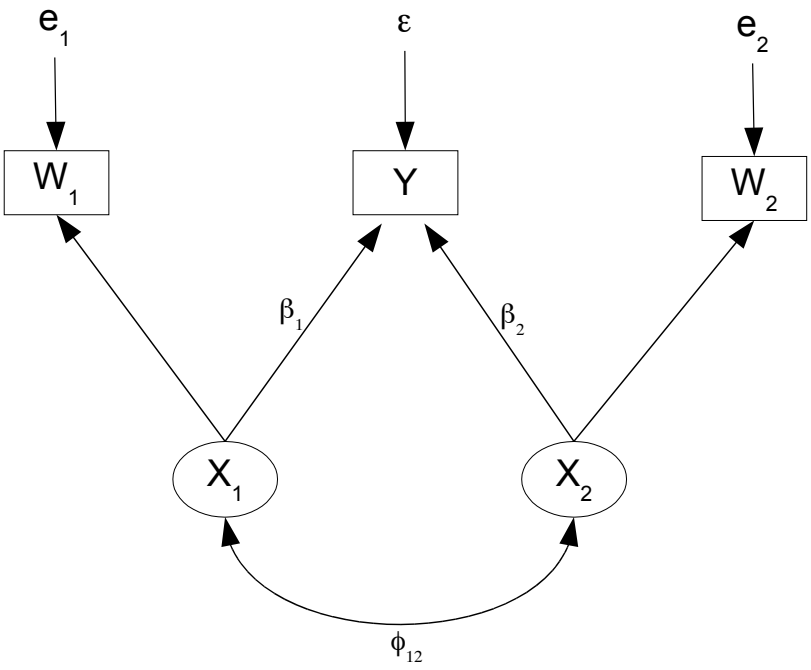
In this example, there are two explanatory variables, both measured with error.

**Example 0.7.3** *Measurement Error in Two Explanatory Variables*

Independently for $i = 1, \ldots, n$,

$$\begin{aligned}
Y_i &= \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i \\
W_{i,1} &= X_{i,1} + e_{i,1} \\
W_{i,2} &= X_{i,2} + e_{i,2},
\end{aligned}$$

Figure 13: Two explanatory variables measured with error




where $E(X_{i,1}) = \mu_1$, $E(X_{i,2}) = \mu_2$, $E(\epsilon_i) = E(e_{i,1}) = E(e_{i,2}) = 0$, $Var(\epsilon_i) = \psi$, $Var(e_{i,1}) = \omega_1$, $Var(e_{i,2}) = \omega_2$, the errors $\epsilon_i, e_{i,1}$ and $e_{i,2}$ are all independent, $X_{i,1}$ is independent of $\epsilon_i, e_{i,1}$ and $e_{i,2}$, $X_{i,2}$ is independent of $\epsilon_i, e_{i,1}$ and $e_{i,2}$, and

$$cov \begin{pmatrix} X_{i,1} \\ X_{i,1} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}.$$

Figure 13 shows the path diagram.

Again, because the actual explanatory variables $X_{i,1}$ and $X_{i,2}$ are latent variables that cannot be observed, $W_{i,1}$ and $W_{i,2}$ are used in their place. The data analyst fits the naive model

$$Y_i = \beta_0 + \beta_1 W_{i,1} + \beta_2 W_{i,2} + \epsilon_i.$$

An attractive feature of multiple regression is its ability to represent the relationship of one or more explanatory variables to the response variable, while *controlling for* other explanatory varables. In fact, this may be the biggest appeal of multiple regression and similar methods for non-experimental data. In Example 0.7.3, our interest is in the relationship of $X_2$ to $Y$ controlling for $X_1$. The main objective is to test $H_0 : \beta_2 = 0$, but we are also interested in the estimation of $\beta_2$.

The argument that follows illustrates a general way to see what happens as $n \to \infty$ for mis-specified (that is, incorrect) regression models. We have already seen special cases of this, three times. In Example 0.4.1 on omitted explanatory variables, the regression coefficient converged to the wrong target in Expression 21 on page 32. In Example 0.7.1

on measurement error in the response variable, the regression coefficient converged to the correct target in Expression 31 on page 46. In Example 0.7.2 on measurement error in a single explanatory variable, the regression coefficient converged to the target multiplied by the reliability of the measurement, in Expression 33 on page 48.

Here is the recipe. Assume some "true" model for how the data are produced, and a mis-specified model corresponding to a natural way that people would analyze the data with a regression model. First, write the regression coefficients in terms of sample variances and covariances. The general answer is given on page 14: $\widehat{\boldsymbol{\beta}}_n = \widehat{\boldsymbol{\Sigma}}_x^{-1}\widehat{\boldsymbol{\Sigma}}_{xy}$. Then, because sample variances and covariances are consistent estimators of their population counterparts, we have the convergence $\widehat{\boldsymbol{\beta}}_n \stackrel{a.s.}{\to} \boldsymbol{\Sigma}_x^{-1}\boldsymbol{\Sigma}_{xy}$ from Page 15. This convergence follows from the formula for the least-squares estimator, and does not depend in any way on the correctness of the model. So, if you can derive $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_{xy}$ under the true model, it is easy to calculate the large-sample target of the ordinary least squares estimates under the mis-specified model.

In the present application, there is just a minor notational issue. Under the naive model, the explanatory variables are called $w$ instead of $x$. Adopting a notation that will be used throughout the book, denote one of the $n$ vectors of observable data by $\mathbf{D}_i$. Here,

$$\mathbf{D}_i = \begin{pmatrix} W_{i,1} \\ W_{i,2} \\ Y_i \end{pmatrix}.$$

Then, let $\boldsymbol{\Sigma} = [\sigma_{i,j}] = cov(\mathbf{D}_i)$. Corresponding to $\boldsymbol{\Sigma}$ is the sample variance covariance matrix $\widehat{\boldsymbol{\Sigma}} = [\widehat{\sigma}_{i,j}]$, with $n$ rather than $n-1$ in the denominators. To make this setup completely explicit,

$$\boldsymbol{\Sigma} = cov\begin{pmatrix} W_{i,1} \\ W_{i,2} \\ Y_i \end{pmatrix} = \begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} & \sigma_{1,3} \\ \sigma_{1,2} & \sigma_{2,2} & \sigma_{2,3} \\ \sigma_{1,3} & \sigma_{2,3} & \sigma_{3,3} \end{pmatrix}$$

Then, we calculate the regression coefficients under the naive model.

$$\begin{aligned} \widehat{\boldsymbol{\beta}}_n &= \begin{pmatrix} \widehat{\beta}_1 \\ \widehat{\beta}_2 \end{pmatrix} \\ &= \widehat{\boldsymbol{\Sigma}}_w^{-1}\widehat{\boldsymbol{\Sigma}}_{wy} \\ &= \begin{pmatrix} \widehat{\sigma}_{1,1} & \widehat{\sigma}_{1,2} \\ \widehat{\sigma}_{1,2} & \widehat{\sigma}_{2,2} \end{pmatrix}^{-1} \begin{pmatrix} \widehat{\sigma}_{1,3} \\ \widehat{\sigma}_{2,3} \end{pmatrix} \\ &= \begin{pmatrix} \frac{\widehat{\sigma}_{22}\widehat{\sigma}_{13} - \widehat{\sigma}_{12}\widehat{\sigma}_{23}}{\widehat{\sigma}_{11}\widehat{\sigma}_{22} - \widehat{\sigma}_{12}^2} \\ \\ \frac{\widehat{\sigma}_{11}\widehat{\sigma}_{23} - \widehat{\sigma}_{12}\widehat{\sigma}_{13}}{\widehat{\sigma}_{11}\widehat{\sigma}_{22} - \widehat{\sigma}_{12}^2} \end{pmatrix}. \end{aligned} \tag{34}$$

Our primary interest is in the estimation of $\beta_2$. Because sample variances and covariances are strongly consistent estimators of the corresponding population quantities,

$$\widehat{\beta}_2 = \frac{\widehat{\sigma}_{11}\widehat{\sigma}_{23} - \widehat{\sigma}_{12}\widehat{\sigma}_{13}}{\widehat{\sigma}_{11}\widehat{\sigma}_{22} - \widehat{\sigma}_{12}^2} \xrightarrow{a.s.} \frac{\sigma_{11}\sigma_{23} - \sigma_{12}\sigma_{13}}{\sigma_{11}\sigma_{22} - \sigma_{12}^2}. \tag{35}$$

This convergence holds provided that the denominator $\sigma_{11}\sigma_{22} - \sigma_{12}^2 \neq 0$. The denominator is a determinant:

$$\sigma_{11}\sigma_{22} - \sigma_{12}^2 = \left| cov \begin{pmatrix} W_{i,1} \\ W_{i,2} \end{pmatrix} \right|.$$

It will be non-zero provided at least one of

$$cov \begin{pmatrix} X_{i,1} \\ X_{i,2} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix} \quad \text{and} \quad cov \begin{pmatrix} e_{i,1} \\ e_{i,2} \end{pmatrix} = \begin{pmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{pmatrix}$$

is positive definite – not a lot to ask.

The convergence of $\widehat{\beta}_2$ in Expression 35 applies regardless of what model is correct. To see what happens when the true model of Example 0.7.3 holds, we need to write the $\sigma_{ij}$ quantities in terms of the parameters of the true model. A straightforward set of scalar variance-covariance calculations yields

$$\begin{aligned}
\boldsymbol{\Sigma} &= cov \begin{pmatrix} W_{i,1} \\ W_{i,2} \\ Y_i \end{pmatrix} \\
&= \begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} & \sigma_{1,3} \\ \sigma_{1,2} & \sigma_{2,2} & \sigma_{2,3} \\ \sigma_{1,3} & \sigma_{2,3} & \sigma_{3,3} \end{pmatrix} \\
&= \begin{pmatrix} \omega_1 + \phi_{11} & \phi_{12} & \beta_1\phi_{11} + \beta_2\phi_{12} \\ \phi_{12} & \omega_2 + \phi_{22} & \beta_1\phi_{12} + \beta_2\phi_{22} \\ \beta_1\phi_{11} + \beta_2\phi_{12} & \beta_1\phi_{12} + \beta_2\phi_{22} & \beta_1^2\phi_{11} + 2\,\beta_1\beta_2\phi_{12} + \beta_2^2\phi_{22} + \psi \end{pmatrix}
\end{aligned}$$

Subsituting into expression 35 and simplifying[20], we obtain

$$\begin{aligned}
\widehat{\beta}_2 &= \frac{\widehat{\sigma}_{11}\widehat{\sigma}_{23} - \widehat{\sigma}_{12}\widehat{\sigma}_{13}}{\widehat{\sigma}_{11}\widehat{\sigma}_{22} - \widehat{\sigma}_{12}^2} \\
&\xrightarrow{a.s.} \frac{\sigma_{11}\sigma_{23} - \sigma_{12}\sigma_{13}}{\sigma_{11}\sigma_{22} - \sigma_{12}^2} \\
&= \frac{\beta_1\omega_1\phi_{12} + \beta_2(\omega_1\phi_{22} + \phi_{11}\phi_{22} - \phi_{12}^2)}{(\phi_{1,1} + \omega_1)(\phi_{2,2} + \omega_2) - \phi_{12}^2} \\
&= \beta_2 + \frac{\beta_1\omega_1\phi_{12} + \beta_2\omega_2(\phi_{11} - \omega_1)}{(\phi_{1,1} + \omega_1)(\phi_{2,2} + \omega_2) - \phi_{12}^2}
\end{aligned} \tag{36}$$

By the asymptotic normality of sample variances and covariances and the multivariate delta method (see Appendix A.5), $\widehat{\beta}_2$ has a distribution that is approximately normal for

[20] The simplification may be elementary, but that does not make it easy. I used Sage; see Appendix B.

large samples, with approximate mean given by expression (36). Thus, it makes sense to call the second term in (36) the *asymptotic bias*. It is also the amount by which the estimate of $\beta_2$ will be wrong as $n \to \infty$.

Clearly, this situation is much more serious than the bias toward zero detected for the case of one explanatory variable. With two explanatory variables, the bias can be positive, negative or zero depending on the values of other unknown parameters.

In particular, consider the problems associated with testing $H_0 : \beta_2 = 0$. The purpose of this test is to determine whether, controlling for $X_1$, $X_2$ has any relationship to $Y$. The supposed ability of multiple regression to answer questions like this is the one of the main reasons it is so widely used in practice. So when measurement error makes this kind of inference invalid, it is a real problem.

Suppose that the null hypothesis is true, so $\beta_2 = 0$. In this case, Expression (36) becomes

$$\widehat{\beta}_2 \stackrel{a.s.}{\to} \frac{\beta_1\omega_1\phi_{12}}{(\phi_{1,1}+\omega_1)(\phi_{2,2}+\omega_2)-\phi_{12}^2}. \tag{37}$$

Recall that $\beta_1$ is the link between $X_1$ and $Y$, $\omega_1 = Var(e_1)$ is the variance of measurement error in $X_1$, and $\phi_{12}$ is the covariance between $X_1$ and $X_2$. Thus, when $H_0 : \beta_2 = 0$ is true, $\widehat{\beta}_2$ converges to a non-zero quantity unless

- There is no relationship between $X_1$ and $Y$, or
- There is no measurement error in $W_1$, or
- There is no correlation between $X_1$ and $X_2$.

Brunner and Austin [14] have shown that whether $H_0$ is true or not, the standard error of $\widehat{\beta}_2$ goes to zero, and when the large-sample target of $\widehat{\beta}_2$ is non-zero, the $p$-value goes almost surely to zero. That is, the probability of making a Type I error goes to one because of measurement error in an explanatory variable — not the one being tested, but the one for which one is "controlling."

This is potentially a disaster, because the primary function of statistical hypothesis testing in the social and biological sciences is to filter out results that might be just random noise, and keep them from reaching the published research literature. Holding down the probability of a Type I error is critical. The preceding calculations show that in the very reasonable scenario where one needs to control for an explanatory variable but the measurement of that variable is imperfect (which is always the case), standard regression methods do not work as advertised. Instead, the probability of getting statistically significant results can go to one even when the null hypothesis is true and there is nothing real to discover. You should be appalled.

**A large-scale simulation study** All this is true as the sample size goes to infinity, but in reality no sample size can approach infinity. So it is important to see what happens for realistic sample sizes. The idea is to use computer-generated pseudo-random numbers to generate data sets in which the true parameter values are known, because actually those true parameter values are inputs to the program. Applying statistical methods to

such simulated data allows one to investigate the performance of the methods empirically as well mathematically. Ideally, empirical and mathematical investigations of statistical questions are complementary, and usually reinforce one another.

Brunner and Austin [14] took this approach to the topic under discussion. They report a large simulation study in which random data sets were generated according to a factorial design with six factors. The factors were

- Sample size: $n = 50, 100, 250, 500, 1000$
- $Corr(X_1, X_2)$: $\phi_{12} = 0.00, 0.25, 0.75, 0.80, 0.90$
- Proportion of variance in $Y$ explained by $X_1$: 0.25, 0.50, 0.75
- Reliability of $W_1$: 0.50, 0.75, 0.80, 0.90, 0.95
- Reliability of $W_2$: 0.50, 0.75, 0.80, 0.90, 0.95
- Distribution of the latent variables and error terms: Normal, Uniform, $t$, Pareto.

Thus there were $5\times5\times3\times5\times5\times4 = 7{,}500$ treatment combinations. Ten thousand random data sets were generated within each treatment combination, for a total of 75 million data sets. All the data sets were generated according to the true model of Example 0.7.3, with $\beta_2 = 0$, so that $H_0 : \beta_2 = 0$ was true in each case. For each data set, we fit the naive model (no measurement error), and tested $H_0 : \beta_2 = 0$ at $\alpha = 0.05$. The proportion of times $H_0$ is rejected is a Monte Carlo estimate of the Type I Error Probability.

The study yielded 7,500 estimated Type I error probabilities, and even looking at all of them is a big job. Table 1 shows a small but representative part of the results. In this table, all the variables and error terms are normally distributed, and the reliability of both explanatory variables was equal to 0.90. This means that 90% of the variance came from the real thing as opposed to random noise – a stellar value. The values of the regression coefficients were $\beta_0 = 1$, $\beta_1 = 1$ and of course $\beta_2 = 0$.

Remember that we are trying to test the effect of $X_1$ on $Y$ controlling for $X_2$, and since we don't have $X_1$ and $X_2$, we are using $W_1$ and $W_2$ instead. In fact, because $H_0 : \beta_2 = 0$ is true, $X_2$ is conditionally independent of $Y$ given $X_1 = x_1$. This means that the estimated Type I error probabilities in Table 1 should all be around 0.05 if the test is working properly.

When the correlation between $X_1$ and $X_2$ is zero (the first column of Table 1), none of the estimated Type I error probabilites is significantly different from 0.05. This is consistent with Equation (37), where $\widehat{\beta}_2$ converges to the right target when the covariance between $X_1$ and $X_2$ is zero. But as the correlation between explanatory variables increases, so does the Type I error probability – especially when the $X_1$ and $Y$ is strong and the sample size is large. Look at the intermediate case in which 50% of variance in $Y$ is explained by $X_1$ (admittedly a strong relationship, at least in the social sciences) and $n = 250$. As the correlation between $X_1$ and $X_2$ increases from zero to 0.90, the Type I error probability increases from 0.05 to about 0.60. With the strongest relationship beween $X_1$ and $Y$, and the largest sample size, the test of $X_2$'s relationship to $Y$ controlling for

Table 1: Estimated Type I Error

| | Correlation between $X_1$ and $X_2$ | | | | |
|---|---|---|---|---|---|
| $n$ | 0.00 | 0.25 | 0.75 | 0.80 | 0.90 |
| 25% of variance in $Y$ is explained by $X_1$ | | | | | |
| 50 | 0.0491$^\dagger$ | 0.0505$^\dagger$ | 0.0663 | 0.0740 | 0.0838 |
| 100 | 0.0541$^\dagger$ | 0.0527$^\dagger$ | 0.0896 | 0.0925 | 0.1227 |
| 250 | 0.0479$^\dagger$ | 0.0577$^\dagger$ | 0.1364 | 0.1688 | 0.2585 |
| 500 | 0.0510$^\dagger$ | 0.0588$^\dagger$ | 0.2399 | 0.2887 | 0.4587 |
| 1000 | 0.0489$^\dagger$ | 0.0734 | 0.4175 | 0.4960 | 0.7391 |
| 50% of variance in $Y$ is explained by $X_1$ | | | | | |
| 50 | 0.0518$^\dagger$ | 0.0535$^\dagger$ | 0.0949 | 0.1081 | 0.1571 |
| 100 | 0.0501$^\dagger$ | 0.0541$^\dagger$ | 0.1512 | 0.1763 | 0.2710 |
| 250 | 0.0487$^\dagger$ | 0.0710 | 0.3065 | 0.3765 | 0.5994 |
| 500 | 0.0518$^\dagger$ | 0.0782 | 0.5499 | 0.6487 | 0.8740 |
| 1000 | 0.0500$^\dagger$ | 0.1132 | 0.8260 | 0.9120 | 0.9932 |
| 75% of variance in $Y$ is explained by $X_1$ | | | | | |
| 50 | 0.0504$^\dagger$ | 0.0554$^\dagger$ | 0.1669 | 0.2072 | 0.3361 |
| 100 | 0.0510$^\dagger$ | 0.0599 | 0.3019 | 0.3791 | 0.5943 |
| 250 | 0.0487$^\dagger$ | 0.0890 | 0.6399 | 0.7542 | 0.9441 |
| 500 | 0.0496$^\dagger$ | 0.1296 | 0.9058 | 0.9599 | 0.9987 |
| 1000 | 0.0502$^\dagger$ | 0.2157 | 0.9969 | 0.9992 | 1.0000 |

$^\dagger$Not Significantly different from 0.05, Bonferroni corrected for 7,500 tests.

$X_1$ was significant 10,000 times out of 10,000. Again, this is when the null hypothesis is true, and $Y$ is conditionally independent of $X_2$, given $X_1$.

Again, this simulation study was a 6-factor experiment with 7,500 treatment combinations. A rough way to see general trends is to look at marginal means, averaging the estimated Type I error probabilities over the other factors, for each factor in the study. Table 2 is actually six subtables, showing marginal estimated Type I error probabilities for each factor. The only one that may not be self-explanatory is "Base distribution." This is the distribution of $X_1, X_2, e_1$ and $e_2$, shifted when necessary to have expected value zero, and scaled to have variance for the particular treatment condition.

The inescapable conclusion is that ignoring measurement error in the explanatory variables can seriously inflate Type I error probabilities in multiple regression. To repeat, ignoring measurement error is what people do all the time. The poison combination is measurement error in the variable for which you are "controlling," and correlation between latent explanatory variables. If either is zero, there is no problem. Factors affecting severity of the problem are

- As the correlation between $X_1$ and $X_2$ increases, the problem gets worse.
- As the correlation between $X_1$ and $Y$ increases, the problem gets worse.

Table 2: Marginal Type I Error Probabilities

| Base Distribution | | | |
|---|---|---|---|
| normal | Pareto | t Distr | uniform |
| 0.38692448 | 0.36903077 | 0.38312245 | 0.38752571 |

| Explained Variance | | |
|---|---|---|
| 0.25 | 0.50 | 0.75 |
| 0.27330660 | 0.38473364 | 0.48691232 |

| Correlation between Latent Independent Variables | | | | |
|---|---|---|---|---|
| 0.00 | 0.25 | 0.75 | 0.80 | 0.90 |
| 0.05004853 | 0.16604247 | 0.51544093 | 0.55050700 | 0.62621533 |

| Sample Size n | | | | |
|---|---|---|---|---|
| 50 | 100 | 250 | 500 | 1000 |
| 0.19081740 | 0.27437227 | 0.39457933 | 0.48335707 | 0.56512820 |

| Reliability of $W_1$ | | | | |
|---|---|---|---|---|
| 0.50 | 0.75 | 0.80 | 0.90 | 0.95 |
| 0.60637233 | 0.46983147 | 0.42065313 | 0.26685820 | 0.14453913 |

| Reliability of $W_2$ | | | | |
|---|---|---|---|---|
| 0.50 | 0.75 | 0.80 | 0.90 | 0.95 |
| 0.30807933 | 0.37506733 | 0.38752793 | 0.41254800 | 0.42503167 |

- As the amount of measurement error in $X_1$ increases, the problem gets worse.
- As the amount of measurement error in $X_2$ increases, the problem gets *less* severe.
- As the sample size increases, the problem gets worse.
- Distribution of the variables does not matter much.

It is particularly noteworthy that the inflation of Type I error probability gets worse with increasing sample size. Generally in statistics, things get better as the sample size increases. This is an exception. For a large enough sample size, no amount of measurement error in the explanatory variables is safe, assuming that the latent explanatory variables are correlated.

It might be objected that null hypotheses are never exactly true in observational studies, so that estimating Type I error probability is a meaningless exercise. However, look at expression (36), the large-sample target of $\widehat{\beta}_2$ when the true value of $\beta_2$ (the parameter being tested) is not necessarily zero. Suppose that the true value of $\beta_2$ is negative, the true value of $\beta_1$ is positive, and the covariance between $X_1$ and $X_2$ is positive. This is a perfectly natural scenario. Depending on the values of the variances

and covariances, it is quite possible for the second term in (36) to be a larger positive value, overwhelming $\beta_2$ and making the large-sample target of $\widehat{\beta}_2$ positive. Brunner and Austin report a smaller-scale simulation of this situation in which measurement error leads to rejection of the null hypothesis in the wrong direction nearly 100% of the time. This is a particularly nasty possibility, because findings that are opposite of the truth (especially if they are published) can only serve to muddy the waters and make scientific progress slower and more difficult.

Brunner and Austin go on to show that the inflation of Type I error probability arising from measurement error is not limited to multiple regression and measurement error of a simple additive type. It applies to other kinds of regression and other types of measurement error, including logistic regression, proportional hazards regression in survival analysis, log-linear models (for testing conditional independence in the presence of classification error, and median splits on explanatory variables, which is a kind of measurement error created by the data analyst. Even converting $X_1$ to ranks inflates Type I Error probability.

This is a serious problem, but only if one is interested in interpreting the results of statistical analyses to find out more about the world. If the only interest is in prediction, you just use the variables you have. You might wish your predictors were measured with less error, because that might make the predictions more accurate. But it doesn't really matter whether a given regression coefficient is positive or negative. On the other hand, if this is science, then it matters.

It's worth observing that the news about true experimental studies is good. The first column of Table 1, where the covariance of explanatory variables is zero, illustrates the primary virtue of random assignment: it erases any relationship between experimental treatment and potential confounding variables. Thinking of $X_2$ as the treatment and $X_1$ as a covariate, it is apparent that in an experimental study, the Type I error probability is not inflated by measurement error in the treatment, the covariate, or both – as long as random assignment has made the latent versions of these variables independent, and the experimental procedure has been of sufficiently high quality that the corresponding measurement errors are uncorrelated.

This example also illustrates that assignment to experimental conditions need not be random to be effective. All that's needed is to somehow break up the relationship between the treatment and any possible confounding variables. In a clinical trial, for example, suppose that patients coming in to a medical clinic are assigned to experimental and control conditions alternately, and not randomly. There is no serious problem with this, because treatment condition would still be unrelated to any characteristic of the patients.

The whole issue of measurement error in the predictors is really just a sentence or two in the narrative about correlation versus causation. It goes like this. If $X$ is related to $Y$, it could be that $X$ is influencing $Y$, or that $Y$ is influencing $X$, or that some confounding variables related to $X$ are influencing $Y$. You might think that if you have an idea what those confounding variables are, you can control for them with regression methods. Unfortunately, if potential confounding variables are measured with error, the

standard ways of controlling for them do not quite work (Brunner and Austin, 2009)[21].

The last two sentences are the addition to the standard narrative. It's only a couple of sentences, but it's still a big deal, because correlation-causation is a fundamental issue in research design. What's the solution? Surely it must be to admit that measurement error exists, and incorporate it directly into the statistical model.

## 0.8 Modeling measurement error

Ignoring measurement error in regression can yield conclusions that are very misleading. But as soon as we try building measurement error into the statistical model, we encounter a technical issue that will occupy a central role in this book: parameter identifiability.

### A first try at including measurement error

**Example 0.8.1** *Model Includes Measurement Error*

The following is basically the true model of Example 0.7.2, with everything normally distributed. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_i + \epsilon_i \\ W_i &= \nu + X_i + e_i, \end{aligned} \tag{38}$$

where

- $X_i$ is normally distributed with mean $\mu_x$ and variance $\phi > 0$
- $\epsilon_i$ is normally distributed with mean zero and variance $\psi > 0$
- $e_i$ is normally distributed with mean zero and variance $\omega > 0$
- $X_i, e_i, \epsilon_i$ are all independent.

The intercept term $\nu$ could be called "measurement bias." If $X_i$ is true amount of exercise per week and $W_i$ is reported amount of exercise per week, $\nu$ is the average amount by which people exaggerate.

Data from Model (38) are just the pairs $(W_i, Y_i)$ for $i = 1, \ldots, n$. The true explanatory variable $X_i$ is a latent variable whose value cannot be known exactly. The model implies that the $(W_i, Y_i)$ are independent bivariate normal with

$$E\begin{pmatrix} W_i \\ Y_i \end{pmatrix} = \boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} \mu_x + \nu \\ \beta_0 + \beta_1 \mu_x \end{pmatrix},$$

[21] I could not resist citing the paper. There is no claim that Brunner and Austin discovered the problem with measurement error in the predictors. The ill effects of measurement error on estimation have been known since the 1930s, though the issue has been mostly ignored by mainstream statisticians and other users of statistical methods. What Brunner and Austin did was to review the literature and document the effect of measurement error on significance testing.

and variance covariance matrix

$$cov\begin{pmatrix} W_i \\ Y_i \end{pmatrix} = \boldsymbol{\Sigma} = [\sigma_{i,j}] = \begin{pmatrix} \phi+\omega & \beta_1\phi \\ \beta_1\phi & \beta_1^2\phi+\psi \end{pmatrix}.$$

There is a big problem here, and the moment structure equations reveal it.

$$\begin{aligned} \mu_1 &= \mu_x + \nu \\ \mu_2 &= \beta_0 + \beta_1\mu_x \\ \sigma_{1,1} &= \phi + \omega \\ \sigma_{1,2} &= \beta_1\phi \\ \sigma_{2,2} &= \beta_1^2\phi + \psi. \end{aligned} \tag{39}$$

It is impossible to solve these five equations for the seven model parameters[22]. That is, even with perfect knowledge of the probability distribution of the data (for the multivariate normal, that means knowing $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, period), it would be impossible to know the model parameters.

To make the problem clearer, look at the table below. It shows two diferent set of parameter values $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ that both yield the same mean vector and covariance matrix, and hence the exact same distribution of the observable data.

| | $\mu_x$ | $\beta_0$ | $\nu$ | $\beta_1$ | $\phi$ | $\omega$ | $\psi$ |
|---|---|---|---|---|---|---|---|
| $\boldsymbol{\theta}_1$ | 0 | 0 | 0 | 1 | 2 | 2 | 3 |
| $\boldsymbol{\theta}_2$ | 0 | 0 | 0 | 2 | 1 | 3 | 1 |

Both $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ imply a bivariate normal distribution with mean zero and covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 4 & 2 \\ 2 & 5 \end{pmatrix},$$

and thus the same distribution of the sample data.

No matter how large the sample size, it will be impossible to decide between $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, because they imply exactly the same probability distribution of the observable data. The problem here is that the parameters of Model (38) are not *identifiable*. This calls for a brief discussion of identifiability, a topic of central importance in structural equation modeling.

## 0.9 Parameter Identifiability

**The Basic Idea** Suppose we have a vector of observable data $\mathbf{D} = (D_1, \ldots, D_n)$, and a statistical model (a set of assertions implying a probability distribution) for $\mathbf{D}$. The model depends on a parameter $\theta$, which is usually a vector. If the probability distribution of $\mathbf{D}$ corresponds uniquely to $\theta$, then we say that the parameter vector is *identifiable*.

[22] That's a strong statement, and a strong theorem is coming to justify it.

But if any two different parameter values yield the same probability distribution, then the parameter vector is not identifiable. In this case, the data cannot be used to decide between the two parameter values, and standard methods of parameter estimation will fail. Even an infinite amount of data cannot tell you the true parameter values.

**Definition 0.4** *A* Statistical Model *is a set of assertions that partly*[23] *specify the probability distribution of a set of observable data.*

**Definition 0.5** *Suppose a statistical model implies* $\mathbf{D} \sim P_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \Theta$. *If no two points in* $\Theta$ *yield the same probability distribution, then the parameter* $\boldsymbol{\theta}$ *is said to be* identifiable. *On the other hand, if there exist* $\boldsymbol{\theta}_1$ *and* $\boldsymbol{\theta}_2$ *in* $\Theta$ *with* $P_{\boldsymbol{\theta}_1} = P_{\boldsymbol{\theta}_2}$, *the parameter* $\boldsymbol{\theta}$ *is* not identifiable.

A good example of non-identifiability appears in Example 0.4.1 in Section 0.4 on omitted variables in regression. There, the correct model has a set of infinitely many parameter values that imply exactly the same probability distribution of the observed data.

**Theorem 0.1** *If the parameter vector is not identifiable, consistent estimation for all points in the parameter space is impossible.*

In Figure 14, $\theta_1$ and $\theta_2$ are two distinct sets of parameter values for which the distribution of the observable data is the same.

Figure 14: Two parameters values yielding the same probability distribution


Let $T_n$ be a estimator that is consistent for both $\theta_1$ and $\theta_2$. What this means is that if $\theta_1$ is the correct parameter value, eventually as $n$ increases, the probability distribution of $T_n$ will be concentrated in the circular neighborhood around $\theta_1$. And if $\theta_1$ is the correct parameter value, it the probability distribution will be concentrated around $\theta_2$.

But the probability distribution of the data, and hence of $T_n$ (a function of the data) is identical for $\theta_1$ and $\theta_2$. This means that for a large enough sample size, most of $T_n$'s probability distribution must be concentrated in the neighborhood around $\theta_1$, and at the same time it must be concentrated in the neighborhood around $\theta_2$. This is impossible, since the two regions do not overlap. Hence there can be no such consistent estimator $T_n$.

[23] Suppose that the distribution is assumed known except for the value of a parameter vector $\boldsymbol{\theta}$. So the distribution is "partly" specified.

Theorem 0.1 says why parameter identifiability is so important. Without it, even an infinite amount of data cannot reveal the values of the parameters.

Surprisingly often, whether a set of parameter values can be recovered from the distribution depends on where in the parameter space those values are located. That is, the parameter vector may be identifiable at some points but not others.

**Definition 0.6** *The parameter is said to be* identifiable *at a point* $\boldsymbol{\theta}_0$ *if no other point in* $\Theta$ *yields the same probability distribution as* $\boldsymbol{\theta}_0$*.*

If the parameter is identifiable at at every point in $\Theta$, it is identifiable.

**Definition 0.7** *The parameter is said to be* locally identifiable *at a point* $\boldsymbol{\theta}_0$ *if there is a neighbourhood of points surrounding* $\boldsymbol{\theta}_0$*, none of which yields the same probability distribution as* $\boldsymbol{\theta}_0$*.*

Obviously, local identifiability at a point is a necessary condition for global identifiability there.

It is possible for individual parameters (or other functions of the parameter vector) to be identifiable even when the entire parameter vector is not.

**Definition 0.8** *Let* $g(\boldsymbol{\theta})$ *be a function of the parameter vector. If* $g(\boldsymbol{\theta}_0) \neq g(\boldsymbol{\theta})$ *implies* $P_{\boldsymbol{\theta}_0} \neq P_{\boldsymbol{\theta}}$ *for all* $\boldsymbol{\theta} \in \Theta$*, then the function* $g(\boldsymbol{\theta})$ *is said to be identifiable at the point* $\boldsymbol{\theta}_0$*.*

For example, let $D_1, \ldots, D_n$ be i.i.d. Poisson random variables with mean $\lambda_1 + \lambda_2$, where $\lambda_1 > 0$ and $\lambda_1 > 0$. The parameter is the pair $\boldsymbol{\theta} = (\lambda_1, \lambda_2)$. The parameter is not identifiable because any pair of $\lambda$ values satisfying $\lambda_1 + \lambda_2 = c$ will produce exactly the same probability distribution. Notice also how maximum likelihood estimation will fail in this case; the likelihood function will have a ridge, a non-unique maximum along the line $\lambda_1 + \lambda_2 = \overline{D}$, where $\overline{D}$ is the sample mean. The function $g(\boldsymbol{\theta}) = \lambda_1 + \lambda_2$, of course, is identifiable.

The failure of maximum likelihood for the Poisson example is very typical of situations where the parameter is not identifiable. Collections of points in the parameter space yield the same probability distribution of the observable data, and hence identical values of the likelihood. Often these form connected sets of infinitely many points, and when a numerical likelihood search reaches such a higher-dimensional ridge or plateau, the software checks to see if it's a maximum, and (if it's good software) complains loudly because the maximum is not unique. The complaints might take unexpected forms, like a statement that the Hessian has negative eigenvalues. But in any case, maximum likelihood estimation fails.

The idea of a *function* of the parameter vector covers a lot of territory. It includes individual parameters and sets of parameters, as well as things like products and ratios of parameters. Look at the moment structure equations (39) of Example 0.8.1 on page 57. If $\sigma_{1,2} = 0$, this means $\beta_1 = 0$, because $\phi$ is a variance, and is greater than zero. Also in this case $\psi = \sigma_{2,2}$ and $\beta_0 = \mu_2$. So, the function $g(\boldsymbol{\theta}) = (\beta_0, \beta_1, \psi)$ is identifiable at

all points in the parameter space where $\beta_1 = 0$. The other four parameters are still not identifiable.

Recall how for the regression model of Example 0.8.1, the moment structure equations (39) consist of five equations in seven unknown parameters. It was shown by a numerical example that there were two different sets of parameter values that produced the same mean vector and covariance matrix, and hence the same distribution of the observable data. Actually, infinitely many parameter values produce the same distribution, and it happens because there are more unknowns than equations. Theorem 0.2 is a strictly mathematical theorem[24] that provides the necessary details.

**Theorem 0.2** *Let*

$$\begin{aligned} y_1 &= f_1(x_1, \ldots, x_p) \\ y_2 &= f_2(x_1, \ldots, x_p) \\ \vdots & \qquad \vdots \\ y_q &= f_q(x_1, \ldots, x_p), \end{aligned}$$

*If the functions $f_1, \ldots, f_q$ are analytic (posessing a Taylor expansion) and $p > q$, the set of points $(x_1, \ldots, x_p)$ where the system of equations has a unique solution occupies at most a set of volume zero in $\mathbb{R}^p$.*

The following corollary to Theorem 0.2 is the fundamental necessary condition for parameter identifiability. It will be called the **Parameter Count Rule**.

**Rule 1:** The Parameter Count Rule. *Suppose identifiability is to be decided based on a set of moment structure equations. If there are more parameters than equations, the parameter vector is identifiable on at most a set of volume zero in the parameter space.*

When the data are multivariate normal (and this will frequently be assumed), then the distribution of the sample data corresponds exactly to the mean vector and covariance matrix, and to say that a parameter value is identifiable means that is can be recovered from elements of the mean vector and covariance matrix. Most of the time, that involves trying to solve the moment structure equations or covariance structure equations for the model parameters.

Even when the data are not assumed multivariate normal, the same process makes sense. Classical structural equation models, including models for regression with measurement error, are based on systems of simultaneous linear equations. Assuming simple random sampling from a large population, the observable data are independent and identically distributed, with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ that may be written as functions of the model parameters in a straightforward way. If it is possible to solve uniquely for a given model parameter in terms of the elements of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, then that

[24] The core of the proof may be found in Appendix 5 of Fisher's (1966) *The identification problem in econometrics.* [27]

parameter is a function of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, which in turn are functions of the probability distribution of the data. A function of a function is a function, and so the parameter is a function of the probability distribution of the data. Hence, it is identifiable.

Another way to reach this conclusion is to observe that if it is possible to solve for the parameters in terms of moments, simply "putting hats on everything" yields Method of Moments estimator. These estimators, though they may be less than ideal in some ways, will still usually be consistent by the Law of Large Numbers and continuous mapping. Theorem 0.1 tells us consistency would be impossible if the parameters were not identifiable.

To summarize, we have arrived at the standard way to check parameter identifiability for any linear simultaneous equation model, not just measurement error regression. *First, calculate the expected value and covariance matrix of the observable data, as a function of the model parameters. If it is possible to solve uniquely for the model parameters in terms of the means, variances and covariances of the observable data, then the model parameters are identifiable.*

If two distinct parameter vectors yield the same pair $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and the distribution is multivariate normal, the parameter vector is clearly not identifiable. When the distribution is *not* multivariate normal this conclusion does not necessarily follow; the parameters might be recoverable from higher moments, or possibly from the moment-generating function or characteristic function.

But this would require knowing exactly what the non-normal distribution of the data might be. When it comes to analyzing actual data using linear models like the ones in this book, there are really only two alternatives. Either the distribution is assumed[25] normal, or it is acknowledged to be completely unknown. In both cases, parameters will either be identifiable from the mean and covariance matrix (usually just the covariance matrix), or they will not be identifiable at all.

The conclusion is that in practice, "identifiable" means identifiable from the moments. This explains why the parameter count rule (Rule 1) is frequently used to label parameters "not identifiable" even when there is no assumption of normality.

## 0.10 Double measurement

Consider again the model of Example 0.8.1, a simple regression with measurement error in the single explanatory variable. This represents something that occurs all too frequently in practice. The statistician or scientist has a data set that seems relevant to a particular topic, and a model for the observable data that is more or less reasonable. But the parameters of the model cannot be identified from the distribution of the data. In such cases, valid inference is very challenging, if indeed it is possible at all.

The best way out of this trap is to avoid getting trapped in the first place. Plan the statistical analysis in advance, and ensure identifiability by collecting the right kind of data. Double measurement is a straightforward way to get the job done. The key is to

[25] Even when the the data are clearly not normal, methods – especially likelihood ratio tests – based on a normal model can work quite well.

measure the explanatory variables twice, preferably using different methods or measuring instruments[26].

### 0.10.1 A scalar example

**Example 0.10.1**

Instead of measuring the explanatory variable only once, suppose we had a second, independent measurement; "independent" means that the measurement errors are statistically independent of one another. Perhaps the two measurements are taken at different times, using different instruments or methods. Then we have the following model. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} W_{i,1} &= \nu_1 + X_i + e_{i,1} \\ W_{i,2} &= \nu_2 + X_i + e_{i,2} \\ Y_i &= \beta_0 + \beta_1 X_i + \epsilon_i, \end{aligned} \tag{40}$$

where

- $X_i$ is normally distributed with mean $\mu_x$ and variance $\phi > 0$
- $\epsilon_i$ is normally distributed with mean zero and variance $\psi > 0$
- $e_{i,1}$ is normally distributed with mean zero and variance $\omega_1 > 0$
- $e_{i,2}$ is normally distributed with mean zero and variance $\omega_2 > 0$
- $X_i, e_{i,1}, e_{i,2}$ and $\epsilon_i$ are all independent.

The model implies that the triples $\mathbf{D}_i = (W_{i,1}, W_{i,2}, Y_i)^\top$ are multivarate normal with

$$E(\mathbf{D}_i) = E\begin{pmatrix} W_{i,1} \\ W_{i,1} \\ Y_i \end{pmatrix} = \begin{pmatrix} \mu_x + \nu_1 \\ \mu_x + \nu_2 \\ \beta_0 + \beta_1 \mu_x \end{pmatrix},$$

[26]The reason for different instruments or methods is to ensure (or try to ensure) that the errors of measurements are independent. For example, suppose a questionnaire is designed to measure racism. Respondents differ in their actual, true unobservable level of racism. They also differ in the extent to which they wish to be perceived as non-racist. If you give people two similar questionnaires in which they agree or disagree with various statements that are obviously about racism, the individuals who fake good on one questionnaire will also fake good on the other one. The result is that if $e_1$ and $e_2$ are the measurement errors in the two questionnaires, then $e_1$ and $e_2$ will surely have positive covariance. If the unknown covariance is assumed zero, the result will almost surely be incorrect estimation and inference. If the unknown covariance is a parameter in the model, it usually will create problems with identifiability. This all may seem quite technical, but there is a common-sense version. Problems with identifiability almost always correspond to shortcomings in research design. If data are collected in a way that is poorly thought out, the data analysis is unlikely to yield valid conclusions. Taking two measurements that are likely to be contaminated in the same way is just not very smart.

and variance covariance matrix

$$cov(\mathbf{D}_i) = \boldsymbol{\Sigma} = [\sigma_{i,j}] = \begin{pmatrix} \phi + \omega_1 & \phi & \beta_1\phi \\ & \phi + \omega_2 & \beta_1\phi \\ & & \beta_1^2\phi + \psi \end{pmatrix}. \tag{41}$$

Here are some comments.

- There are now nine moment structure equations in nine unknown parameters. This model passes the test of the parameter count rule, meaning that identifiability is possible, but not guaranteed.
- Notice that the model dictates $\sigma_{1,3} = \sigma_{2,3}$. This *model-induced constraint* upon $\boldsymbol{\Sigma}$ is testable. If $H_0 : \sigma_{1,3} = \sigma_{2,3}$ were rejected, the correctness of the model would be called into question[27]. Thus, the study of parameter identifiability leads to a useful test of model fit.
- The constraint $\sigma_{1,3} = \sigma_{2,3}$ allows two solutions for $\beta_1$ in terms of the moments: $\beta_1 = \sigma_{13}/\sigma_{12}$ and $\beta_1 = \sigma_{23}/\sigma_{12}$. Does this mean the solution for $\beta_1$ is not "unique?" No; everything is okay. Because $\sigma_{1,3} = \sigma_{2,3}$, the two solutions are actually the same. If a parameter can be recovered from the moments in any way at all, it is identifiable.
- For the other model parameters appearing in the covariance matrix, the additional measurement of the explanatory variable also appears to have done the trick. It is easy to solve for $\phi, \omega_1, \omega_2$ and $\psi$ in terms of $\sigma_{i,j}$ values. Thus, these parameters are identifiable.
- On the other hand, the additional measurement did not help with the means and intercepts *at all*. Even assuming $\beta_1$ known because it can be recovered from $\boldsymbol{\Sigma}$, the remaining three linear equations in four unknowns have infinitely many solutions. There are still infinitely many solutions if $\nu_1 = \nu_2$.

Maximum likelihood for the parameters in the covariance matrix would work up to a point, but the lack of unique values for $\mu_x, \nu_1, \nu_2$ and $\beta_0$ would cause numerical problems. A good solution is to *re-parameterize* the model, absorbing $\mu_x + \nu_1$ into a parameter called $\mu_1$, $\mu_x + \nu_2$ into a parameter called $\mu_2$, and $\beta_0 + \beta_1\mu_x$ into a parameter called $\mu_3$. The parameters in $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)^\top$ lack meaning and interest[28], but we can estimate them with the vector of sample means $\overline{\mathbf{D}}$ and focus on the parameters in the covariance matrix.

[27] Philosophers of science agree that *falsifiability* – the possibility that a scientific model can be challenged by empirical data – is a very desirable property. The Wikipedia has a good discussion under *Falsifiability* — see http://en.wikipedia.org/wiki/Falsifiable. Statistical models may be viewed as primitive scientific models, and should be subject to the same scrutiny. It would be nice if scientists who use statistical methods would take a cold, clear look at the statistical models they are using, and ask "Is this a reasonable model for my data?"

[28] If $X_i$ is true amount of exercise, $\mu_x$ is the average amount of exercise in the population; it's very meaningful. Also, the quantity $\nu_1$ is interesting; it's the average amount people exaggerate how much they exercise using Questionnaire One. But when you add these two interesting quantities together, you get garbage. The parameter $\boldsymbol{\mu}$ in the re-parametrerized model is a garbage can.

Here is the multivariate normal likelihood from Appendix A.4, simplified so that it's clear that the likelihood depends on the data only through the MLEs $\overline{\mathbf{D}}$ and $\widehat{\boldsymbol{\Sigma}}$. This is just a reproduction of expression (A.20) from Appendix A.

$$L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-np/2} \exp -\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) + (\overline{\mathbf{D}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{D}} - \boldsymbol{\mu})\right\} \tag{42}$$

Notice that if $\boldsymbol{\Sigma}$ is positive definite then so is $\boldsymbol{\Sigma}^{-1}$, and so for *any* positive definite $\boldsymbol{\Sigma}$ the likelihood is maximized when $\boldsymbol{\mu} = \overline{\mathbf{D}}$. In that case, the last term just disappears. So, re-parameterizing and then letting $\widehat{\boldsymbol{\mu}} = \overline{\mathbf{D}}$ leaves us free to conduct inference on the model parameters in $\boldsymbol{\Sigma}$.

Just to clarify, after re-parameterization and estimation of $\boldsymbol{\mu}$ with $\overline{\mathbf{D}}_n$, the likelihood function may be written

$$L(\boldsymbol{\theta}) = |\boldsymbol{\Sigma}(\boldsymbol{\theta})|^{-n/2}(2\pi)^{-np/2} \exp -\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1})\right\}, \tag{43}$$

where $\boldsymbol{\theta}$ is now a vector of just those parameters appearing in the covariance matrix. This formulation is general. For the specific case of the scalar double measurement Example 0.10.1, $\boldsymbol{\theta} = (\phi, \omega_1, \omega_2, \beta_1, \psi)^\top$, and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is given by Expression (41). Maximum likelihood estimation is numerical, and the full range of large-sample likelihood methods described in Section A.6.3 of Appendix A is available.

### Testing goodness of model fit

When there are more covariance structure equations than unknown parameters and the parameters are identifiable, the parameters are said to be *over-identified*. In this case, the model implies functional connections between some variances and covariances. In the small example we are considering, it is clear from Expression (41) on page 64 that $\sigma_{13} = \sigma_{23}$, because they both equal $\beta_1\phi$. This is a testable null hypothesis, and if it is rejected, the model is called into question.

The traditional way to do the test[29] is to compare the fit of the model to the fit of a completely unrestricted multivariate normal using the test statistic

$$G^2 = -2\ln\frac{L\left(\overline{\mathbf{D}}, \boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\right)}{L(\overline{\mathbf{D}}, \widehat{\boldsymbol{\Sigma}})} = n\left(tr\left(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\right) - \ln\left|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\right| - p\right), \tag{44}$$

where $\widehat{\boldsymbol{\Sigma}}$ is the ordinary sample variance-covariance matrix with $n$ in the denominator, and $L(\cdot, \cdot)$ is the multivariate normal likelihood (42) on page 65. The degrees of freedom equals the number of covariance structure equations minus the number of parameters. The idea is that if there are $r$ parameters and $m$ unique variances and covariances, the

[29] The test is documented on page 447 of Jöreskog's classic (1978) article [37] in *Psychometrika*, but I believe it had been in Jöreskog and Sörbom's LISREL software for years before that.

model imposes $m - r$ equality constraints on the variances and covariances[30]. Those are the constraints being tested, even when we don't know exactly what they are. The goodness of fit test is examined more closely in Chapter 7.

The matrix $\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$ is called the *reproduced covariance matrix*. It is the covariance matrix of the observable data, written as a function of the model parameters and evaluated at the MLE. For the present example,

$$\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}}) = \begin{pmatrix} \widehat{\phi} + \widehat{\omega}_1 & \widehat{\phi} & \widehat{\beta}_1\widehat{\phi} \\ & \widehat{\phi} + \widehat{\omega}_2 & \widehat{\beta}_1\widehat{\phi} \\ & & \widehat{\beta}_1^2\widehat{\phi} + \widehat{\psi} \end{pmatrix}$$

The reproduced covariance matrix obeys all model-induced constraints, while $\widehat{\boldsymbol{\Sigma}}$ does not. However, they should be close if the model is right. In the limiting case where $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$, the $G^2$ statistic in (44) equals zero.

When the parameter vector is identifiable and there are more unique variances and covariances than parameters, we call the parameter vector *over-identifiable*. An alternative terminology is to say that the "model is over-identified." The equality restrictions on $\boldsymbol{\Sigma}$ imposed by the model are called *over-identifying restrictions*. The likelihood ratio test for goodness of fit is testing the null hypothesis that the over-identifying restrictions are true.

Suppose that the entire parameter vector is identifiable, and $m = k$. That is, the number of parameters is equal to the number of unique variances and covariances. In this case, identifiability is established by solving $k$ equations in $k$ unknowns. The function from parameters to the variances and covariances is one-to-one (injective), and the model imposes no constraints on the variances and covariances. In this case the parameter vector is said to be *just identifiable*. Alternatively, the model is often said to be "just identified," or *saturated*. In this case, $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$ by the invariance principle, and the likelihood ratio test statistics for goodness of fit automatically equals zero. The degrees of freedom $m - k = 0$ also. These values are usually displayed by software, which could be confusing unless you know why. It means the model is not testable. It is incapable of being challenged by any data set, at least using this technology.

### 0.10.2 Computation with `lavaan`

A variety of commercial software is available for fitting structural equation models, including LISREL, EQS, Amos and Mplus. I myself have used mostly SAS `proc calis` until recently. In keeping with the open-source philosophy of this text, we will use the free, open-source R package `lavaan`; the name is short for LAtent VAriable ANalysis. The

[30]Here's why. In most cases, it is possible to choose just $r$ of the $m$ variances and covariances, and establish identifiability by solving $r$ equations in $r$ unknowns. In this case, there are $m - r$ unused, redundant equations. Each sets a variance or covariance equal to some function of the model parameters. Substituting the solutions for the parameters in terms of $\sigma_{ij}$ back into the unused equations will yield $m - r$ equality constraints on the variances and covariances.

software is described very well by Rosseel [48] in his 2012 article in the *Journal of Statistical Software*. The capabilities of `lavaan` have grown since the article was published. A nice tutorial is available at http://lavaan.ugent.be/tutorial.

This first illustration of `lavaan` will use a data set simulated from the model of Example 0.10.1, the same little double measurement example we have been studying. It may be a toy example, but it's an educational toy. Readers familiar with `lavaan` will notice that for now, I am using synax that favours explicitness over brevity. R input and output will be interspersed with explanation.

When I begin an R session, I like to clear the deck with `rm(list=ls())`, removing any existing R objects that may be in the workspace. The statement `options(scipen=999)` suppresses scientific notation. This is just a matter of taste.

The `lavaan` package may be installed with the `install.packages` command. You only need to do this once, which is why it's commented out below. `library(lavaan)` is necessary to load the package, every time.

```
> rm(list=ls()); options(scipen=999)
> # install.packages("lavaan", dependencies = TRUE)
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
```

Next, we read the data, look at the first few lines, and obtain a summary and correlation matrix. Notice that that the data file has only observable variables (obviously), and that their means are certainly not zero. In practice, we would examine the data much more carefully. This vital step in data analysis will not be mentioned again.

```
> babydouble = read.table("http://www.utstat.toronto.edu/~brunner/openSEM
 /data/Babydouble.data.txt")
> head(babydouble)
     W1    W2     Y
1  9.94 12.24 15.23
2 12.42 11.32 14.55
3 10.43 10.40 12.40
4  9.07  9.85 17.09
5 11.04 11.98 16.83
6 10.40 10.85 15.04
> summary(babydouble)
       W1               W2               Y
 Min.   : 6.190   Min.   : 6.76   Min.   : 3.98
 1st Qu.: 8.932   1st Qu.: 9.11   1st Qu.:10.97
 Median : 9.720   Median :10.05   Median :13.22
 Mean   : 9.809   Mean   :10.06   Mean   :13.10
 3rd Qu.:10.655   3rd Qu.:10.99   3rd Qu.:15.46
 Max.   :12.830   Max.   :13.57   Max.   :21.62
```

```
> cor(babydouble)
          W1        W2         Y
W1 1.0000000 0.5748331 0.1714324
W2 0.5748331 1.0000000 0.1791539
Y  0.1714324 0.1791539 1.0000000
```

Notice that the sample correlations of $W_1$ with $Y$ and $W_2$ with $Y$ are very close. This is consistent with the model-induced constraint $\sigma_{13} = \sigma_{23}$, especially if $\omega_1 = \omega_2$.

Next comes specification of the model to be fit. Again, this is the model of Example 0.10.1 on page 63. The entire model specification is in a *model string*, assigned to the string variable `dmodel1`. If the model is big and you are using it repeatedly, you can compose the model string in a separate file and bring it in with `readlines`.

```
> dmodel1 =    'Y ~ beta1*X          # Latent variable model (even though Y is observed)
               X =~ 1*W1 + 1*W2      # Measurement model
               # Variances (covariances would go here too)
               X~~phi*X      # Var(X) = phi
               Y~~psi*Y      # Var(epsilon) = psi
               W1~~omega1*W1 # Var(e1) = omega1
               W2~~omega2*W2 # Var(e2) = omega2
              '
```

It's best to discuss the model string line by line.

`Y ~ beta1*X`: This is reminiscent of R's `lm` syntax. The translation is $Y = \beta_1 X + \epsilon$. Notice that there is no $\beta_0$. Though you can specify intercepts and expected values in `lavaan` if you wish, by default they are invisible. Thus the whole process of re-parameterization and swallowing all the non-identifiable expected values and intercepts into $\boldsymbol{\mu}$ (see page 65) is implicit.

`X =~ 1*W1 + 1*W2`: This looks like $X$ is being produced by $W_1$ and $W_2$, when actually it's the other way around. However, if you read $\sim$ and $=\sim$ as two different flavours of "is modelled as," it makes more sense. The statment stands for two model equations:

$$\begin{aligned} W_1 &= 1 * X + e_1 \\ W_2 &= 1 * X + e_2 \end{aligned}$$

These two statements constitute the *measurement model* for this simple example. The observable variables $W_1$ and $W_2$ are called *indicators* of $X$. An indicator of a latent variable is an observable variable that arises from only that latent variable plus an error term. In `lavaan`, a latent variable must have indicators. Otherwise, it is assumed observable even if it's not in the input data set. The explicit "1*" syntax is necessary if you want the coefficients to equal one. Otherwise, `lavaan` will assume you want coefficients that

are free parameters in the model, but you don't feel like naming them. It will try to be helpful, with results that are unfortunate in this case[31].

`X~~ phi*X`: As the comment statement says, this means $Var(X) = \phi$. The double tilde is a way of naming variances, or setting them equal to numeric constants if that's what you want to do. Notice that the symbol $X$ appears on both sides. If you had two different variable names, the statement would specify a covariance. Since a variance may be viewed as the covariance of a random variable with itself, this is good notation. Also be aware that if a covariance is not specified, it equals zero.

`Y~~ psi*Y`: In contrast to the preceding statement, this one is *not* saying that $Var(Y) = \psi$. It is saying $Var(\epsilon) = \psi$. Here's the rule. If a variable appears on the left side of any model equation, then the $\sim\sim$ notation specifies the variance or covariance of the error term in the equation. If the variable appears only on the right side (possibly in more than one equation), the $\sim\sim$ notation specifies the variance or covariance of the variable itself. In this way, though error terms are never named in `lavaan`, you can name their variances, and you can name their covariances with other variables and error terms.

`W1~~omega1*W1`: $Var(e_1) = \omega_1$

`W2~~omega2*W2`: $Var(e_2) = \omega_2$

A covariance between the measurement errors $e_1$ and $e_2$ would be specified with something like `W1~~omega12*W2`. A covariance of $c$ between $X$ and $\epsilon$ would be specified with `X~~c*Y`.

Next, we fit the model and look at a summary. We use the `lavaan` function[32] (same name as the `lavaan` package).

```
> dfit1 = lavaan(dmodel1, data=babydouble)
> summary(dfit1)
lavaan 0.6-7 ended normally after 23 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                          5

  Number of observations                           150

Model Test User Model:
```

[31] `lavaan`'s "helpful" behaviour really is helpful for many users under many circumstances. It is based on rules for parameter identifiability that will be developed later in this text.

[32] Model fitting can also be accomplished with the `sem` and `cfa` functions. With these "user friendly" alternatives, the model specification in the model string is less elaborate, and the software makes choices about the model for you. These choices are intended to be helpful, and may or may not be what you want.

```
  Test statistic                                0.007
  Degrees of freedom                                1
  P-value (Chi-square)                          0.933

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  X =~
    W1                1.000
    W2                1.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X       (bet1)    0.707    0.290    2.442    0.015

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    X        (phi)    1.104    0.181    6.104    0.000
   .Y        (psi)    9.775    1.153    8.481    0.000
   .W1      (omg1)    0.834    0.158    5.265    0.000
   .W2      (omg2)    0.800    0.156    5.123    0.000
```

We first learn that the numerical parameter estimation converged in 23 iterations, $n = 150$, and estimation was by maximum likelihood – the default. Under "`Model Test User Model`," the `Test statistic` is exactly the $G^2$ statistic given in expression (44) on page 65: the likelihood ratio test for goodness of model fit. The small value of $G^2$ and the correspondingly large $p$-value indicate that the model passes this test, and is not called into question.

The next section in the output is entitled `Latent Variables`, saying that $X$ is manafested by the indicators $W_1$ and $W_2$. The "estimates" are the fixed numerical constants of 1.000, specified in the model string. More generally, this section would include all the latent variables in a model. If coefficients (factor loadings) linking the latent variables to their indicators were not pre-specified, their estimates would appear here, together with tests of difference from zero.

The next section of the summary is `Regressions`. These correspond to all the model equations using the $\sim$ rather than the $\sim=$ notation, whether the variables involved are latent or observed. Here, we have maximum likelihood estimates, standard errors, $Z$-tests for whether the parameter equals zero, and two-sided $p$-values. The standard errors are

what you would expect. They are square roots of the diagonal elements of the inverse of the Hessian of the minus log likelihood. If this does not make sense, see the maximum likelihood review in Appendix A. Also, observe that in the `summary` display, the parameter names are abbreviated to four characters.

The last section of the summary is `Variances`. Covariances would go here too, if any had been specified in the model. We have maximum likelihood estimates of the variance parameters, standard errors, and two-sided $Z$-tests for whether the parameter equals zero. When the variance in question is the variance of an error term rather than of the variable itself, the variable name is preceded by a dot, as in `.Y`, `.W1` and `.W2`.

**Testing whether variances equal zero** It might seem strange to test whether variances equal zero, when they are automatically greater than zero according to the model. It's not as silly as you might think. Look at Equation (41) on page 64, which gives the covariance matrix of the observable variables for this model, in terms of the model parameters. The covariance $\sigma_{1,2}$ equals $\phi$, which is a variance. That means that the covariance between $W_1$ and $W_2$ must be greater than zero if the model is correct; this would not necessarily be true for an arbitrary covariance matrix.

The other variance parameters, because they are identifiable, can also be written as functions of the variances and covariances $\sigma_{i,j}$. This means that they also correspond to functions of the variances and covariances — functions that must be greater than zero if the model is correct. In this way, we see that the model also imposes *inequality constraints* on the covariance matrix $\mathbf{\Sigma}$. The most obvious of these constraints[33] can be tested by looking at the estimates of the variance parameters in the model. If the variance estimates are less than zero, particularly if they are *significantly* less than zero, the model is thrown into question.

The conclusion is that testing whether variances equal zero is another way to test model fit. A good practice is to check the equality constraint first with the likelihood ratio test for goodness of fit, and then worry about inequality constraints provided that the first test is non-significant. It is quite common for inequality violations to disappear once the equality violations have been fixed.

The R object created by the `lavaan` function contains a large amount of additional information. The `parameterEstimates` function returns a data frame that gives more detail about the parameter estimates, including confidence intervals.

```
> parameterEstimates(dfit1)
  lhs op rhs  label   est    se     z pvalue ci.lower ci.upper
1   Y  ~   X  beta1 0.707 0.290 2.442  0.015    0.140    1.275
2   X =~  W1        1.000 0.000    NA     NA    1.000    1.000
3   X =~  W2        1.000 0.000    NA     NA    1.000    1.000
4   X ~~   X    phi 1.104 0.181 6.104  0.000    0.750    1.459
5   Y ~~   Y    psi 9.775 1.153 8.481  0.000    7.516   12.034
6  W1 ~~  W1 omega1 0.834 0.158 5.265  0.000    0.524    1.145
```

[33] It can be challenging to obtain all the inequality constraints in a useful form. See Chapter 7.

```
7  W2 ~~  W2 omega2 0.800 0.156 5.123  0.000    0.494    1.105
```

The `parTable` function yields details about the model fitting, including the starting values for the numerical search.

```
> parTable(dfit1)
  id lhs op rhs user block group free ustart exo  label plabel start   est    se
1  1   Y  ~   X    1     1     1    1     NA   0  beta1   .p1. 0.000 0.707 0.290
2  2   X =~  W1    1     1     1    0      1   0          .p2. 1.000 1.000 0.000
3  3   X =~  W2    1     1     1    0      1   0          .p3. 1.000 1.000 0.000
4  4   X ~~   X    1     1     1    2     NA   0    phi   .p4. 0.050 1.104 0.181
5  5   Y ~~   Y    1     1     1    3     NA   0    psi   .p5. 5.164 9.775 1.153
6  6  W1 ~~  W1    1     1     1    4     NA   0 omega1   .p6. 0.968 0.834 0.158
7  7  W2 ~~  W2    1     1     1    5     NA   0 omega2   .p7. 0.953 0.800 0.156
```

A vector containing the parameter estimates may be obtained with the `coef` function. This is useful when the parameter estimates are to be used in further calculations.

```
> coef(dfit1) # A vector of MLEs
 beta1    phi    psi omega1 omega2
 0.707  1.104  9.775  0.834  0.800
```

The `fitted` function returns a list of two matrices. The first element is the reproduced covariance matrix $\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$. The second element is what might be called the "reproduced mean vector" $\boldsymbol{\mu}(\widehat{\boldsymbol{\theta}})$. It will be nonzero if means are specified in the model.

```
> fitted(dfit1) # Sigma(thetahat) and mu(thetahat)
$cov
   W1     W2     Y
W1  1.939
W2  1.104  1.904
Y   0.781  0.781 10.327

$mean
W1 W2  Y
 0  0  0
```

As usual with R, the `vcov` function returns the estimated asymptotic covariance matrix, the inverse of the observed Fisher information (Hessian).

```
> vcov(dfit1)
        beta1  phi    psi    omega1 omega2
beta1   0.084
phi    -0.007  0.033
psi    -0.035  0.002  1.328
omega1  0.003 -0.004 -0.002  0.025
omega2  0.003 -0.005 -0.002 -0.007  0.024
```

Even though the upper triangular entries are not shown, that's just a display method. The whole symmetric matrix is available for furter calculation.

The `logLik` function returns the log likelihood evaluated at the MLE.

```
> logLik(dfit1)
'log Lik.' -878.512 (df=5)
```

It would be possible to use `logLik` to compute likelihood ratio tests, but the `anova` function is more convenient. One can fit a restricted model by specifying the constraints in the `lavaan` statement.

```
> # Fit a restricted model (restricted by H0)
> dfit1r = lavaan(dmodel1, data=babydouble, constraints = 'omega1==omega2')
> anova(dfit1r,dfit1)
Chi Square Difference Test

       Df  AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
dfit1   1 1767 1782.1 0.0071
dfit1r  2 1765 1777.1 0.0262   0.019189       1     0.8898
```

To test a null hypothesis with multiple constraints, put the constraints on separate lines. This is the code for testing $H_0 : \omega_1 = \omega_2, \phi = 1$.

```
> # Put multiple constraints on separate lines.
> dfit1r2 = lavaan(dmodel1, data=babydouble, constraints = 'omega1==omega2
+                                                           phi==1')
> anova(dfit1r2,dfit1)
```

Illustrating a Wald test[34] of $H_9 : \omega_1 = \omega_2$, we first define the publicly available `Wtest` function, and then enter the $\mathbf{L}$ matrix and do the calculation.

```
> # For Wald tests: Wtest = function(L,Tn,Vn,h=0) # H0: L theta = h
> source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/Wtest.txt")
> LL = cbind(0,0,0,1,-1); LL
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    1   -1

> Wtest(LL,coef(dfit1),vcov(dfit1))
         W         df    p-value
0.01918586 1.00000000 0.88983498
```

It is only a little surprising that the Wald and likelihood ratio test statistics are so close. The two tests are asymptotically equivalent under the null hypothesis, meaning that the

[34] The Wald test of the linear null hypothesis $\mathbf{L}\boldsymbol{\theta} = \mathbf{h}$ is given in Section A.6.7 of Appendix A, Equation (A.37) on page 600.

difference between the two test statistic values goes to zero in probability when $H_0$ is true. In this case, the null hypothesis is exactly true (these are simulated data), and the sample size of $n = 150$ is fairly large.

The `lavaan` software makes it remarkably convenient to estimate non-linear functions of the parameters, along with standard errors calculated using the multivariate delta method (see the end of Section A.5 in Appendix A). This is accomplished with the `:=` operator, as shown below. In this example, two functions of the parameter vector are specified. The first function is $\omega_1 - \omega_2$. Because this function is linear, the $Z$-test for whether it equals zero is equivalent to the Wald test of $H_0 : \omega_1 = \omega_2$ directly above. The second function is the reliability of $W_1$. Using Equation (29) on page 41, this is $\frac{\phi}{\phi+\omega_1}$.

```
> # Non-linear functions of the parameters with :=
> dmodel1b = 'Y ~ beta1*X          # Latent variable model
+             X =~ 1*W1 + 1*W2     # Measurement model
+             # Variances (covariances would go here too)
+             X~~phi*X       # Var(X) = phi
+             Y~~psi*Y       # Var(epsilon) = psi
+             W1~~omega1*W1 # Var(e1) = omega1
+             W2~~omega2*W2 # Var(e2) = omega2
+             diff := omega1-omega2
+             rel1 := phi/(omega1+phi)
+             '
> dfit1b = lavaan(dmodel1b, data=babydouble)
> parameterEstimates(dfit1b)
   lhs op              rhs  label   est    se     z pvalue ci.lower ci.upper
1    Y  ~                X  beta1 0.707 0.290 2.442  0.015    0.140    1.275
2    X =~               W1        1.000 0.000    NA     NA    1.000    1.000
3    X =~               W2        1.000 0.000    NA     NA    1.000    1.000
4    X ~~                X    phi 1.104 0.181 6.104  0.000    0.750    1.459
5    Y ~~                Y    psi 9.775 1.153 8.481  0.000    7.516   12.034
6   W1 ~~               W1 omega1 0.834 0.158 5.265  0.000    0.524    1.145
7   W2 ~~               W2 omega2 0.800 0.156 5.123  0.000    0.494    1.105
8 diff :=    omega1-omega2   diff 0.035 0.252 0.139  0.890   -0.458    0.528
9 rel1 := phi/(omega1+phi)   rel1 0.570 0.066 8.657  0.000    0.441    0.699
```

Apart from rounding error, the $Z$ statistic of 0.139 for the null hypothesis $\omega_1 - \omega_2 = 0$ matches the Wald test of the same null hypothesis, with $W = Z^2$.

```
> 0.139^2
[1] 0.019321
```

**Trying to fit models with non-identifiable parameters** This sub-section contains more details about how `lavaan` works, and also some valuable material on the connection of identifiability to maximum likelihood estimation. The account of how double measurement can help with identifiability is continued on page 83.

Trying to estimate the parameters of a structural equation model without first checking identifiability is like jumping out of an airplane without checking that your backpack contains a parachute and not just a sleeping bag. You shouldn't do it. Unfortunately, people do it all the time. Sometimes it's because they have little or no idea what parameter identifiability is. Sometimes it's because the model is a little non-standard, and checking identifiability is too much work[35]. Sometimes, it's because of coding errors. Typos in the model string can easily specify a model that's non-identifiable, because a mis-spelled parameter name is assumed to represent a different parameter. Anyway, it's interesting to see how `lavaan` deals with models you *know* are not identified. The main lesson is that sometimes it complains, and sometimes it just returns a meaningless answer with no obvious indication that anything is wrong. This is not a criticism of `lavaan`. It's a reminder that you need to know what you are doing.

**Example 0.10.2**

In this first example, non-identifiability causes `lavaan` to complain loudly. The model is obtained by taking `dmodel1` (that's the model of Example 0.10.1 on page 63) and adding unknown coefficients $\lambda_1$ and $\lambda_2$ linking $X$ to $W_1$ and $W_2$ respectively[36]. The result is that there are now two more parameters, for a total of seven. There are still only six variances and covariances, so the model fails the parameter count rule, and we know the parameters can be identifiable on at most a set of volume zero in the parameter space.

```
> dmodel2  =  'Y ~ beta1*X          # Latent variable model
+             X =~ lambda1*W1 + lambda2*W2    # Measurement model
+             # Variances (covariances would go here too)
+             X~~phi*X      # Var(X) = phi
+             Y~~psi*Y      # Var(epsilon) = psi
+             W1~~omega1*W1 # Var(e1) = omega1
+             W2~~omega2*W2 # Var(e2) = omega2
+             '
```

When we try to fit the model, it's clear that something is wrong.

```
> dfit2 = lavaan(dmodel2, data=babydouble)
Warning message:
In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING: could not compute standard errors!
  lavaan NOTE: this may be a symptom that the model is not identified.
```

In this case, `lavaan` correctly guessed that the parameters were not identifiable. Here's what happened.

[35] In later chapters, we will use Sage to ease the burden of symbolic calculation. See Appendix B.

[36] This is surely a more believable model.

When `lavaan` does maximum likelihood estimation, it is minimizing a function proportional to the minus log likelihood plus a constant[37]. If the parameter vector is massively non-identifiable as in the present case, the typical parameter vector belongs to an infinite, connected set whose members all yield exactly the same covariance matrix and hence the same value of the function being minimized. The graph of the function does not look like a high-dimensional bowl. Instead, it resembles a high-dimensional river valley. The non-unique minimum is on the flat surface of the water at the bottom of the valley. The numerical search starts somewhere up in the hills, and then trickles downhill, usually until it comes to the river. Then it stops. The stopping place (the MLE) depends entirely on where the search began.

The surface is not strictly concave up at the stopping point, so the Hessian matrix (see Expression A.29 in Appendix A) is not positive definite. However, the valley function is convex, so that the Hessian has to be non-negative definite. Consequently all its eigenvalues are greater than or equal to zero. They can't all be positive, or the Hessian would be positive definite. This means there must be at least one zero eigenvalue. Hence, the determinant of the Hessian is zero and its inverse does not exist.

The standard errors of the MLEs are the square roots of the diagonal elements of the estimated asymptotic variance-covariance matrix. This matrix is obtained by inverting the Hessian of the minus log likelihood; see Expression (A.35) in Appendix A. Since the inverse does not exist, the standard errors can't be computed, and `lavaan` issues a warning about it. This whole scenario is so common that `lavaan` also speculates – correctly in this case – that the problem arises from lack of parameter identifiability.

This is not an error; it's just a warning. A model fit object is created.

```
> summary(dfit2)
lavaan 0.6-7 ended normally after 26 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                          7

  Number of observations                           150

Model Test User Model:

  Test statistic                                    NA
  Degrees of freedom                                -1
  P-value (Unknown)                                 NA

Parameter Estimates:
```

[37]The constant is $L(\overline{\mathbf{D}}, \widehat{\mathbf{\Sigma}})$, the multivariate normal likelihood evaluated at the unrestricted MLE of $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$. The function is also divided by $n$, which can help with numerical accuracy. When the search finds a minimum, multiplication by $2n$ yields the test statistic given in Equation (44).

```
  Standard errors                                Standard
  Information                                    Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  X =~
    W1      (lmb1)    0.962 NA
    W2      (lmb2)    0.998 NA

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X       (bet1)    0.693 NA

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    X        (phi)    1.151 NA
   .Y        (psi)    9.776 NA
   .W1      (omg1)    0.871 NA
   .W2      (omg2)    0.761 NA
```

After "normal" convergence (hummm), the `Minimum Function Test Statistic` is `NA`, or missing even though it could be computed. The degrees of freedom are -1, impossible for a chi-squared statistic. The degrees of freedom are calculated as number of unique variances and covariances minus number of parameters. When it's negative, this is a sure sign the model has failed the parameter count rule, and the parameter vector can't be identifiable. The software could check this and inform the user, but as of this writing it does not. Parameter estimates (corresponding to the point where the search stopped) are given, but standard errors are `NA` and there are no significance tests.

### Example 0.10.3

In this next example, we modify the model of Example 0.10.1 again, keeping the unknown factor loadings $\lambda_1$ and $\lambda_2$ that connect the latent explanatory variable $F$ to its indicators $W_1$ and $W_2$, but making the two measurement error variances equal: $\omega_1 = \omega_2 = \omega$. Everything else remains the same. The model has six unknown parameters and six unique variances and covariances, so it passes the test of the parameter count rule. This means identifiability is possible, but not guaranteed.

```
> # dmodel3 passes the parameter count rule, but its parameters are not identifiable.
> dmodel3  = 'Y ~ beta1*X                    # Latent variable model
+             X =~ lambda1*W1 + lambda2*W2    # Measurement model
+             X~~phi*X      # Var(X) = phi
```

```
+             Y~~psi*Y      # Var(epsilon) = psi
+             W1~~omega*W1 # Var(e1) = omega
+             W2~~omega*W2 # Var(e2) = omega
+             '
> dfit3 = lavaan(dmodel3, data=babydouble)
>
```

`lavaan` fits the model and generates a useful warning.

```
Warning message:
In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING:
    The variance-covariance matrix of the estimated parameters (vcov)
    does not appear to be positive definite! The smallest eigenvalue
    (= 1.121048e-18) is close to zero. This may be a symptom that the
    model is not identified.
```

So, even though `lavaan` is able to numerically invert the Fisher information to get an asymptotic covariance matrix of the MLEs, it correctly speculates that there is a problem with identifiability, and the answer should not be trusted. Looking at `summary`,

```
> summary(dfit3)
lavaan 0.6-7 ended normally after 19 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                          7
  Number of equality constraints                     1

  Number of observations                           150

Model Test User Model:

  Test statistic                                 0.014
  Degrees of freedom                                 0

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  X =~
```

```
    W1      (lmb1)    0.987    0.085   11.575    0.000
    W2      (lmb2)    0.975    0.085   11.443    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Y ~
    X       (bet1)    0.693    0.264    2.624    0.009

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    X        (phi)    1.148    0.078   14.757    0.000
   .Y        (psi)    9.776    1.153    8.481    0.000
   .W1      (omeg)    0.817    0.094    8.660    0.000
   .W2      (omeg)    0.817    0.094    8.660    0.000
```

Except for the warning message, everything seems to be fine. However, it's not fine! The parameters of this model are not identifiable, and as in the previous example (Example 0.10.2), the MLE is not unique. At first glance, it's not obvious why.

The matrix equation (45) gives the covariance matrix of $(W_{i,1}, W_{i,2}, Y_i)^\top$, expressing the six covariance structure equations in six unknowns, in a compact form.

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{pmatrix} = \begin{pmatrix} \lambda_1^2\phi + \omega & \lambda_1\lambda_2\phi & \lambda_1\beta_1\phi \\ & \lambda_2^2\phi + \omega & \lambda_2\beta_1\phi \\ & & \beta_1^2\phi + \psi \end{pmatrix}. \tag{45}$$

First, it is clear that if just one of $\lambda_1 = 0$, $\lambda_2 = 0$ or $\beta_1 = 0$, the zero value would be detectable from the covariance matrix, making that parameter identifiable. However, the remaining four equations in five unknowns would fail the parameter count rule, so that the other parameters would not be identifiable. If two or three of $\lambda_1$, $\lambda_2$ and $\beta_1$ were equal to zero, it would be impossible to tell which ones they were. Solving the remaining three equations in six unknowns is a hopeless task, and the entire parameter vector would be non-identifiable.

All these identifiability problems are local, and would have no effect on numerical maximum likelihood unless the true parameter values in question were zero. So consider points in the parameter space where $\lambda_1$, $\lambda_2$ and $\beta_1$ are all non-zero. In this case, $\omega$ and $\psi$ are identifiable, because

$$\omega = \sigma_{11} - \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \quad \text{and} \quad \psi = \sigma_{33} - \frac{\sigma_{13}\sigma_{23}}{\sigma_{12}}.$$

In fact, $\omega$ is over-identified, and this imposes the testable constraint $\sigma_{11} = \sigma_{22}$ on the covariance matrix, even though the `Model Test` degrees of freedom equal zero in the output. As for the other parameters, let $\boldsymbol{\theta}_1$ be an arbitrary point in the parameter space. Letting $c \neq 0$, consider the two parameter vectors

| | | | | | | |
|---|---|---|---|---|---|---|
| $\boldsymbol{\theta}_1$ | $\lambda_1$ | $\lambda_2$ | $\beta_1$ | $\phi$ | $\omega$ | $\psi$ |
| $\boldsymbol{\theta}_c$ | $c\lambda_1$ | $c\lambda_2$ | $c\beta_1$ | $\frac{\phi}{c^2}$ | $\omega$ | $\psi$ |

(46)

It is clear that $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_c$ both yield the same covariance matrix (45), and hence the same value of the likelihood function. In fact, every point in the parameter space belongs to an infinite family $\{\boldsymbol{\theta}_c : c \neq 0\}$ whose members all have the same the same likelihood. This means that if a numerical search locates a minimum, that point is just one of an infinite number of points in the parameter space where that same minimum value is attained. Furthermore the set is connected, and we are back to the river valley picture of Example 0.10.2.

A good way to confirm this account of what's happening is to choose a different set of starting values. Then, the numerical search should trickle downhill into the valley until it reaches a different point on the likelihood river. The estimated parameters should be very different (except for $\psi$ and $\omega$), but the value of the likelihood function (the height of the point on the river) should be the same. In the first test, I will try to start the search exactly in the river, at a point fairly distant from the first MLE. If the map provided by the table in (46) is correct, this should work.

To specify starting value of a regression coefficient in `lavaan`, one replaces the coefficient with `start(`*number*`)`, where *number* is a numeric starting value. A generic example is `Y~start(4.2)*X`. This is excellent when you are letting `lavaan` name parameters automatically, but what if you want to also name the regression coefficient? Somewhat oddly, you specify the connection between $X$ and $Y$ twice, and `lavaan` picks up the information in two passes through the syntax. The generic example would look like this: `Y~beta*X + start(4.2)*X`. A similar syntax works for variances, like this: `Y~~sigmasq*Y + start(1.0)*Y`.

Since the estimated $\beta_1$ for model `dmodel3` was positive, we will make it negative this time. As far as I can tell, the starting values have to be literal numbers, and not R variables.

```
> c = -2
> thetac = coef(dfit3); thetac
  beta1 lambda1 lambda2     phi     psi   omega   omega
  0.693   0.987   0.975   1.148   9.776   0.817   0.817

> thetac[1] = c*thetac[1]; thetac[2] = c*thetac[2]; thetac[3] = c*thetac[3]
> thetac[4] = thetac[4]/c^2
> cat(thetac)
-1.386474 -1.974219 -1.949046 0.2870302 9.775661 0.816833 0.816833
```

The `cat` function was used to get more decimal places in the output, because I needed to copy and paste the numbers into the model string. To start right in the river, we need as much accuracy as possible.

```
> dmodel3b = 'Y ~ beta1*X + start(-1.386474)*X
+             X =~ lambda1*W1 + start(-1.974219)*W1 +
+                  lambda2*W2 + start(-1.949046)*W2
+             # Variances (covariances would go here too)
```

```
+             X~~phi*X + start(0.2870302)*X      # Var(X) = phi
+             Y~~psi*Y + start(9.775661)*Y       # Var(epsilon) = psi
+             W1~~omega*W1 + start(0.816833)*W1  # Var(e1) = omega
+             W2~~omega*W2 + start(0.816833)*W2  # Var(e2) = omega
+             '
> dfit3b = lavaan(dmodel3b, data=babydouble)
```

There is a warning about a near-zero eigenvalue, similar to the last one. Then,

```
> show(dfit3b)
lavaan 0.6-7 ended normally after 2 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                          7
  Number of equality constraints                     1

  Number of observations                           150

Model Test User Model:

  Test statistic                                 0.014
  Degrees of freedom                                 0
```

This time the search found a minimum in two iterations rather than 19. The value of `Test Statistic` is the same as last time, suggesting that the height of the minus log likelihood function is the same with the new starting values.

Binding the starting and ending values into a matrix for easy inspection, we see that they are identical, at least to R's accuracy of display. This means that essentially, we started the numerical search at one of the infinitely many MLEs — as planned.

```
> rbind(thetac,coef(dfit3b))


           beta1   lambda1   lambda2       phi      psi    omega    omega
thetac -1.386474 -1.974219 -1.949046 0.2870302 9.775661 0.816833 0.816833
       -1.386474 -1.974219 -1.949046 0.2870302 9.775661 0.816833 0.816833
```

Also as expected, the parameter estimates are quite different from the first set we located, except for the estimates of the identifiable parameters $\psi$ and $\omega$.

```
> rbind(coef(dfit3),coef(dfit3b))
          beta1    lambda1    lambda2       phi      psi    omega    omega
[1,]  0.6932368  0.9871093  0.9745232 1.1481206 9.775661 0.816833 0.816833
[2,] -1.3864740 -1.9742186 -1.9490464 0.2870302 9.775661 0.816833 0.816833
```

Though the locations of the MLEs are different, the log likelihood at those points is the same. Again, the theoretical analysis is confirmed.

```
> c( logLik(dfit3), logLik(dfit3b) )
[1] -878.5155 -878.5155
```

In one last variation, the search starts fairly close to the river[38] but not exactly on target, and finds its way to yet another MLE. Here, starting values are provided for $\lambda_1$, $\lambda_2$, $\beta_1$ and $\phi$. `lavaan` provides starting values for $\psi$ and $\omega$.

```
> dmodel3c = 'Y ~ beta1*X + start(6)*X
            X =~ lambda1*W1 + start(8)*W1 +
                 lambda2*W2  + start(8)*W2
            # Variances (covariances would go here too)
            X~~phi*X + start(1/64)*X      # Var(X) = phi
            Y~~psi*Y       # Var(epsilon) = psi
            W1~~omega*W1   # Var(e1) = omega
            W2~~omega*W2   # Var(e2) = omega
            '
> dfit3c = lavaan(dmodel3c, data=babydouble)
Warning message:
In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING:
    The variance-covariance matrix of the estimated parameters (vcov)
    does not appear to be positive definite! The smallest eigenvalue
    (= 1.285532e-12) is close to zero. This may be a symptom that the
    model is not identified.

> c( logLik(dfit3), logLik(dfit3b), logLik(dfit3b) )
[1] -878.5155 -878.5155 -878.5155

> rbind( coef(dfit3), coef(dfit3b), coef(dfit3c) )
          beta1    lambda1    lambda2       phi      psi    omega    omega
[1,]  0.6932368  0.9871093  0.9745232 1.1481206 9.775661 0.816833 0.816833
[2,] -1.3864740 -1.9742186 -1.9490464 0.2870302 9.775661 0.816833 0.816833
[3,]  5.7803725  8.2307505  8.1258046 0.0165135 9.775661 0.816833 0.816833
```

So the search located another point with the same maximum log likelihood, fairly far from the other two. For the parameters that are not identifiable, the answer depends on the starting value.

When the parameters of a model are all identifiable, the minus log likelihood should have a unique global minimum, and `lavaan`'s default starting values should be adequate

[38] To find a point that is "fairly close," observe from (46) that the product $\lambda_1\lambda_2\phi$ must be constant for all points on the river. The constant is pretty close to 1, and $\beta_1$ should be around 3/4 of $\lambda_1$. So $\beta_1 = 6$, $\lambda_1 = \lambda_2 = 8$ and $\phi = 1/64$ should do it.

most of the time. However even when the parameters are identifiable, local maxima and minima are possible. If you suspect the search may have located a local minimum (perhaps because some of the MLEs are extremely large), you may need to specify your own starting values. Try several sets. The `parTable` function can be used to verify that the starting values were the ones you intended. In the display below, `ustart` are the starting values given by the user, some of which are `NA` because they were not specified. The `start` column are the starting values used by the software, and the `est` column (estimates) is where the search ended — at the parameter estimates.

```
> parTable(dfit3c)
  id  lhs op  rhs user block group free ustart exo   label plabel start   est    se
1  1    Y  ~    X    1     1     1    1  6.000   0   beta1   .p1. 6.000 5.780 1.895
2  2    X =~   W1    1     1     1    2  8.000   0 lambda1   .p2. 8.000 8.231 0.822
3  3    X =~   W2    1     1     1    3  8.000   0 lambda2   .p3. 8.000 8.126 0.819
4  4    X ~~    X    1     1     1    4  0.016   0     phi   .p4. 0.016 0.017 0.004
5  5    Y ~~    Y    1     1     1    5     NA   0     psi   .p5. 5.164 9.776 1.153
6  6   W1 ~~   W1    1     1     1    6     NA   0   omega   .p6. 0.968 0.817 0.094
7  7   W2 ~~   W2    1     1     1    7     NA   0   omega   .p7. 0.953 0.817 0.094
8  8 .p6. == .p7.    2     0     0    0     NA   0                0.000 0.000 0.000
```

### 0.10.3 The Double Measurement Design in Matrix Form

Consider the general case of regression with measurement error in both the explanatory variables and the response variables. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} \mathbf{w}_{i,1} &= \boldsymbol{\nu_1} + \mathbf{x}_i + \mathbf{e}_{i,1} \\ \mathbf{v}_{i,1} &= \boldsymbol{\nu_2} + \mathbf{y}_i + \mathbf{e}_{i,2} \\ \mathbf{w}_{i,2} &= \boldsymbol{\nu_3} + \mathbf{x}_i + \mathbf{e}_{i,3} \\ \mathbf{v}_{i,2} &= \boldsymbol{\nu_4} + \mathbf{y}_i + \mathbf{e}_{i,4}, \\ \mathbf{y}_i &= \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{x}_i + \boldsymbol{\epsilon}_i \end{aligned} \tag{47}$$

where

$\mathbf{y}_i$ is a $q \times 1$ random vector of latent response variables. Because $q$ can be greater than one, the regression is multivariate.

$\boldsymbol{\beta}$ is a $q \times p$ matrix of unknown constants. These are the regression coefficients, with one row for each response variable and one column for each explanatory variable.

$\mathbf{x}_i$ is a $p \times 1$ random vector of latent explanatory variables, with expected value zero and variance-covariance matrix $\boldsymbol{\Phi}$, a $p \times p$ symmetric and positive definite matrix of unknown constants.

$\boldsymbol{\epsilon}_i$ is the error term of the latent regression. It is a $q \times 1$ random vector with expected value zero and variance-covariance matrix $\boldsymbol{\Psi}$, a $q \times q$ symmetric and positive definite matrix of unknown constants.

$\mathbf{w}_{i,1}$ and $\mathbf{w}_{i,2}$ are $p \times 1$ observable random vectors, each consisting of $\mathbf{x}_i$ plus random error and a set of constant terms that represent *measurement bias*[39].

$\mathbf{v}_{i,1}$ and $\mathbf{v}_{i,2}$ are $q \times 1$ observable random vectors, each consisting of $\mathbf{y}_i$ plus random error and measurement bias.

$\mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,1}$ are the measurement errors in $\mathbf{w}_{i,1}, \mathbf{v}_{i,1}, \mathbf{w}_{i,2}$ and $\mathbf{v}_{i,2}$ respectively. Joining the vectors of measurement errors into a single long vector $\mathbf{e}_i$, its covariance matrix may be written as a partitioned matrix

$$cov(\mathbf{e}_i) = cov \begin{pmatrix} \mathbf{e}_{i,1} \\ \mathbf{e}_{i,2} \\ \mathbf{e}_{i,3} \\ \mathbf{e}_{i,4} \end{pmatrix} = \left(\begin{array}{c|c|c|c} \boldsymbol{\Omega}_{11} & \boldsymbol{\Omega}_{12} & \mathbf{0} & \mathbf{0} \\ \hline \boldsymbol{\Omega}_{12}^{\top} & \boldsymbol{\Omega}_{22} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \boldsymbol{\Omega}_{33} & \boldsymbol{\Omega}_{34} \\ \hline \mathbf{0} & \mathbf{0} & \boldsymbol{\Omega}_{34}^{\top} & \boldsymbol{\Omega}_{44} \end{array}\right) = \boldsymbol{\Omega}.$$

The matrices of covariances between $\mathbf{x}_i, \boldsymbol{\epsilon}_i$ and $\mathbf{e}_i$ are all zero.

$\boldsymbol{\alpha}$, $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$, $\boldsymbol{\nu}_3$ and $\boldsymbol{\nu}_4$ are vectors of constants.

$E(\mathbf{x}_i) = \boldsymbol{\mu}_x$.

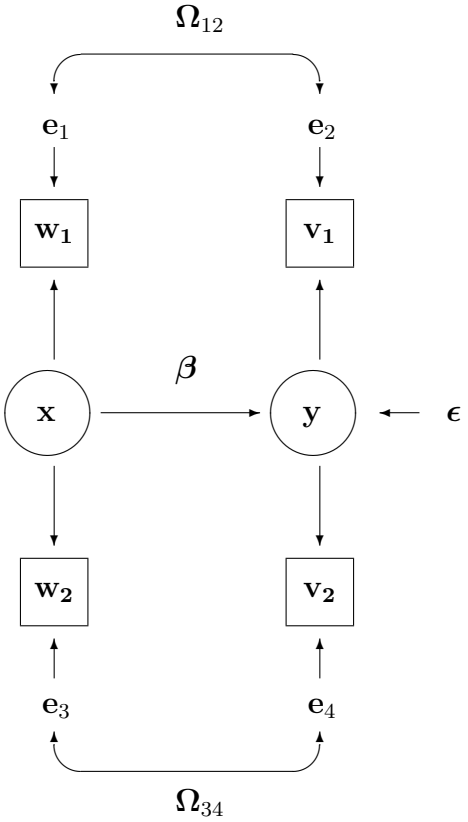
The main idea of the Double Measurement Design is that every variable is measured by two different methods. Errors of measurement may be correlated within measurement methods, but not between methods. So for example, farmers who overestimate their number of pigs may also overestimate their number of cows. On the other hand, if the number of pigs is counted once by the farm manager at feeding time and on another occasion by a research assistant from an areal photograph, then it would be fair to assume that the errors of measurement for the different methods are uncorrelated. In general, correlation within measurement methods is almost unavoidable. The ability of the double measurement model to admit the existence of correlated measurement error and still be identifiable is a real advantage.

In symbolic terms, $\mathbf{e}_{i,1}$ is error in measuring the explanatory variables by method one, and $\mathbf{e}_{i,2}$ is error in measuring the response variables by method one. $cov(\mathbf{e}_{i,1}) = \boldsymbol{\Omega}_{11}$ need not be diagonal, so method one's errors of measurement for the explanatory variables may be correlated with one another. Similarly, $cov(\mathbf{e}_{i,2}) = \boldsymbol{\Omega}_{22}$ need not be diagonal, so method one's errors of measurement for the response variables may be correlated with one another. And, errors of measurement using the same method may be correlated between the explanatory and response variables. For method one, this is represented by the matrix $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) = \boldsymbol{\Omega}_{12}$. The same pattern holds for method two. On the other hand, $\mathbf{e}_{i,1}$ and $\mathbf{e}_{i,2}$ are each uncorrelated with both $\mathbf{e}_{i,3}$ and $\mathbf{e}_{i,4}$.

To emphasize an important practical point, the matrices $\boldsymbol{\Omega}_{11}$ and $\boldsymbol{\Omega}_{33}$ must be of the same dimension, just as $\boldsymbol{\Omega}_{22}$ and $\boldsymbol{\Omega}_{44}$ must be of the same dimension – but none of the corresponding elements have to be equal. In particular, the corresponding diagonal

[39]For example, if one of the elements of $\mathbf{w}_{i,1}$ is reported amount of exercise, the corresponding element of $\boldsymbol{\nu}_1$ would be the average amount by which people exaggerate how much they exercise.

Figure 15: The Double Measurement Model




elements may be unequal. This means that measurements of a variable by two different methods do not need to be equally precise.

The model is depicted in Figure 15. It follows the usual conventions for path diagrams of structural equation models. Straight arrows go from *exogenous* variables (that is, explanatory variables, those on the right-hand side of equations) to *endogenous* varables (response variables, those on the left side). Correlations among exogenous variables are represented by two-headed curved arrows. Observable variables are enclosed by rectangles or squares, while latent variables are enclosed by ellipses or circles. Error terms are not enclosed by anything.

**Parameter identifiability** As usual in structural equation models, the moments (specifically, the expected values and variance-covariance matrix) of the observable data are functions of the model parameters. If the model parameters are also functions of the moments, then they are identifiable[40]. For the double measurement model, the parameters appearing in the covariance matrix of the observable variables are identifiable, but the parameters appearing only in the mean vector are not. Accordingly, we split the job

[40]Meaning identifiable from the moments. For multivariate normal models and also in general practice, a parameter is identifiable from the mean vector and covariance matrix, or not at all.

into two parts, starting with the covariance matrix. The first part is typical of easier proofs for structural equation models. The goal is to solve for the model parameters in terms of elements of the variance-covariance matrix of the observable data. This shows the parameters are functions of the distribution, so that no two distinct parameter values could yield the same distribution of the observed data.

Collecting $\mathbf{w}_{i,1}$, $\mathbf{v}_{i,1}$, $\mathbf{w}_{i,2}$ and $\mathbf{v}_{i,2}$ into a single long data vector $\mathbf{d}_i$, we write its variance-covariance matrix as a partitioned matrix:

$$\boldsymbol{\Sigma} = \left(\begin{array}{c|c|c|c} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} & \boldsymbol{\Sigma}_{13} & \boldsymbol{\Sigma}_{14} \\ \hline & \boldsymbol{\Sigma}_{22} & \boldsymbol{\Sigma}_{23} & \boldsymbol{\Sigma}_{24} \\ \hline & & \boldsymbol{\Sigma}_{33} & \boldsymbol{\Sigma}_{34} \\ \hline & & & \boldsymbol{\Sigma}_{44} \end{array}\right),$$

where the covariance matrix of $\mathbf{w}_{i,1}$ is $\boldsymbol{\Sigma}_{11}$, the covariance matrix of $\mathbf{v}_{i,1}$ is $\boldsymbol{\Sigma}_{22}$, the matrix of covariances between $\mathbf{w}_{i,1}$ and $\mathbf{v}_{i,1}$ is $\boldsymbol{\Sigma}_{12}$, and so on.

Now we express all the $\boldsymbol{\Sigma}_{ij}$ sub-matrices in terms of the parameter matrices of Model (47) by straightforward variance-covariance calculations. Students may be reminded that things go smoothly if one substitutes for everything in terms of explanatory variables and error terms before actually starting to calculate covariances. For example,

$$\begin{aligned}
\boldsymbol{\Sigma}_{12} &= cov(\mathbf{w}_{i,1}, \mathbf{v}_{i,1}) \\
&= cov\left(\boldsymbol{\nu_1} + \mathbf{x}_i + \mathbf{e}_{i,1},\ \boldsymbol{\nu_2} + \mathbf{y}_i + \mathbf{e}_{i,2}\right) \\
&= cov\left(\boldsymbol{\nu_1} + \mathbf{x}_i + \mathbf{e}_{i,1},\ \boldsymbol{\nu_2} + \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{x}_i + \boldsymbol{\epsilon}_i + \mathbf{e}_{i,2}\right) \\
&= cov\left(\mathbf{x}_i + \mathbf{e}_{i,1},\ \boldsymbol{\beta}\mathbf{x}_i + \boldsymbol{\epsilon}_i + \mathbf{e}_{i,2}\right) \\
&= cov(\mathbf{x}_i, \boldsymbol{\beta}\mathbf{x}_i) + cov(\mathbf{x}_i, \boldsymbol{\epsilon}_i) + cov(\mathbf{x}_i, \mathbf{e}_{i,2}) + cov(\mathbf{e}_{i,1}, \boldsymbol{\beta}\mathbf{x}_i) + cov(\mathbf{e}_{i,1}, \boldsymbol{\epsilon}_i) + cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) \\
&= cov(\mathbf{x}_i, \mathbf{x}_i)\boldsymbol{\beta}^\top + 0 + 0 + 0 + 0 + \boldsymbol{\Omega}_{12} \\
&= \boldsymbol{\Phi}\boldsymbol{\beta}^\top + \boldsymbol{\Omega}_{12}.
\end{aligned}$$

In this manner, we obtain the partitioned covariance matrix of the observable data $\mathbf{d}_i = (\mathbf{w}_{i,1}^\top, \mathbf{v}_{i,1}^\top, \mathbf{w}_{i,2}^\top, \mathbf{v}_{i,2}^\top)^\top$ as

$$\begin{aligned}
\boldsymbol{\Sigma} &= \left(\begin{array}{c|c|c|c} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} & \boldsymbol{\Sigma}_{13} & \boldsymbol{\Sigma}_{14} \\ \hline & \boldsymbol{\Sigma}_{22} & \boldsymbol{\Sigma}_{23} & \boldsymbol{\Sigma}_{24} \\ \hline & & \boldsymbol{\Sigma}_{33} & \boldsymbol{\Sigma}_{34} \\ \hline & & & \boldsymbol{\Sigma}_{44} \end{array}\right) \\
&= \left(\begin{array}{c|c|c|c} \boldsymbol{\Phi} + \boldsymbol{\Omega}_{11} & \boldsymbol{\Phi}\boldsymbol{\beta}^\top + \boldsymbol{\Omega}_{12} & \boldsymbol{\Phi} & \boldsymbol{\Phi}\boldsymbol{\beta}^\top \\ \hline & \boldsymbol{\beta}\boldsymbol{\Phi}\boldsymbol{\beta}^\top + \boldsymbol{\Psi} + \boldsymbol{\Omega}_{22} & \boldsymbol{\beta}\boldsymbol{\Phi} & \boldsymbol{\beta}\boldsymbol{\Phi}\boldsymbol{\beta}^\top + \boldsymbol{\Psi} \\ \hline & & \boldsymbol{\Phi} + \boldsymbol{\Omega}_{33} & \boldsymbol{\Phi}\boldsymbol{\beta}^\top + \boldsymbol{\Omega}_{34} \\ \hline & & & \boldsymbol{\beta}\boldsymbol{\Phi}\boldsymbol{\beta}^\top + \boldsymbol{\Psi} + \boldsymbol{\Omega}_{44} \end{array}\right)
\end{aligned} \tag{48}$$

The equality (48) corresponds to a system of ten matrix equations in nine matrix unknowns. The unknowns are the parameter matrices of Model (47): $\boldsymbol{\Phi}$, $\boldsymbol{\beta}$, $\boldsymbol{\Psi}$, $\boldsymbol{\Omega}_{11}$, $\boldsymbol{\Omega}_{22}$, $\boldsymbol{\Omega}_{33}$, $\boldsymbol{\Omega}_{44}$, $\boldsymbol{\Omega}_{12}$, and $\boldsymbol{\Omega}_{34}$. In the solution below, notice that once a parameter has been

identified, it may be used to solve for other parameters without explicitly substituting in terms of $\boldsymbol{\Sigma}_{ij}$ quantities. Sometimes a full explicit solution is useful, but to show identifiability all you need to do is show that the moment structure equations *can* be solved.

$$\begin{array}{rcl} \boldsymbol{\Phi} & = & \boldsymbol{\Sigma}_{13} \\ \boldsymbol{\beta} & = & \boldsymbol{\Sigma}_{23}\boldsymbol{\Phi}^{-1} = \boldsymbol{\Sigma}_{14}^{\top}\boldsymbol{\Phi}^{-1} \\ \boldsymbol{\Psi} & = & \boldsymbol{\Sigma}_{24} - \boldsymbol{\beta}\boldsymbol{\Phi}\boldsymbol{\beta}^{\top} \\ \boldsymbol{\Omega}_{11} & = & \boldsymbol{\Sigma}_{11} - \boldsymbol{\Phi} \\ \boldsymbol{\Omega}_{22} & = & \boldsymbol{\Sigma}_{22} - \boldsymbol{\beta}\boldsymbol{\Phi}\boldsymbol{\beta}^{\top} - \boldsymbol{\Psi} \\ \boldsymbol{\Omega}_{33} & = & \boldsymbol{\Sigma}_{33} - \boldsymbol{\Phi} \\ \boldsymbol{\Omega}_{44} & = & \boldsymbol{\Sigma}_{44} - \boldsymbol{\beta}\boldsymbol{\Phi}\boldsymbol{\beta}^{\top} - \boldsymbol{\Psi} \\ \boldsymbol{\Omega}_{12} & = & \boldsymbol{\Sigma}_{12} - \boldsymbol{\Phi}\boldsymbol{\beta}^{\top} \\ \boldsymbol{\Omega}_{34} & = & \boldsymbol{\Sigma}_{34} - \boldsymbol{\Phi}\boldsymbol{\beta}^{\top} \end{array} \tag{49}$$

The solution (49) shows that the parameters appearing in the covariance matrix $\boldsymbol{\Sigma}$ are identifiable. This includes the critical parameter matrix $\boldsymbol{\beta}$, which determines the connection between explanatory variables and response variables.

## Intercepts

In Model (47), let $\boldsymbol{\mu} = E(\mathbf{d}_i)$. This vector of expected values may be written as a partitioned vector, as follows.

$$\boldsymbol{\mu} = \left(\begin{array}{c} \boldsymbol{\mu}_1 \\ \hline \boldsymbol{\mu}_2 \\ \hline \boldsymbol{\mu}_3 \\ \hline \boldsymbol{\mu}_4 \end{array}\right) = \left(\begin{array}{c} E(\mathbf{w}_{i,1}) \\ \hline E(\mathbf{v}_{i,1}) \\ \hline E(\mathbf{w}_{i,2}) \\ \hline E(\mathbf{v}_{i,2}) \end{array}\right) = \left(\begin{array}{c} \boldsymbol{\nu}_1 + \boldsymbol{\mu}_x \\ \hline \boldsymbol{\nu}_2 + \boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{\mu}_x \\ \hline \boldsymbol{\nu}_3 + \boldsymbol{\mu}_x \\ \hline \boldsymbol{\nu}_4 + \boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{\mu}_x \end{array}\right). \tag{50}$$

The parameters that appear in $\boldsymbol{\mu}$ but not $\boldsymbol{\Sigma}$ are contained in $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$, $\boldsymbol{\nu}_3$, $\boldsymbol{\nu}_4$, $\boldsymbol{\mu}_x$ and $\boldsymbol{\alpha}$. To identify these parameters, one would need to solve the equations in (50) uniquely for these six parameter vectors. Even with $\boldsymbol{\beta}$ considered known and fixed because it is identified in (49), this is impossible in most of the parameter space, because (50) specifies $2m + 2p$ equations in $3m + 3p$ unknowns.

It is tempting to assume the measuremant bias terms $\boldsymbol{\nu}_1 \ldots, \boldsymbol{\nu}_4$ to be zero; this would allow identification of $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}_x$. Unfortunately, it is doubtful that such an assumption could be justified very often in practice. Most of the time, all we can do is identify the parameter matrices that appear in the covariance matrix, and also the *functions* $\boldsymbol{\mu_1} \ldots, \boldsymbol{\mu_4}$ of the parameters as given in equation (50). This can be viewed as a re-parameterization of the model. In practice, the functions $\boldsymbol{\mu_1} \ldots, \boldsymbol{\mu_4}$ of the parameters are usually not of much interest. They are estimated by the corresponding sample means, conveniently forgotten, and almost never mentioned.

To summarize, the parameters appearing in the covariance matrix are identifiable. This includes $\boldsymbol{\beta}$, the quantity of primary interest. Means and intercepts are not identifiable, but they are absorbed in a re-parameterization and set aside. It's no great loss. In practice, if data are collected following the double measurement recipe, then the data analysis may proceed with no worries about parameter identifiability.

For the double measurement model, there are more covariance structure equations than unknowns. Thus the model is over-identified, and testable. Notice in the covariance structure equations (48), that $\boldsymbol{\Sigma}_{14} = \boldsymbol{\Sigma}_{23}^\top$. As in the scalar Example 0.10.1 (see page 63), this constraint on the covariance matrix $\boldsymbol{\Sigma}$ arises from the model, and provides a way to test whether the model is correct. These $pq$ equalities are not the only ones implied by the model. Because $\boldsymbol{\Sigma}_{13} = \boldsymbol{\Phi}$, the $p \times p$ matrix of covariances $\boldsymbol{\Sigma}_{13}$ is actually a covariance matrix, so it is symmetric. This implies $p(p-1)/2$ more equalities.

## Estimation and testing

**Normal model** As in Example 0.10.1, the (collapsed) expected values are estimated by the corresponding vector of sample means, and then set aside. Under a multivariate normal model, these terms literally disappear from the likelihood function (42) on page 65. The resulting likelihood is (43) on page 65. The full range of large-sample likelihood methods is available. Maximum likelihood estimates are asymptotically normal, and asymptotic standard errors are convenient by-products of the numerical minimization as described in Section A.6.3 of Appendix A; most software produces them by default. Dividing an estimated regression coefficient by its standard error gives a $Z$-test for whether the coefficient is different from zero. My experience is that likelihood ratio tests can substantially outperform both these $Z$-tests and the Wald tests that are their generalizations, especially when there is a lot of measurement error, the explanatory variables are strongly related to one another, and the sample size is not huge.

**Distribution-free** In presenting models for regression with measurement error, it is often convenient to assume that everything is multivariate normal. This is especially true when giving examples of models where the parameters are *not* identifiable. But normality is not necessary. Suppose Model (47) holds, and that the distributions of of the latent explanatory variables and error terms are unknown, except that they possess covariance matrices, with $\mathbf{e}_{i,1}$ and $\mathbf{e}_{i,2}$ having zero covariance with $\mathbf{e}_{i,3}$ and $\mathbf{e}_{i,4}$. In this case the parameter of the model could be expressed as $\theta = (\boldsymbol{\beta}, \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{\Omega}, F_{\mathbf{x}}, F_{\boldsymbol{\epsilon}}, F_{\mathbf{e}})$, where $F_{\mathbf{x}}$, $F_{\boldsymbol{\epsilon}}$ and $F_{\mathbf{e}}$ are the (joint) cumulative distribution functions of $\mathbf{x}_i$, $\boldsymbol{\epsilon}_i$ and $\mathbf{e}_i$ respectively.

Note that the parameter in this "non-parametric" problem is of infinite dimension, but that presents no conceptual difficulty. The probability distribution of the observed data is still a function of the parameter vector, and to show identifiability, we would have to be able to recover the parameter vector from the probability distribution of the data. While in general we cannot recover the whole thing, we certainly can recover a useful *function* of the parameter vector, namely $\boldsymbol{\beta}$. In fact, $\boldsymbol{\beta}$ is the only quantity of interest; the remainder of the parameter vector consists only of nuisance parameters, whether it is of finite dimension or not.

To make the reasoning explicit, the covariance matrix $\boldsymbol{\Sigma}$ is a function of the probability distribution of the observed data, whether that probability distribution is normal or not. The calculations leading to (49) still hold, showing that $\boldsymbol{\beta}$ is a function of $\boldsymbol{\Sigma}$, and hence of the probability distribution of the data. Therefore, $\boldsymbol{\beta}$ is identifiable.

This is all very well, but can we actually *do* anything without knowing what the distributions are? Certainly! Looking at (49), one is tempted to just put hats on everything to obtain Method-of-Moments estimators. However, we can do a little better. Note that while $\boldsymbol{\Phi} = \boldsymbol{\Sigma}_{12}$ is a symmetric matrix in the population and $\widehat{\boldsymbol{\Sigma}}_{12}$ *converges* to a symmetric matrix, $\widehat{\boldsymbol{\Sigma}}_{12}$ will be non-symmetric for any finite sample size (with probability one if the distributions involved are continuous). A better estimator is obtained by averaging pairs of off-diagonal elements:

$$\widehat{\boldsymbol{\Phi}}_M = \frac{1}{2}(\widehat{\boldsymbol{\Sigma}}_{13} + \widehat{\boldsymbol{\Sigma}}_{13}^\top), \tag{51}$$

where the subscript $M$ indicates a Method-of-Moments estimator. Using the second line of (49), a reasonable though non-standard estimator of $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}}_M = \frac{1}{2}\left(\widehat{\boldsymbol{\Sigma}}_{14}^\top + \widehat{\boldsymbol{\Sigma}}_{23}\right)\widehat{\boldsymbol{\Phi}}_M^{-1} \tag{52}$$

Consistency follows from the Law of Large Numbers and a continuity argument. All this assumes the existence only of second moments and cross-moments. With the assumption of fourth moments (so that sample variances possess variances), Theorem A.1 in Appendix A, combined with the multivariate delta method, provides a basis for large-sample interval estimation and testing.

However, there is no need to bother. As described in Chapter 5, the normal-theory tests and confidence intervals for $\boldsymbol{\beta}$ can be trusted when the data are not normal. Note that this does not extend to the other model parameters. For example, if the vector of latent variables $\mathbf{x}_i$ is not normal, then normal-theory inference about its covariance matrix will be flawed. In any event, the estimation method of choice will maximum likelihood, with interpretive focus on the regression coefficients in $\boldsymbol{\beta}$ rather than on the other model parameters.

### 0.10.4 The BMI Health Study

Body mass index (BMI) is defined as weight in kilograms divided by height in meters squared. It represents weight relative to height, and is a measure of how thick, or hefty a person is. People with a BMI less than 18 are described as underweight, those over 25 are described as overweight, and those over 30 are described as obese. However, many professional athletes have BMI numbers in the overweight range.

High BMI tends to be associated with poor health, and with indicators such as high blood pressure and high cholesterol. However, people with high BMI also tend to be older and fatter. Perhaps age and physical condition are responsible for the association of BMI to health. The natural idea is to look at the connection of BMI to health indicators, controlling for age and some indicator of physical condition like percent body fat. The problem is that percent body fat (and to a lesser extent, age) are measured with error. As

discussed in Section 0.7, standard ways of controlling for them with ordinary regression are highly suspect. The solution is double measurement regression.

**Example 0.10.4** *The BMI health study*[41]

In this study, there are five latent variables. Each one was were measured twice, by different personnel at different locations and mostly by different methods. The variables are age, BMI, percent body fat, cholesterol level, and diastolic blood pressure.

- In measurement set one, age was self report. In measurement set two, age was based on a passport or birth certificate.
- In measurement set one, the height and weight measurements making up BMI were conducted in a doctor's office, following no special procedures. In measurement set two, they were conducted by a lab technician. Patients had to remove their shoes, and wore a hospital gown.
- In measurement set one, estimated percent body fat was based on measurements with tape and calipers, conducted in the doctor's office. In measurement set two, percent body fat was estimated by submerging the participant in a water tank (hydrostatic weighing).
- In measurement set one, serum (blood) cholesterol level was measured in lab 1. In measurement set two, it was measured in lab 2. There is no known difference between the labs in quality.
- In measurement set one, diastolic blood pressure was measured in the doctor's office using a standard manual blood pressure cuff. In measurement set two, blood pressure was measured in the lab by a digital device, and was mostly automatic.

Measurement set two was of generally higher quality than measurement set one. Correlation of measurement errors is possible within sets, but unlikely between sets.

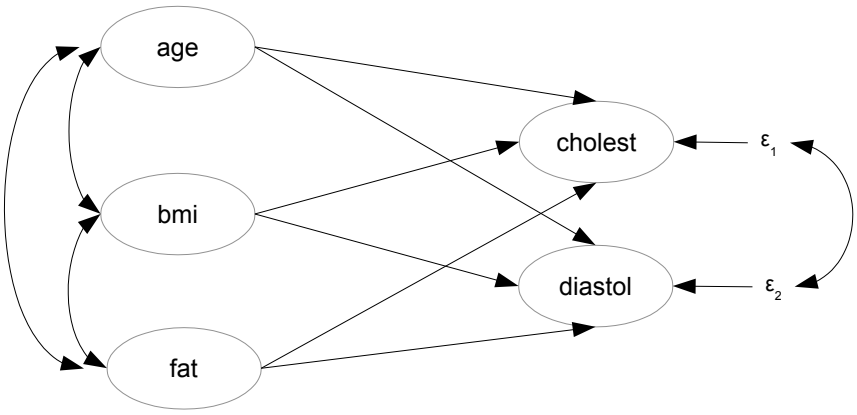
Figure 16 shows a regression model for the latent variables. Because all the variables are latent, they are enclosed in ovals. There are two response variables, so this is multivariate regression.

First, we read the data and take a look. The variables are self-explanatory. There are 500 cases.

```
 > bmidata = read.table("http://www.utstat.toronto.edu/~brunner/openSEM/data/bmi.data.txt")
> head(bmidata)
  age1 bmi1 fat1 cholest1 diastol1 age2 bmi2 fat2 cholest2 diastol2
1   63 24.5 16.5    195.4       38   60 23.9 20.1    203.5       66
2   42 13.0  1.9    184.3       86   44 14.8  2.6    197.3       78
3   32 22.5 14.6    354.1      104   33 21.7 20.4    374.3       73
```

[41] This study is fictitious, and the data come from a combination of random number generation and manual editing. As far as I know, nothing like this has actually been done. I believe it should be.

Figure 16: Latent variable model for the BMI health study




```
4   59 25.5 19.0    214.6       93   58 28.5 20.0    203.7      106
5   45 26.5 17.8    324.8       97   43 25.0 12.3    329.7       92
6   31 19.4 17.1    280.7       92   42 19.9 19.9    276.7       87
```

The standard, naive approach to analyzing these data is to ignore the possibility of measurement error, and use ordinary linear regression. One could either use just the better set of measurements (set 2), or average them. Averaging is a little better, because it improves reliability.

```
> age = (age1+age2)/2; bmi = (bmi1+bmi2)/2; fat = (fat1+fat2)/2
> cholest = (cholest1+cholest2)/2; diastol = (diastol1+diastol2)/2
```

There are two response variables (cholesterol level and diastolic blood pressure), so we fit a conventional multivariate linear model, and look at the multivariate test of BMI controlling for age and percent body fat. The full model has age, percent body fat and BMI, while the restricted model has just age and percent body fat.

```
> fullmod = lm( cbind(cholest,diastol) ~ age + fat + bmi)
> restrictedmod = update(fullmod, . ~ . - bmi) # Remove var(s) being tested
> anova(fullmod,restrictedmod) # Gives multivariate test.
Analysis of Variance Table

Model 1: cbind(cholest, diastol) ~ age + fat + bmi
Model 2: cbind(cholest, diastol) ~ age + fat
  Res.Df Df Gen.var.  Pillai approx F num Df den Df    Pr(>F)
1    496      591.89
2    497  1   599.36 0.02869   7.3106      2    495 0.0007431 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

The conclusion is that controlling for age and percent body fat, BMI is related to cholesterol, or diastolic blood pressure, or both. The `summary` function gives two sets of univariate output. Primary interest is in the $t$-tests for `bmi`.

```
> summary(fullmod) # Two sets of univariate output
Response cholest :

Call:
lm(formula = cholest ~ age + fat + bmi)

Residuals:
     Min       1Q   Median       3Q      Max
-148.550  -34.243    2.626   33.661  165.582

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)
(Intercept) 220.0610    21.0109  10.474 < 0.0000000000000002 ***
age          -0.2714     0.2002  -1.356              0.17578
fat           2.2334     0.5792   3.856              0.00013 ***
bmi           0.5164     1.0154   0.509              0.61128
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 52.43 on 496 degrees of freedom
Multiple R-squared:  0.09701,Adjusted R-squared:  0.09155
F-statistic: 17.76 on 3 and 496 DF,  p-value: 0.00000000005762


Response diastol :

Call:
lm(formula = diastol ~ age + fat + bmi)

Residuals:
    Min      1Q  Median      3Q     Max
-44.841  -7.140  -0.408   7.612  41.377

Coefficients:
            Estimate Std. Error t value             Pr(>|t|)
(Intercept) 49.69194    4.52512  10.981 < 0.0000000000000002 ***
age          0.12648    0.04311   2.934             0.003504 **
fat          0.64056    0.12474   5.135          0.000000406 ***
bmi          0.82627    0.21869   3.778             0.000177 ***
---
```

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 11.29 on 496 degrees of freedom
Multiple R-squared:  0.3333,Adjusted R-squared:  0.3293
F-statistic: 82.67 on 3 and 496 DF,  p-value: < 0.00000000000000022
```

For cholesterol, we have $t = 0.509$ and $p = 0.61128$. The conclusion is that controlling for age and percent body fat, there is no evidence of a connection between body mass index and serum cholesterol level.

For diastolic blood pressure, the test of BMI controlling for age and percent body fat is $t = 3.778$ and $p = 0.000177$. This time the conclusion is that even controlling for age and percent body fat, higher BMI is associated with higher average diastolic blood pressure – a bad sign for health. However, this "even controlling for" conclusion is exactly the kind of mistake that is often caused by ignoring measurement error; see Section 0.7. So, we specify a proper double measurement regression model. The names of latent variables begin with `L`. I did this because I'd already used the natural names like `age`, `bmi` and `cholest` earlier, and I wanted to avoid accidental conflicts.

```
bmimodel1 =
    ############################################################
    # Latent variable model
    # ---------------------
   'Lcholest ~ beta11*Lage + beta12*Lbmi + beta13*Lfat
    Ldiastol ~ beta21*Lage + beta22*Lbmi + beta23*Lfat
    #
    # Measurement model
    # -----------------
    Lage =~ 1*age1 + 1*age2
    Lbmi =~ 1*bmi1 + 1*bmi2
    Lfat =~ 1*fat1 +1*fat2
    Lcholest =~ 1*cholest1 + 1*cholest2
    Ldiastol =~ 1*diastol1 + 1*diastol2
    #
    # Variances and covariances
    # -------------------------
    # Of latent explanatory variables
    Lage ~~ phi11*Lage; Lage ~~ phi12*Lbmi; Lage ~~ phi13*Lfat
                        Lbmi ~~ phi22*Lbmi; Lbmi ~~ phi23*Lfat
                                            Lfat ~~ phi33*Lfat
    # Of error terms in latent the regression (epsilon_ij)
    Lcholest ~~ psi11*Lcholest; Lcholest ~~ psi12*Ldiastol
                                Ldiastol ~~ psi22*Ldiastol
    # Of measurement errors (e_ijk) for measurement set 1
    age1 ~~ w111*age1; age1 ~~ w112*bmi1; age1 ~~ w113*fat1;
    age1 ~~ w114*cholest1; age1 ~~ w115*diastol1
            bmi1 ~~ w122*bmi1; bmi1 ~~ w123*fat1; bmi1 ~~ w124*cholest1; bmi1 ~~ w125*diastol1
                    fat1 ~~ w133*fat1; fat1 ~~ w134*cholest1; fat1 ~~ w135*diastol1
                            cholest1 ~~ w144*cholest1; cholest1 ~~ w145*diastol1
                                                       diastol1 ~~ w155*diastol1
    # Of measurement errors (e_ijk) for measurement set 2
```

```
    age2 ~~ w211*age2; age2 ~~ w212*bmi2; age2 ~~ w213*fat2;
    age2 ~~ w214*cholest2; age2 ~~ w215*diastol2
           bmi2 ~~ w222*bmi2; bmi2 ~~ w223*fat2; bmi2 ~~ w224*cholest2; bmi2 ~~ w225*diastol2
                  fat2 ~~ w233*fat2; fat2 ~~ w234*cholest2; fat2 ~~ w235*diastol2
                         cholest2 ~~ w244*cholest2; cholest2 ~~ w245*diastol2
                                                    diastol2 ~~ w255*diastol2
  ' ################## End of bmimodel1 ##################
```

When we try to fit this perfectly nice model, there is trouble.

```
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
> fit1 = lavaan(bmimodel1, data=bmidata)
Warning message:
Warning messages:
1: In lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable,  :
  lavaan WARNING: the optimizer warns that a solution has NOT been found!
2: In lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable,  :
  lavaan WARNING: the optimizer warns that a solution has NOT been found!
3: In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING:
    Could not compute standard errors! The information matrix could
    not be inverted. This may be a symptom that the model is not
    identified.
4: In lav_object_post_check(object) :
  lavaan WARNING: some estimated lv variances are negative
```

We are warned that a numerical solution has not been found, and that the information matrix (that's the Fisher Information, the Hessian of the minus log likelihood) could not be inverted. This means that the minus log likelihood is not strictly concave up in every direction at the point where the search stopped, so the search has not located a local minimum. `lavaan` speculates that "this may be a symptom that the model is not identified," but the guess is wrong. This is standard double measurement regression, and we have proved that all the parameters are identifiable. At the end of the red warnings, we are also informed that some estimated latent variable variances are negative. This means that the numerical search for the MLE has left the parameter space.

The output of `summary(fit1)` is quite voluminous. There are 45 parameters, and everything we do will generate a lot of output. It starts like this.

```
lavaan 0.6-7 ended normally after 4241 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         45
```

```
  Number of observations                           500

Model Test User Model:

  Test statistic                                89.369
  Degrees of freedom                               10
  P-value (Chi-square)                          0.000
```

That's a lot of iterations, and the criteria for "normal" convergence appear to be quite forgiving. The output goes on. The last section gives variance estimates; as the warning message said, one of them is negative.

```
Variances:
                   Estimate   Std.Err  z-value  P(>|z|)
    Lage    (ph11)   146.720        NA
    Lbmi    (ph22)    12.318        NA
    Lfat    (ph33)    42.615        NA
   .Lcholst (ps11)   169.820        NA
   .Ldiastl (ps22) -2785.532        NA
   .age1    (w111)    18.767        NA
   .bmi1    (w122)     9.177        NA
   .fat1    (w133)    18.669        NA
   .cholst1 (w144)   200.123        NA
   .diastl1 (w155)   204.316        NA
   .age2    (w211)     8.326        NA
   .bmi2    (w222)     2.460        NA
   .fat2    (w233)     9.975        NA
   .cholst2 (w244)   344.031        NA
   .diastl2 (w255)    59.441        NA
```

Besides being negative, the value of $\widehat{\psi}_{22}$ is very large in absolute value compared to the other variances. This, combined with the large number of iterations, suggests that the numerical search wandered off and gotten lost somewhere far from the actual MLE.

The minus log likelihood functions for structural equation models are characterized by hills and valleys. There can be lots of local maxima and minima. While there will be a deep hole somehere for a sufficiently large sample is the model is correct, the only guarantee of finding it is to start the search close to the hole, where the surface is already sloping down in the right direction. Otherwise, what happens will depend on the detailed topography of the minus log likelihood, and finding the correct MLE is far from guaranteed.

Here, it seems that that `lavaan`'s default starting values, which often work quite well, were fairly far from the global minimum. The search proceeded downhill, but only slightly downhill after a while[42], off into the distance in an almost featureless plain. It was never

[42]The `verbose = TRUE` option on the `lavaan` statement generated thousands of lines of output, not shown here. The decrease in the minus log likelihood was more and more gradual near the end.

going to arrive anywhere meaningful.

I tried setting boundaries to prevent variances from becoming negative, hoping the search would bounce off the barrier into a better region of the parameter space. I added the following to the model string `bmimodel1`,

```
# Bounds (Variances are positive)
# ------
phi11 > 0; phi22 > 0 ; phi33 > 0
psi11 > 0; psi22 > 0
w111 > 0; w122 > 0; w133 > 0; w144 > 0; w155 > 0;
w211 > 0; w222 > 0; w233 > 0; w244 > 0; w255 > 0
```

and then re-ran `lavaan`. The search converged "normally" after 1,196 iterations. This time $\widehat{\psi}_{22}$ was (just barely) positive, but we get this warning.

```
lavaan WARNING: covariance matrix of latent variables
              is not positive definite;
              use lavInspect(fit, "cov.lv") to investigate.
```

The `lavInspect` function is very useful and powerful. See `help(lavInspect)` for details. Following their suggestion,

```
> lavInspect(fit1, "cov.lv")
          Lage     Lbmi     Lfat     Lchlst   Ldistl
Lage      146.667
Lbmi        3.021   11.672
Lfat       24.479   21.887   43.473
Lcholest   21.588   65.420  121.015 2893.067
Ldiastol   37.581   26.730   54.471  109.211  140.689
```

That's the estimated covariance matrix of the latent variables – very nice! It does not really tell me much, except that the estimated variance of latent cholesterol level is suspiciously large compared to the other numbers in the matrix. To see that the matrix not positive definite, one can look at the eigenvalues.

```
lvcov = lavInspect(fit1, "cov.lv"); eigen(lvcov)$values
[1] 2904.4720798211  198.2045328588  111.6623591169   21.2286105008   -0.0003796765
```

Sure enough, there's a negative eigenvlue, so the matrix is not positive definite.

The only cure for this disease is better starting values. Commercial software for structural equation modeling uses a deep and sophisticated bag of tricks to pick starting values, and SAS `proc calis` has no trouble with this model and these data model. However, as of this writing, `lavaan`'s automatic starting values work okay only most of the time[43].

[43] I'm not complaining. I am deeply grateful for `lavaan`, and if I want better starting values I should develop the software myself. To me, this is not the most interesting project in the world, so it is on the back burner.

Here is a way to obtain good starting values for any structural equation model, provided the parameters are identifiable. Recall how the proof of identifiability goes. For any model, the covariance matrix is a function of the model parameters: $\boldsymbol{\Sigma} = g(\boldsymbol{\theta})$. This equality represents the *covariance structure equations*. The parameters that appear in $\boldsymbol{\Sigma}$ are identifiable if the covariance structure equations can be solved to yield $\boldsymbol{\theta} = g^{-1}(\boldsymbol{\Sigma})$. Provided the solution is available explicitly[44], a method of moments estimator is $\widehat{\boldsymbol{\theta}}_M = g^{-1}(\widehat{\boldsymbol{\Sigma}})$, where $\widehat{\boldsymbol{\Sigma}}$ denotes the sample variance-covariance matrix. Typically, the function $g^{-1}$ is continuous in most of the parameter space. In this case, the method of moments estimator is guaranteed to be consistent by the Law of Large Numbers and continuous mapping. Since the MLE is also consistent, it will be close to $\widehat{\boldsymbol{\theta}}_M$ for large samples, and $\widehat{\boldsymbol{\theta}}_M$ should provide an excellent set of starting values.

For double measurement regression, the solution (49) represents $\boldsymbol{\theta} = g^{-1}(\boldsymbol{\Sigma})$. One may start with Expression (51) for $\widehat{\boldsymbol{\Phi}}_M$ and Expression (52) for $\widehat{\boldsymbol{\beta}}_M$ (see page 89), and then use (49) for the rest of the parameters. This is done in the R work below.

```
> # Obtain the MOM estimates to use as starting values.
> head(bmidata)
  age1 bmi1 fat1 cholest1 diastol1 age2 bmi2 fat2 cholest2 diastol2
1   63 24.5 16.5    195.4       38   60 23.9 20.1    203.5       66
2   42 13.0  1.9    184.3       86   44 14.8  2.6    197.3       78
3   32 22.5 14.6    354.1      104   33 21.7 20.4    374.3       73
4   59 25.5 19.0    214.6       93   58 28.5 20.0    203.7      106
5   45 26.5 17.8    324.8       97   43 25.0 12.3    329.7       92
6   31 19.4 17.1    280.7       92   42 19.9 19.9    276.7       87

> W1 = as.matrix(bmidata[,1:3])  # age1 bmi1 fat1
> V1 = as.matrix(bmidata[,4:5])  # cholest1 diastol1
> W2 = as.matrix(bmidata[,6:8])  # age2 bmi2 fat2
> V2 = as.matrix(bmidata[,9:10]) # cholest2 diastol2
> var(W1,W2) # Matrix of sample covariances
           age2      bmi2     fat2
age1 148.220782  3.621581 25.29808
bmi1   5.035726 13.194016 21.42201
fat1  23.542289 20.613490 45.13296

> # Using S as short for Sigmahat, and not worrying about n vs. n-1,
> S11 = var(W1); S12 = var(W1,V1); S13 = var(W1,W2); S14 = var(W1,V2)
>                S22 = var(V1);    S23 = var(V1,W2); S24 = var(V1,V2)
>                                  S33 = var(W2);    S34 = var(W2,V2)
>                                                    S44 = var(V2)
> # The matrices below should all have "hat" in the name, because they are estimates
```

[44] For some models, an explicit solution is hard to obtain, even if you can prove it exists. That's the main obstacle to automating this process.

```
> Phi = (S13+t(S13))/2
> rownames(Phi) = colnames(Phi) = c('Lage','Lbmi','Lfat'); Phi
           Lage      Lbmi     Lfat
Lage 148.220782  4.328654 24.42019
Lbmi   4.328654 13.194016 21.01775
Lfat  24.420185 21.017749 45.13296

> Beta = 0.5*(t(S14)+S23) %*% solve(Phi)
> rownames(Beta) = c('Lcholest','Ldiastol')
> colnames(Beta) = c('Lage','Lbmi','Lfat'); Beta
               Lage       Lbmi     Lfat
Lcholest -0.3851327 -0.1885072 2.968322
Ldiastol  0.0224190 -0.3556138 1.407425

> Psi = S24 - Beta %*% Phi %*% t(Beta)
> rownames(Psi) = colnames(Psi) = c('Lcholest','Ldiastol') # epsilon1, epsilon2
> Psi
           Lcholest  Ldiastol
Lcholest 2548.17303 -44.56069
Ldiastol  -28.70087  57.64153

> # Oops, it should be symmetric.
> Psi = ( Psi+t(Psi) )/2; Psi
           Lcholest  Ldiastol
Lcholest 2548.17303 -36.63078
Ldiastol  -36.63078  57.64153

> Omega11 = S11 - Phi; Omega11
          age1     bmi1      fat1
age1 19.640040 4.610807  1.634183
bmi1  4.610807 8.699533  8.754484
fat1  1.634183 8.754484 15.033932

> Omega12 = S12 - ( S14+t(S23) )/2; Omega12
      cholest1  diastol1
age1  4.499017 12.164192
bmi1 -1.517733 10.671443
fat1  3.888565 -2.196681

> Omega22 = S22-S24 # A little rough but consistent
> Omega22 = (Omega22 + t(Omega22)  )/2
> Omega22
          cholest1  diastol1
cholest1 213.76117  11.24971
```

```
diastol1  11.24971 196.44520

> Omega33 = S33 - Phi; Omega33
          age2      bmi2      fat2
age2  5.862661 -1.219843 -2.155736
bmi2 -1.219843  1.146991 -1.714769
fat2 -2.155736 -1.714769 10.033984

> Omega34 = S34 - ( S14+t(S23) )/2; Omega34
      cholest2   diastol2
age2 -2.978041  0.7795992
bmi2 -1.206256  2.1081739
fat2 -6.422983 -4.9125882

> Omega44 = S44 - S24 ; Omega44 = ( Omega44 + t(Omega44) )/2
> Omega44
          cholest2  diastol2
cholest2 333.45335 -21.65923
diastol2 -21.65923  47.23065

> round(Beta,3)
           Lage   Lbmi  Lfat
Lcholest -0.385 -0.189 2.968
Ldiastol  0.022 -0.356 1.407
```

Please look at the last set of numbers. It is worth noting how far these method-of-moments estimates are from the stopping place of the first numerical search. Here is a piece of the output from the first `summary(fit1)`, not shown before.

```
                 Estimate   Std.Err  z-value  P(>|z|)
Lcholest ~
  Lage    (bt11)   -26.391       NA
  Lbmi    (bt12)  -354.932       NA
  Lfat    (bt13)   203.432       NA
Ldiastol ~
  Lage    (bt21)   -28.583       NA
  Lbmi    (bt22)  -390.464       NA
  Lfat    (bt23)   221.685       NA
```

While the method-of-moments estimates are promising as starting values, there is no doubt that entering them all manually is a major pain. I was motivated and I was confident it would work, so I did it. The model string is given below. As in Example 0.10.3, variables appear twice, once to specify the parameter name and a second time to specify the starting value.

```
> bmimodel2 =
```

```
+     #
+     # Latent variable model
+     # ---------------------
+         'Lcholest ~ beta11*Lage        + beta12*Lbmi        + beta13*Lfat +
+                     start(-0.385)*Lage + start(-0.189)*Lbmi + start(2.968)*Lfat
+          Ldiastol ~ beta21*Lage        + beta22*Lbmi        + beta23*Lfat +
+                     start(0.022)*Lage  + start(-0.356)*Lbmi + start(1.407)*Lfat
+     #
+     # Measurement model
+     # -----------------
+         Lage =~ 1*age1 + 1*age2
+         Lbmi =~ 1*bmi1 + 1*bmi2
+         Lfat =~ 1*fat1 +1*fat2
+         Lcholest =~ 1*cholest1 + 1*cholest2
+         Ldiastol =~ 1*diastol1 + 1*diastol2
+     #
+     # Variances and covariances
+     # -------------------------
+     # Of latent explanatory variables
+         Lage ~~ phi11*Lage + start(148.220782)*Lage
+         Lage ~~ phi12*Lbmi + start(4.328654)*Lbmi
+         Lage ~~ phi13*Lfat + start(24.42019)*Lfat
+         Lbmi ~~ phi22*Lbmi + start(13.194016)*Lbmi
+         Lbmi ~~ phi23*Lfat + start(21.01775)*Lfat
+         Lfat ~~ phi33*Lfat + start(45.13296)*Lfat
+     # Of error terms in latent the regression (epsilon_ij)
+         Lcholest ~~ psi11*Lcholest + start(2548.17303)*Lcholest
+         Lcholest ~~ psi12*Ldiastol + start(-36.63078)*Ldiastol
+         Ldiastol ~~ psi22*Ldiastol + start(57.64153)*Ldiastol
+     # Of measurement errors (e_ijk) for measurement set 1
+         age1 ~~ w111*age1 + start(19.640040)*age1
+         age1 ~~ w112*bmi1 + start(4.610807)*bmi1
+         age1 ~~ w113*fat1 + start(1.634183)*fat1
+         age1 ~~ w114*cholest1 + start(4.499017)*cholest1
+         age1 ~~ w115*diastol1 + start(12.164192)*diastol1
+         bmi1 ~~ w122*bmi1 + start(8.699533)*bmi1
+         bmi1 ~~ w123*fat1 + start(8.754484)*fat1
+         bmi1 ~~ w124*cholest1 + start(-1.517733)*cholest1
+         bmi1 ~~ w125*diastol1 + start(10.671443)*diastol1
+         fat1 ~~ w133*fat1 + start(15.033932)*fat1
+         fat1 ~~ w134*cholest1 + start(3.888565)*cholest1
+         fat1 ~~ w135*diastol1 + start(-2.196681)*diastol1
+         cholest1 ~~ w144*cholest1 + start(213.76117)*cholest1
+         cholest1 ~~ w145*diastol1 + start(11.24971)*diastol1
```

```
+         diastol1 ~~ w155*diastol1 + start(196.44520)*diastol1
+     # Of measurement errors (e_ijk) for measurement set 2
+         age2 ~~ w211*age2 + start(5.862661)*age2
+         age2 ~~ w212*bmi2 + start(-1.219843)*bmi2
+         age2 ~~ w213*fat2 + start(-2.155736)*fat2
+         age2 ~~ w214*cholest2 + start(-2.978041)*cholest2
+         age2 ~~ w215*diastol2 + start(0.7795992)*diastol2
+         bmi2 ~~ w222*bmi2 + start(1.146991)*bmi2
+         bmi2 ~~ w223*fat2 + start(-1.714769)*fat2
+         bmi2 ~~ w224*cholest2 + start(-1.206256)*cholest2
+         bmi2 ~~ w225*diastol2 + start(2.1081739)*diastol2
+         fat2 ~~ w233*fat2 + start(10.033984)*fat2
+         fat2 ~~ w234*cholest2 + start(-6.422983)*cholest2
+         fat2 ~~ w235*diastol2 + start(-4.9125882)*diastol2
+         cholest2 ~~ w244*cholest2 + start(333.45335)*cholest2
+         cholest2 ~~ w245*diastol2 + start(-21.65923)*diastol2
+         diastol2 ~~ w255*diastol2 + start(47.23065)*diastol2
+     # Bounds (Variances are positive)
+     # ------
+         phi11 > 0; phi22 > 0 ; phi33 > 0
+         psi11 > 0; psi22 > 0
+         w111 > 0; w122 > 0; w133 > 0; w144 > 0; w155 > 0;
+         w211 > 0; w222 > 0; w233 > 0; w244 > 0; w255 > 0
+     ' ################## End of bmimodel2 ##################
> fit2 = lavaan(bmimodel2, data=bmidata)
> summary(fit2)
lavaan 0.6-7 ended normally after 327 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         45
  Number of inequality constraints                  15

  Number of observations                           500

Model Test User Model:

  Test statistic                                 4.654
  Degrees of freedom                                10
  P-value (Chi-square)                           0.913

Parameter Estimates:

  Standard errors                             Standard
```

```
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  Lage =~
    age1              1.000
    age2              1.000
  Lbmi =~
    bmi1              1.000
    bmi2              1.000
  Lfat =~
    fat1              1.000
    fat2              1.000
  Lcholest =~
    cholest1          1.000
    cholest2          1.000
  Ldiastol =~
    diastol1          1.000
    diastol2          1.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  Lcholest ~
    Lage    (bt11)   -0.320    0.228   -1.404    0.160
    Lbmi    (bt12)    0.393    1.708    0.230    0.818
    Lfat    (bt13)    2.774    0.980    2.829    0.005
  Ldiastol ~
    Lage    (bt21)    0.020    0.050    0.407    0.684
    Lbmi    (bt22)   -0.480    0.419   -1.145    0.252
    Lfat    (bt23)    1.480    0.235    6.312    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  Lage ~~
    Lbmi    (ph12)    4.161    2.141    1.944    0.052
    Lfat    (ph13)   23.321    3.986    5.851    0.000
  Lbmi ~~
    Lfat    (ph23)   20.976    1.584   13.244    0.000
 .Lcholest ~~
   .Ldiastl (ps12)  -45.870   24.969   -1.837    0.066
 .age1 ~~
   .bmi1    (w112)    3.998    0.945    4.231    0.000
   .fat1    (w113)    2.389    1.505    1.587    0.112
```

```
   .cholst1 (w114)    2.705    9.091    0.297    0.766
   .diastl1 (w115)   10.562    3.824    2.762    0.006
 .bmi1 ~~
   .fat1    (w123)    8.968    0.956    9.382    0.000
   .cholst1 (w124)   -0.888    4.178   -0.212    0.832
   .diastl1 (w125)   10.060    2.274    4.424    0.000
 .fat1 ~~
   .cholst1 (w134)    7.916    6.741    1.174    0.240
   .diastl1 (w135)   -2.928    3.409   -0.859    0.390
 .cholest1 ~~
   .diastl1 (w145)   -0.107   16.907   -0.006    0.995
 .age2 ~~
   .bmi2    (w212)   -0.661    0.735   -0.899    0.369
   .fat2    (w213)   -2.703    1.369   -1.974    0.048
   .cholst2 (w214)   -1.964    8.962   -0.219    0.827
   .diastl2 (w215)    2.274    2.710    0.839    0.401
 .bmi2 ~~
   .fat2    (w223)   -1.849    0.705   -2.624    0.009
   .cholst2 (w224)   -2.650    3.476   -0.762    0.446
   .diastl2 (w225)    2.652    1.487    1.784    0.074
 .fat2 ~~
   .cholst2 (w234)  -11.370    6.546   -1.737    0.082
   .diastl2 (w235)   -4.839    2.536   -1.908    0.056
 .cholest2 ~~
   .diastl2 (w245)   -8.964   12.605   -0.711    0.477

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    Lage    (ph11)  147.330    9.699   15.190    0.000
    Lbmi    (ph22)   13.341    0.986   13.528    0.000
    Lfat    (ph33)   44.485    3.101   14.345    0.000
   .Lcholst (ps11) 2534.507  171.258   14.799    0.000
   .Ldiastl (ps22)   56.169    9.221    6.092    0.000
   .age1    (w111)   18.584    2.914    6.378    0.000
   .bmi1    (w122)    8.665    0.708   12.239    0.000
   .fat1    (w133)   16.124    1.659    9.717    0.000
   .cholst1 (w144)  200.103   57.422    3.485    0.000
   .diastl1 (w155)  195.040   14.323   13.617    0.000
   .age2    (w211)    6.861    2.701    2.540    0.011
   .bmi2    (w222)    1.089    0.491    2.220    0.026
   .fat2    (w233)    9.332    1.539    6.065    0.000
   .cholst2 (w244)  344.454   60.290    5.713    0.000
   .diastl2 (w255)   48.350    8.246    5.864    0.000
```

```
Constraints:
                                        |Slack|
    phi11 - 0                          147.330
    phi22 - 0                           13.341
    phi33 - 0                           44.485
    psi11 - 0                         2534.507
    psi22 - 0                           56.169
    w111 - 0                            18.584
    w122 - 0                             8.665
    w133 - 0                            16.124
    w144 - 0                           200.103
    w155 - 0                           195.040
    w211 - 0                             6.861
    w222 - 0                             1.089
    w233 - 0                             9.332
    w244 - 0                           344.454
    w255 - 0                            48.350
```

With these starting values, the maximum likelihood search converged after 327 iterations. The likelihood ratio chi-squared test of model fit indicated no problems: $G^2 = 4.654, df = 10, p = 0.913$. Primary interest is in the relationship of latent (true) BMI to latent cholesterol level and latent blood pressure, controlling for latent age and latent percent body fat. When measurement error was taken into account using double measurement, neither relationship was statistically significant at the 0.05 level. For cholesterol, $Z = 0.230$ and $p = 0.818$. For diastolic blood pressure, $Z = -1.145$ and $p = 0.252$. This is in contrast to the conclusion from naive ordinary least squarees regression, which was that controlling for age and percent body fat, higher BMI was associated with higher average diastolic blood pressure. Brunner and Austin (1992; also see Section 0.7) have shown how this kind of "even controlling for" conclusion is the kind of error that tends to creep in with ordinary regression, when the explanatory variables are measured with error. Double measurement regression has more credibility.

Plenty more tests based on this model are possible and worthwhile, but BMI controlling for age and percent body fat is the main issue. Just as a demonstration, let's look at one more test, a likelihood ratio test of BMI controlling for age and percent body fat, for cholesterol and diastolic blood pressure simultaneously. The null hypothesis is $H_0 : \beta_{21} = \beta_{22} = 0$. We begin by fitting a restricted model[45]. Note that each constraint has to go on a separate line.

```
> nobmi = lavaan(bmimodel2, data=bmidata,
+             constraints = 'beta12 == 0
+                            beta22 == 0')
```

[45]It is a relief that the non-zero starting values for $\beta_{21}$ and $\beta_{22}$ in `bmimodel2` do not conflict with the constraint that sets them equal to zero.

```
>
> anova(nobmi,fit2)
Chi Square Difference Test

      Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit2  10 35758 35947 4.6537
nobmi 12 35755 35936 6.1457      1.492       2     0.4743
```

Again, the conclusion is that allowing for age and percent body fat, there is no evidence of a connection between BMI and either health indicator.

## 0.11 Extra Response Variables

Sometimes, double measurement is not a practical alternative. Perhaps the data are already collected, and the study was designed without planning for a latent variable analysis. The guilty parties might be academic or private sector researchers who do not know what a parameter is, much less parameter identifiability. Or, the data might have been collected for some purpose other than research. For example, a paper mill might report the amount and concentrations of poisonous chemicals they dump into a nearby river. They take the measurements because they have agreed to do so, or because they are required to do it by law — but they certainly are not going to do it twice. Much economic data and public health data is of this kind.

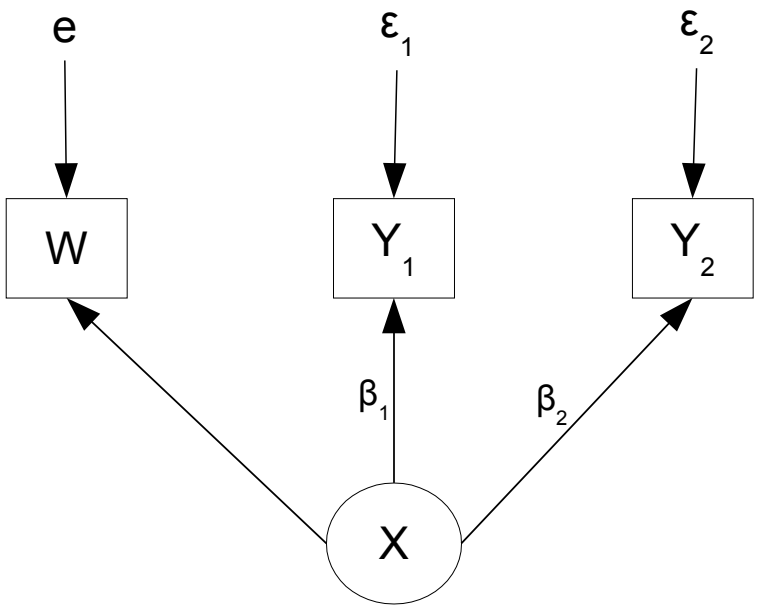
In such situations, all one can do is to use what information happens to be available. While most research studies will not contain multiple measurements of the explanatory variables, they often will have quite a few possible response variables. These variables might already be part of the data set, or possibly the researchers could go back and collect them without an unbearable amount of effort. It helps if these extra response variables are from a different domain than the response variable of interest, so one can make a case that the extra variables and the response variables of interest are not affected by common omitted variables. In the path diagrams, this is represented by the absence of curved, double-headed arrows connecting error terms. It is a critical part of the recipe.

### One explanatory variable

In a simple measurement error regression model like the one in Example 0.8.1, suppose that we have access to data for a second response variable that depends on the latent explanatory variable $X_i$. Our main interest is still in the response variable $Y_i$. The other response variable may or may not be interesting in its own right; it is included as a way of getting around the identifiability problem.

**Example 0.11.1** *One Extra Response Variable*

Figure 17: $Y_2$ is an extra response variable




Here is the expanded version of the model. The original response variable $Y_i$ is now called $Y_{i,1}$. Independently for $i = 1, \ldots, n$.

$$\begin{aligned} W_i &= \nu + X_i + e_i \\ Y_{i,1} &= \alpha_1 + \beta_1 X_i + \epsilon_{i,1} \\ Y_{i,2} &= \alpha_2 + \beta_2 X_i + \epsilon_{i,2} \end{aligned} \tag{53}$$

where $e_i$, $\epsilon_{i,1}$ and $\epsilon_{i,2}$ are all independent, $Var(X_i) = \phi$, $Var(\epsilon_{i,1}) = \psi_1$, $Var(\epsilon_{i,2}) = \psi_2$, $Var(e_i) = \omega$, $E(X_i) = \mu_x$, and the expected values of all error terms are zero. Figure 17 shows a path diagram of this model.

It is usually helpful to check the parameter count rule (Rule 1) before doing detailed calculations. For this model, there are ten parameters: $\boldsymbol{\theta} = (\nu, \alpha_1, \alpha_2, \beta_1, \beta_2, \mu_x, \phi, \omega, \psi_1, \psi_2)$. Writing the vector of observable data for case $i$ as $\mathbf{D}_i = (W_i, Y_{i,1}, Y_{i,2})^\top$, we see that $\boldsymbol{\mu} = E(\mathbf{D}_i)$ has three elements and $\boldsymbol{\Sigma} = cov(\mathbf{D}_i)$ has $3(3+1)/2 = 6$ unique elements. Thus identifiability of the entire parameter vector is ruled out in most of the parameter space. However, it turns out that useful *functions* of the parameter vector are identifiable, and this includes $\beta_1$, the parameter of primary interest.

Based on our experience with the double measurement model, we are pessimistic about identifying expected values and intercepts. So consider first the covariance matrix. Elements of $\boldsymbol{\Sigma} = cov(\mathbf{D}_i)$ may be obtained by elementary one-variable calculations, like

$Var(W_i) = Var(\nu + X_i + e_i) = Var(X_i) + Var(e_i) = \phi + \omega$, and

$$\begin{aligned} Cov(W_i, Y_{i,i}) &= Cov(X_i + e_i,\ \beta_1 X_i + \epsilon_{i,1}) \\ &= \beta_1 Cov(X_i, X_i) + Cov(X_i, \epsilon_{i,1}) + \beta_1 Cov(e_i, X_i) + Cov(e_i, \epsilon_{i,1}) \\ &= \beta_1 Var(X) + 0 + 0 + 0 \\ &= \beta_1 \phi \end{aligned}$$

In this way we obtain

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{pmatrix} = \begin{pmatrix} \phi + \omega & \beta_1\phi & \beta_2\phi \\ & \beta_1^2\phi + \psi_1 & \beta_1\beta_2\phi \\ & & \beta_2^2\phi + \psi_2 \end{pmatrix},$$

which is a nice compact way to look at the six covariance structure equations in six unknown parameters. The fact that there are the same number of equations and unknowns does not guarantee the existence of a unique solution; it merely tells us that a unique solution is possible. It turns out that for this model, identifiability depends on where in the parameter space the true parameter is located. In the following, please bear in mind that the only parameter we really care about is $\beta_1$, which represents the connection between $X$ and $Y_1$. All the other parameters are just nuisance parameters.

Since $\sigma_{12} = 0$ if and only if $\beta_1 = 0$, the parameter $\beta_1$ is identifiable whenever it equals zero. But then both $\sigma_{12} = 0$ and $\sigma_{23} = 0$, reducing the six equations in six unknowns to four equations in five unknowns, meaning the other parameters in the covariance matrix can't all be recovered.

But what if $\beta_1$ does not equal zero? At those points in the parameter space where $\beta_2$ is non-zero, $\beta_1 = \frac{\sigma_{23}}{\sigma_{13}}$. This means that adding $Y_2$ to the model bought us what we need, which is the possibility of correct estimation and inference about $\beta_1$. Note that stipulating $\beta_2 \neq 0$ is not a lot to ask, because it just means that the extra variable is related to the response variable. Otherwise, why include it[46]?

If both $\beta_1 \neq 0$ and $\beta_2 \neq 0$, all six parameters in the covariance matrix can be recovered

[46]Moreover, one can rule out $\beta_2 = 0$ by a routine test of the correlation between $W$ and $Y_2$. This kind of test is very helpful (assuming the data are in hand), because for successful inference it's not necessary for the entire parameter to be identifiable everywhere in the parameter space. It's only necessary for the interesting part of the parameter vector to be identifiable in the region of the parameter space where the true parameter is located.

by simple substitutions as follows:

$$\begin{aligned} \beta_1 &= \frac{\sigma_{23}}{\sigma_{13}} \\ \beta_2 &= \frac{\sigma_{23}}{\sigma_{12}} \\ \phi &= \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \\ \omega &= \sigma_{11} - \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \\ \psi_1 &= \sigma_{22} - \frac{\sigma_{12}\sigma_{23}}{\sigma_{13}} \\ \psi_2 &= \sigma_{33} - \frac{\sigma_{13}\sigma_{23}}{\sigma_{12}} \end{aligned} \tag{54}$$

This is a success, but actually the job is not done yet. Four additional parameters appear only in the expected value of the data vector; they are the expected value and intercepts: $\nu, \mu_x, \alpha_1$, and $\alpha_2$. We have

$$\begin{aligned} \mu_1 &= \nu + \mu_x \\ \mu_2 &= \alpha_1 + \beta_1\mu_x \\ \mu_3 &= \alpha_2 + \beta_2\mu_x \end{aligned} \tag{55}$$

Even treating $\beta_1$ and $\beta_2$ as known because they can be identified from the covariance matrix, this system of three linear equations in four unknowns does not have a unique solution.

As in the double measurement case, this lack of identifiability is really not too serious, because our primary interest is in $\beta_1$. So we re-parameterize, absorbing the expected value and intercepts into $\boldsymbol{\mu}$ exactly as defined in the mean structure equations (55). The new parameters $\mu_1, \mu_2$ and $\mu_3$ may not be very interesting in their own right, but they can be safely estimated by the vector of sample means and then disregarded.

To clarify, the original parameter was

$$\boldsymbol{\theta} = (\nu, \mu_x, \alpha_1, \alpha_2, \beta_1, \beta_2, \phi, \omega, \psi_1, \psi_2).$$

Now it's

$$\boldsymbol{\theta} = (\mu_1, \mu_2, \mu_3, \beta_1, \beta_2, \phi, \omega, \psi_1, \psi_2).$$

The dimension of the parameter space is now one less, and we haven't lost anything that is either accessible or important. This is all the more true because the model pretends that the response variables are measured without error. Actually, the equations for $Y_{i,1}$ and $Y_{i,2}$ should be viewed as re-parameterizations like the one in Expression (32) on page 46, and the intercepts $\alpha_1$ and $\alpha_2$ are already the original intercepts plus un-knowable measurement bias terms.

To an important degree, this is the story of structural equation models. The models usually used in practice are not what the scientist or statistician originally had in

mind. Instead, they are the result of judicious re-parameterizations, in which the original parameter vector is collapsed into a vector of *functions* of the parameters that are identifiable, and at the same time allow valid inference about the original parameters that are of primary interest.

Example 0.11.1 is interesting for another reason. The purpose of all this is to test $H_0 : \beta_1 = 0$, but even if an assumption of normality is justified, the usual normal theory tests will break down if the null hypothesis is true. Though $\beta_1$ is identifiable when the null hypothesis is true, the entire parameter vector is not. There will be trouble fitting the restricted model needed for a likelihood ratio test, because infinitely many sets $(\beta_2, \phi, \psi_2, \omega)$ yield the same covariance matrix when $\beta_1 = 0$.

The Wald test will suffer too, even though it requires fitting only the unrestricted model. For one thing, local identifiability at the true parameter value is assumed in the proof of asymptotic normality of the MLE, and I don't see a way of getting around it; see for example Davison [20], p. 119 and Wald [66]. Even setting theoretical considerations aside, the experience of fitting the unrestricted model and trying to test $H_0 : \beta_1 = 0$ is likely to be unpleasant. This is illustrated in a small-scale simulation study.

### A little simulation study

Using R, $n$ sets of independent $(W_i, Y_{i,1}, Y_{i,2})$ triples were generated from Model (53), with $\beta_1 = 0$, $\beta_2 = 1$, and $\phi = \omega = \psi_1 = \psi_2 = 1$. Note that this makes $H_0 : \beta_1 = 0$ true, and the entire parameter vector is not identifiable. The expected values and intercepts were all zero, and all the variables were normally distributed. This was carried out 1,000 times for $n = 50, 100, 500$ and 1000, and `lavaan` was used to fit the model to each simulated data set. Here is the code.

```
################################################
# Run n = 50, 100, 500, 1000 separately
################################################
rm(list=ls()); options(scipen=999)
# install.packages("lavaan", dependencies = TRUE) # Only need to do this once
library(lavaan)
n = 50  # Set the sample size here
# Parameters
beta1 = 0; beta2 = 1; phi = 1; omega = 1; psi1 = 1; psi2 = 1
# Initialize
M = 1000
converged = logical(M) # Did the numerical search converge?
posvar = logical(M)    # Are all the estimated variances positive?
# Only have to define the model once.
mod1 = 'Y1 ~ beta1*X # Latent variable model
        Y2 ~ beta2*X
        X =~ 1.0*W  # Measurement model
        # Variances (covariances would go here too)
```

```
        X~~phi*X      # Var(X) = phi
        W ~~ omega*W  # Var(e) = omega
        Y1 ~~ psi1*Y1 # Var(epsilon1) = psi1
        Y2 ~~ psi2*Y2 # Var(epsilon2) = psi2
       '
# Simulate: Random number seed is sample size
set.seed(n)
for(sim in 1:M)
{
    x = rnorm(n,0,sqrt(phi)); e = rnorm(n,0,sqrt(omega))
    epsilon1 = rnorm(n,0,sqrt(psi1)); epsilon2 = rnorm(n,0,sqrt(psi2))
    W = x + e
    Y1 = beta1*x + epsilon1
    Y2 = beta2*x + epsilon2
    simdat = data.frame( cbind(W,Y1,Y2) ) # Data must be in a data frame
    fit1 = lavaan(mod1, data = simdat) # Fit the model
    # Gather data on this simulation
    converged[sim] = lavInspect(fit1,"converged") # Checking convergence
    posvar[sim] = lavInspect(fit1,"post.check") # All estimated variances positive?
} # Next sim
addmargins(table(converged,posvar)) # Look at results
```

Table 3 shows that the numerical maximum likelihood search converged to a point in the parameter space only about one third of the time. For about one third of the simulations,

Table 3: Simulation from Model (53)

| | Sample Size | | | |
|---|---|---|---|---|
| | $n = 50$ | $n = 100$ | $n = 500$ | $n = 1,000$ |
| Did not converge | 366 | 310 | 327 | 355 |
| Converged, but at least one negative variance estimate | 322 | 336 | 315 | 302 |
| Converged, variance estimates all positive | 312 | 354 | 358 | 343 |
| Total | 1,000 | 1,000 | 1,000 | 1,000 |

the search failed to converge, and for one third the search converged, but to an answer with negative variance estimates[47]. I expected the problems to be worse with larger

[47]You might be thinking that convergence to a solution with negative variance estimates could be caused by poor starting values. This was not the case. When the numerical search converged, it was almost always to the correct MLE; this happened 2,614 times out of 2,642. How do we know what the

sample sizes, but this did not happen. In any case, fitting the unrestricted model will be confusing and frustrating about two-thirds of the time for this example.

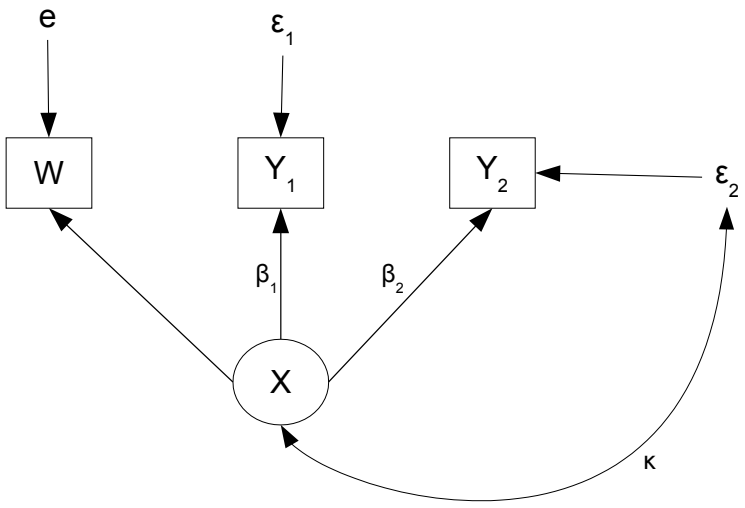
There is a general lesson here, and a way out in this particular case. The general lesson is to re-verify parameter identifiability when the null hypothesis is true, bearing in mind that likelihood methods depend on identifiability of the entire parameter vector. It is better to anticipate trouble and avoid it than to be confused by it once it happens.

As for the way out of the haunted house, note that if $\beta_2 \neq 0$, the null hypothesis $\beta_1 = 0$ is true if and only if $\sigma_{12} = \sigma_{23} = 0$. This null hypothesis can be tested using a generic, unstructured multivariate normal model for the observable data. The likelihood ratio test, like the Wald test, will have two degrees of freedom. If the normal assumption is a source of discomfort, try testing a couple of Spearman rank correlations with a Bonferroni correction. More generally, we will see shortly that having more than one extra response variable can yield identifiability whether or not $H_0 : \beta_1 = 0$ is true. This is a better solution if it's possible, because it makes the analysis more routine.

**Example 0.11.2** *Correlation between explanatory variables and error terms*

Recalling Section 0.4 on omitted variables in regression, it is remarkable that while the explanatory variable $X_i$ must not be correlated with the error term $\epsilon_{i,1}$, the error term $\epsilon_{i,2}$ (corresponding to the extra variable $Y_{i,2}$) is allowed to be correlated with $X_i$, perhaps reflecting the operation of omitted explanatory variables that affect $Y_{i,2}$ and have non-zero covariance with $X_i$. Figure 18 shows a path diagram of this model.

Figure 18: Error term correlated with the explanatory variable




correct MLE was? By invariance. This is a saturated model, in which the number of parameters equals the number of unique variances and covariances. Thus, putting hats on the solution (54) yields the exact maximum likelihood estimates. Note that under the normal model, the joint distribution of the unique elements of $\widehat{\boldsymbol{\Sigma}}$ is continuous, so that with probability one there will be no division by zero.

Suppose $Cov(X_i, \epsilon_{i,2}) = \kappa$, which might be non-zero. This means that seven unknown parameters appear in the six covariance structure equations, and the parameter count rule warns us that it will be impossible to identify them all. Proceeding anyway, the covariance matrix of $\mathbf{D}_i$ becomes

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{pmatrix} = \begin{pmatrix} \phi+\omega & \beta_1\phi & \beta_2\phi+\kappa \\ & \beta_1^2\phi+\psi_1 & \beta_1\beta_2\phi+\beta_1\kappa \\ & & \beta_2^2\phi+\psi_2+2\beta_2\kappa \end{pmatrix}.$$

Assuming as before that $Y_2$ is a useful extra variable so that $\beta_2 \neq 0$,

$$\frac{\sigma_{23}}{\sigma_{13}} = \frac{\beta_1(\beta_2\phi+\kappa)}{\beta_2\phi+\kappa} = \beta_1. \tag{56}$$

In fact, if $\kappa \neq 0$, we don't even need $\beta_2 \neq 0$ to identify $\beta_1$. That is, the extra response variable does not need be influenced by the latent explanatory variable. It need only be influenced by some unknown variable or variables that are *correlated* with the explanatory variable. Far from being a problem in this case, the omitted variables made it easier to get at $\beta_1$. In Figure 18, $Y_2$ is an instrumental variable, a point to which we will return in Section 0.12.

As in Example 0.11.1, testing $H_0 : \beta_1 = 0$ is non-standard because while $\beta_1$ is identifiable, the entre parameter vector is not. We can deal with this kind of complication if we need to, but everything is much easier with more than one extra variable.

**Example 0.11.3** *More Than One Extra Response Variable*

Suppose that the data set contains another *two* variables that depend on the latent explanatory variable $X_i$. Our main interest is still in the response variable $Y_{i,1}$; the other two are just to help with identifiability. Now the model is, independently for $i = 1, \ldots, n$,

$$\begin{aligned} W_i &= \nu + X_i + e_i \\ Y_{i,1} &= \alpha_1 + \beta_1 X_i + \epsilon_{i,1} \\ Y_{i,2} &= \alpha_2 + \beta_2 X_i + \epsilon_{i,2} \\ Y_{i,3} &= \alpha_3 + \beta_3 X_i + \epsilon_{i,3}, \end{aligned} \tag{57}$$

where $e_i$, $e_i$, $\epsilon_{i,1}$, $\epsilon_{i,2}$ and $\epsilon_{i,3}$ are all independent, $Var(X_i) = \phi$, $Var(\epsilon_{i,1}) = \psi_1$, $Var(\epsilon_{i,2}) = \psi_2$, $Var(\epsilon_{i,3}) = \psi_3$, $Var(e_i) = \omega$, $E(X_i) = \mu_x$ and the expected values of all error terms are zero.

Writing the vector of observable data for case $i$ as $\mathbf{D}_i = (W_i, Y_{i,1}, Y_{i,2}, Y_{i,3})^\top$, we have

$$\boldsymbol{\mu} = E\begin{pmatrix} W_i \\ Y_{i,1} \\ Y_{i,2} \\ Y_{i,3} \end{pmatrix} = \begin{pmatrix} \nu+\mu_x \\ \alpha_1+\beta_1\mu_x \\ \alpha_2+\beta_2\mu_x \\ \alpha_3+\beta_3\mu_x \end{pmatrix}$$

and

$$\mathbf{\Sigma} = \begin{pmatrix} \phi + \omega & \beta_1\phi & \beta_2\phi & \beta_3\phi \\ & \beta_1^2\phi + \psi_1 & \beta_1\beta_2\phi & \beta_1\beta_3\phi \\ & & \beta_2^2\phi + \psi_2 & \beta_2\beta_3\phi \\ & & & \beta_3^2\phi + \psi_3 \end{pmatrix}. \tag{58}$$

As before, it is impossible to identify the intercepts and expected values, so we re-parameterize, absorbing them into a vector of expected values which we estimate with the corresponding vector of sample means; we never mention them again.

To establish identifiability of the parameters that appear in the covariance matrix, the task is to solve the following ten equations for the eight unknown parameters $\phi$, $\omega$, $\beta_1$, $\beta_2$, $\beta_3$, $\psi_1$, $\psi_2$, and $\psi_3$:

$$\begin{aligned} \sigma_{11} &= \phi + \omega \\ \sigma_{12} &= \beta_1\phi \\ \sigma_{13} &= \beta_2\phi \\ \sigma_{14} &= \beta_3\phi \\ \sigma_{22} &= \beta_1^2\phi + \psi_1 \\ \sigma_{23} &= \beta_1\beta_2\phi \\ \sigma_{24} &= \beta_1\beta_3\phi \\ \sigma_{33} &= \beta_2^2\phi + \psi_2 \\ \sigma_{34} &= \beta_2\beta_3\phi \\ \sigma_{44} &= \beta_3^2\phi + \psi_3 \end{aligned} \tag{59}$$

Assuming the extra variables are well-chosen so that both $\beta_2$ and $\beta_3$ are both non-zero,

$$\frac{\sigma_{13}\sigma_{14}}{\sigma_{34}} = \frac{\beta_2\beta_3\phi^2}{\beta_2\beta_3\phi} = \phi. \tag{60}$$

Then, simple substitutions allow us to solve for the rest of the parameters, yielding the complete solution

$$
\begin{aligned}
\phi &= \frac{\sigma_{13}\sigma_{14}}{\sigma_{34}} \\
\omega &= \sigma_{11} - \frac{\sigma_{13}\sigma_{14}}{\sigma_{34}} \\
\beta_1 &= \frac{\sigma_{12}\sigma_{34}}{\sigma_{13}\sigma_{14}} \\
\beta_2 &= \frac{\sigma_{34}}{\sigma_{14}} \\
\beta_3 &= \frac{\sigma_{34}}{\sigma_{13}} \\
\psi_1 &= \sigma_{22} - \frac{\sigma_{12}^2\sigma_{34}}{\sigma_{13}\sigma_{14}} \\
\psi_2 &= \sigma_{33} - \frac{\sigma_{13}\sigma_{34}}{\sigma_{14}} \\
\psi_3 &= \sigma_{44} - \frac{\sigma_{14}\sigma_{34}}{\sigma_{13}}
\end{aligned} \tag{61}
$$

This proves identifiability at all points in the parameter space where $\beta_2 \neq 0$ and $\beta_3 \neq 0$. The extra variables $Y_2$ and $Y_3$ have been chosen so as to guarantee this, and in any case the assumption is testable.

The solution (61) is thorough but somewhat tedious, even for this simple example. The student may wonder how much work really needs to be shown. I would suggest showing the calculations leading to the covariance matrix (58), saying "Denote the $i, j$ element of $\mathbf{\Sigma}$ by $\sigma_{ij}$," skipping the system of equations (59) because they are present in (58), and showing the solution for $\phi$ in (60), *including* the stipulation that $\beta_2$ and $\beta_3$ are both non-zero. Then, instead of the explicit solution (61), write something like this:

$$
\begin{aligned}
\omega &= \sigma_{11} - \phi \\
\beta_1 &= \frac{\sigma_{12}}{\phi} \\
\beta_2 &= \frac{\sigma_{13}}{\phi} \\
\beta_3 &= \frac{\sigma_{14}}{\phi} \\
\psi_1 &= \sigma_{22} - \beta_1^2\phi \\
\psi_2 &= \sigma_{33} - \beta_2^2\phi \\
\psi_3 &= \sigma_{44} - \beta_3^2\phi
\end{aligned}
$$

Notice how once we have solved for a model parameter, we use it to solve for other parameters without explicitly substituting in terms of $\sigma_{ij}$. The objective is to prove that a unique solution exists by showing how to get it. A full statement of the solution is not necessary unless you need it for some other purpose, like method of moments estimation.

With two (or more) extra variables, the identifiability argument does not need to be as fussy about the locations in the parameter space where different functions of the parameter vector are identifiable. In particular, there is no loss of identifiability under the natural null hypothesis that $\beta_1 = 0$, and testing that null hypothesis presents no special difficulties.

**Constraints on the covariance matrix** Like the double measurement model, the model of Example 0.11.3 imposes equality constraints on the covariance matrix of the observable data. In the solution given by (61), the critical parameter $\beta_1$ is recovered by $\beta_1 = \frac{\sigma_{12}\sigma_{34}}{\sigma_{13}\sigma_{14}}$, but a look at the covariance structure equations (59) shows that $\beta_1 = \frac{\sigma_{23}}{\sigma_{13}}$ and $\beta_1 = \frac{\sigma_{24}}{\sigma_{14}}$ are also correct. These seemingly different ways of solving for the parameter must be the same. That is,

$$\frac{\sigma_{12}\sigma_{34}}{\sigma_{13}\sigma_{14}} = \frac{\sigma_{23}}{\sigma_{13}} \text{ and } \frac{\sigma_{12}\sigma_{34}}{\sigma_{13}\sigma_{14}} = \frac{\sigma_{24}}{\sigma_{14}}.$$

Simplifying a bit yields

$$\sigma_{12}\sigma_{34} = \sigma_{14}\sigma_{23} = \sigma_{13}\sigma_{24}. \tag{62}$$

Since all three products equal $\beta_1\beta_2\beta_3\phi^2$, the model clearly implies the equality constraints (62) even where the identifiability conditions $\beta_2 \neq 0$ and $\beta_3 \neq 0$ do not hold.

What is happening geometrically is that the covariance structure equations are mapping a parameter space[48] of dimension eight into a moment space of dimension ten. The image of the parameter space is an eight-dimensional surface in the moment space, contained in the set defined by the relations (62). Ten minus eight equals two, the number of over-identifying restrictions.

We will see later that even models with non-identifiable parameters can imply equality constraints. Also, models usually imply *inequality* constraints on the variances and covariances, whether the parameters are identifiable or not. For example, in (61), $\phi = \frac{\sigma_{13}\sigma_{14}}{\sigma_{34}}$. Because $\phi$ is a variance, we have the inequality restriction $\frac{\sigma_{13}\sigma_{14}}{\sigma_{34}} > 0$, something that is not automatically true of covariance matrices in general. Inequalities like this are testable, and provide a valuable way of challenging, or disconfirming a model.

## Multiple explanatory variables

Most real-life models have more than one explanatory variable. No special difficulties arise for the device of introducing extra response variables. In fact, the presence of multiple explanatory variables only provides more ways to identify the parameters and more over-identifying restrictions.

**Example 0.11.4** *Two explanatory variables and two extra response variables*

[48] Actually it's a subset of the parameter space, containing just those parameters that appear in the covariance matrix,

Here is an example with two explanatory variables and a single extra response variable for each one. Independently for $i = 1, \ldots, n$,

$$\begin{aligned} W_{i,1} &= \nu_1 + X_{i,1} + e_{i,1} \\ Y_{i,1} &= \alpha_1 + \beta_1 X_{i,1} + \epsilon_{i,1} \\ Y_{i,2} &= \alpha_2 + \beta_2 X_{i,1} + \epsilon_{i,2} \\ W_{i,2} &= \nu_2 + X_{i,2} + e_{i,2} \\ Y_{i,3} &= \alpha_3 + \beta_3 X_{i,2} + \epsilon_{i,3} \\ Y_{i,4} &= \alpha_4 + \beta_4 X_{i,2} + \epsilon_{i,4} \end{aligned} \tag{63}$$

where $E(X_{i,j}) = \mu_j$, $e_{i,j}$ and $\epsilon_{i,j}$ are independent of one another and of $X_{i,j}$, $Var(e_{i,j}) = \omega_j$, $Var(\epsilon_{i,j}) = \psi_j$, and

$$cov\begin{pmatrix} X_{i,1} \\ X_{i,1} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}.$$

As usual, intercepts and expected values can't be recovered individually. Eight parameters are intercepts and expected values of latent variables that appear in the expressions for only six expected values of the observable variables. So we re-parameterize, absorbing them into $\mu_1, \ldots, \mu_6$. Then we estimate $\boldsymbol{\mu}$ with the vector of 6 sample means and set it aside, forever.

Denoting the data vectors by $\mathbf{D}_i = (W_{i,1}, Y_{i,1}, Y_{i,2}, W_{i,2}, Y_{i,3}, Y_{i,4})^\top$, the covariance matrix $\boldsymbol{\Sigma} = cov(\mathbf{D}_i)$ is

$$[\sigma_{ij}] = \begin{pmatrix} \phi_{11}+\omega_1 & \beta_1\phi_{11} & \beta_2\phi_{11} & \phi_{12} & \beta_3\phi_{12} & \beta_4\phi_{12} \\ & \beta_1^2\phi_{11}+\psi_1 & \beta_1\beta_2\phi_{11} & \beta_1\phi_{12} & \beta_1\beta_3\phi_{12} & \beta_1\beta_4\phi_{12} \\ & & \beta_2^2\phi_{11}+\psi_2 & \beta_2\phi_{12} & \beta_2\beta_3\phi_{12} & \beta_2\beta_4\phi_{12} \\ & & & \phi_{22}+\omega_2 & \beta_3\phi_{22} & \beta_4\phi_{22} \\ & & & & \beta_3^2\phi_{22}+\psi_3 & \beta_3\beta_4\phi_{22} \\ & & & & & \beta_4^2\phi_{22}+\psi_4 \end{pmatrix}$$

Disregarding the expected values, the parameter[49] is

$$\boldsymbol{\theta} = (\beta_1, \beta_2, \beta_3, \beta_4, \phi_{11}, \phi_{12}, \phi_{22}, \omega_1, \omega_2, \psi_1, \psi_2, \psi_3, \psi_4).$$

Since $\boldsymbol{\theta}$ has 13 elements and $\boldsymbol{\Sigma}$ has $\frac{6(6+1)}{2} = 21$ variances and non-redundant covariances, this problem easily passes the test of the parameter count rule. Provided the parameter vector is identifiable, the model will impose $21 - 13 = 8$ over-identifying restrictions on $\boldsymbol{\Sigma}$.

First notice that if $\phi_{12} \neq 0$, all the regression coefficients are immediately identifiable. Since the extra variables $Y_2$ and $Y_4$ are presumably well-chosen, it may be assumed that

[49] Since the distributions of the random variables in the model are unspecified, one could say that they are also unknown parameters. In this case, the quantity $\boldsymbol{\theta}$ is really a function of the full parameter vector, even after the re-parameterization of intercepts and expected values.

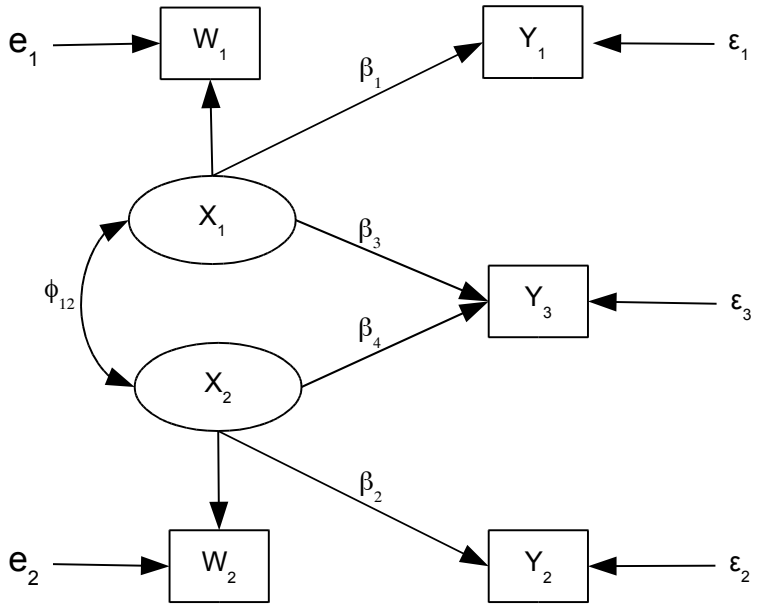
$\beta_2 \neq 0$ and $\beta_4 \neq 0$. In that case, the entire parameter vector is identifiable — for example identifying $\phi_{11}$ from $\sigma_{12}$ and then $\omega_1$ from $\sigma_{11}$ ....

Since it is very common for explanatory variables to be related to one another in non-experimental studies, assumptions like $\phi_{12} \neq 0$ are very reasonable, and in any case are testable as part of an exploratory data analysis. So, extension of this design to data sets with more than two explanatory variables is straightforward, and identifiability follows without detailed calculations.

**Example 0.11.5** *Two explanatory variables, one response variable of primary interest, and one extra response variable for each explanatory variable.*

In this example, each explanatory variable has its own extra response variable, but they share a response variable of primary interest. This is more interesting, because now one can speak of one explanatory variable *controlling* for the other, as in ordinary regression. Figure 19 shows the path diagram.

Figure 19: Two explanatory variables with one extra response variable each, plus a single response variable of interest




The formal statement of this model dispenses with intercepts and expected values. They are really present, but because they are not identifiable separately, they are not even mentioned. This is common in structural equation modeling. Independently for

$i = 1, \ldots, n$, let

$$\begin{aligned} W_{i,1} &= X_{i,1} + e_{i,1} \\ W_{i,2} &= X_{i,2} + e_{i,2} \\ Y_{i,1} &= \beta_1 X_{i,1} + \epsilon_{i,1} \\ Y_{i,2} &= \beta_2 X_{i,2} + \epsilon_{i,2} \\ Y_{i,3} &= \beta_3 X_{i,1} + \beta_4 X_{i,2} + \epsilon_{i,3} \end{aligned}$$

where

- The $X_{i,j}$ variables are latent, while the $W_{i,j}$ and $Y_{i,j}$ variables are observable.
- $e_{i,1} \sim N(0, \omega_1)$ and $e_{i,2} \sim N(0, \omega_2)$.
- $\epsilon_{i,j} \sim N(0, \psi_j)$ for $j = 1, 2, 3$.
- $e_{i,j}$ and $\epsilon_{i,j}$ are independent of each other and of $X_{i,j}$.
- $X_{i,j}$ have covariance matrix

$$cov \begin{pmatrix} X_{i,1} \\ X_{i,2} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}.$$

Denote the vector of observable data by $\mathbf{D}_i = (W_{i,1}, Y_{i,1}, W_{i,2}, Y_{i,2}, Y_{i,3})^\top$, with $cov(\mathbf{D}_i) = \mathbf{\Sigma} = [\sigma_{ij}]$.

Among other things, this example illustrates how the search for identifiability can be supported by exploratory data analysis. Hypotheses about *single* covariances, like $H_0 : \sigma_{ij} = 0$ can be tested by looking at tests of the corresponding correlations. These tests, including non-parametric tests based on the Spearman rank correlation, are easily obtained using R's `cor.test` function.

The parameter vector[50] for this problem is $\boldsymbol{\theta} = (\phi_{11}, \phi_{12}, \phi_{22}, \omega_1, \omega_2, \beta_1, \beta_2, \beta_3, \beta_4, \psi_1, \psi_2, \psi_3)^\top$. There are 12 parameters and 5 observable variables, so that the covariance matrix has $5(5+1)/2 = 15$ unique variances and covariances. Thus there are 15 covariance structure equations in 12 unknowns, and the parameter count rule tells us that identifiability in most of the parameter space is possible but not guaranteed.

The matrix equation 64 shows the covariance structure equations in a compact form.

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} \\ & \sigma_{22} & \sigma_{23} & \sigma_{24} & \sigma_{25} \\ & & \sigma_{33} & \sigma_{34} & \sigma_{35} \\ & & & \sigma_{44} & \sigma_{45} \\ & & & & \sigma_{55} \end{pmatrix} = \tag{64}$$

[50] That is, the vector of parameters appearing in $\mathbf{\Sigma} = cov(\mathbf{D}_i)$.

$$\begin{pmatrix} \omega_1+\phi_{11} & \beta_1\phi_{11} & \phi_{12} & \beta_2\phi_{12} & \beta_3\phi_{11}+\beta_4\phi_{12} \\ & \beta_1^2\phi_{11}+\psi_1 & \beta_1\phi_{12} & \beta_1\beta_2\phi_{12} & \beta_1(\beta_3\phi_{11}+\beta_4\phi_{12}) \\ & & \omega_2+\phi_{22} & \beta_2\phi_{22} & \beta_3\phi_{12}+\beta_4\phi_{22} \\ & & & \beta_2^2\phi_{22}+\psi_2 & \beta_2(\beta_3\phi_{12}+\beta_4\phi_{22}) \\ & & & & (\beta_3\phi_{11}+\beta_4\phi_{12})\beta_3+(\beta_3\phi_{12}+\beta_4\phi_{22})\beta_4+\psi_3 \end{pmatrix}$$

In our study of identifiability for this example, we will confine our attention to that part of the parameter space where $\beta_1 \neq 0$ and $\beta_2 \neq 0$. After all, the variables $Y_1$ and $Y_2$ were introduced only to help with identifiability, and they are useless unless they are related to the explanatory variables. The issue may be resolved empirically by testing $H_0 : \sigma_{23} = 0$ and $H_0 : \sigma_{14} = 0$ with `cor.test`. One should proceed to model fitting only if both null hypotheses are comfortably rejected. In any case, the rest of this discussion assumes that $\beta_1$ and $\beta_2$ are both non-zero.

The parameter $\phi_{12}$ is identifiable, since $\phi_{12} = \sigma_{13}$. Consider two cases. The first case is $\phi_{12} \neq 0$. In this region of the parameter space, $\beta_1$ is identified from $\beta_1 = \sigma_{23}/\phi_{12}$, and $\beta_2$ is identified from $\beta_2 = \sigma_{14}/\phi_{12}$. Then, $\phi_{11} = \sigma_{12}/\beta_1$ and $\phi_{22} = \sigma_{34}/\beta_2$.

With $\phi_{11}$, $\phi_{12}$ and $\phi_{22}$ identified, they may be treated as known. Then, $\beta_3$ and $\beta_4$ are identified from $\sigma_{14}$ and $\sigma_{34}$ by solving two linear equations in two unknowns. Writing the equations in matrix form,

$$\begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix} \begin{pmatrix} \beta_3 \\ \beta_4 \end{pmatrix} = \begin{pmatrix} \sigma_{14} \\ \sigma_{34} \end{pmatrix}.$$

There is a unique solution if and only if the covariance matrix of the latent explanatory variables has an inverse, which is not much to ask. At this point, all parameters have been identified except the variances of the $e_{ij}$ and $\epsilon_{ij}$. Accordingly, $\omega_1$, $\psi_1$, $\omega_2$, $\psi_2$ and $\psi_3$ are obtained from the diagonal elements of $\mathbf{\Sigma}$, by subtraction. The conclusion is that all parameters are identifiable provided $\phi_{12} \neq 0$. In most observational studies, explanatory variables will be correlated. That means the parameters of this model are identifiable for most applications.

Now consider the case where $\phi_{12} = 0$; that is, the latent explanatory variables are uncorrelated. This might apply in a designed experiment with random assignment. The covariance structure equations are now

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} \\ & \sigma_{22} & \sigma_{23} & \sigma_{24} & \sigma_{25} \\ & & \sigma_{33} & \sigma_{34} & \sigma_{35} \\ & & & \sigma_{44} & \sigma_{45} \\ & & & & \sigma_{55} \end{pmatrix} = \tag{65}$$

$$\begin{pmatrix} \omega_1+\phi_{11} & \beta_1\phi_{11} & 0 & 0 & \beta_3\phi_{11} \\ & \beta_1^2\phi_{11}+\psi_1 & 0 & 0 & \beta_1\beta_3\phi_{11} \\ & & \omega_2+\phi_{22} & \beta_2\phi_{22} & \beta_4\phi_{22} \\ & & & \beta_2^2\phi_{22}+\psi_2 & \beta_2\beta_4\phi_{22} \\ & & & & \beta_3^2\phi_{11}+\beta_4^2\phi_{22}+\psi_3 \end{pmatrix}.$$

The parameter $\phi_{12}$ is still identifiable from $\sigma_{13}$, but three equations are lost since $\phi_{12} = 0$ also implies $\sigma_{14} = \sigma_{23} = \sigma_{24} = 0$. Thus there are eleven equations in the eleven remaining unknown parameters. The condition of the the parameter count rule is satisfied, and identifiability of the entire parameter vector is still possible.

Using (65), $\beta_3 = \sigma_{25}/\sigma_{12}$ and $\beta_4 = \sigma_{45}/\sigma_{34}$. If $\beta_3$ and $\beta_4$ are non-zero, solution for the rest of the parameters is routine. But if $\beta_3 = 0$, then $\beta_1$ is no longer identifiable. Similarly, if $\beta_4 = 0$, then $\beta_2$ is no longer identifiable. Since the whole point of this model is likely to test something like $H_0 : \beta_3 = 0$, it's important to examine the situation where this null hypothesis is true.

Suppose one could be sure that $Cov(X_i, X_2) = \phi_{12} = 0$, and consider the problem of testing $H_0 : \beta_3 = 0$. The first thought might be to just compare the likelihood ratio test statistic to a chi-squared critical value with one degree of freedom. As in Example 0.11.1 (one extra response variable), this won't work. In Wilks' (1938) proof of the likelihood ratio test [70], identifiability under the null hypothesis is regularity condition zero, and we are in a situation that Davison [20] (pp. 144-48) would call non-regular. As a practical matter, the numerical search for the restricted MLE (restricted by $H_0$) will not converge except by a numerical fluke. As in the little simulation study on page 109, there is also likely to be trouble fitting even the unrestricted model. If by chance the search for an unrestricted MLE were to converge, the the theory behind $Z$-test of $H_0 : \beta_3 = 0$ fails, because it is equivalent to a Wald test.

Instead, look at equality (65) and observe that $\beta_3 = 0$ implies both $\sigma_{15} = 0$ and $\sigma_{25} = 0$. This hypothesis may be tested using a likelihood ratio or Wald test, with two degrees of freedom. Again, the moral of this story is that the study of identifiability should specifically consider those parts of the parameter space where important null hypotheses are true.

Also, be aware that the models presented here are actually re-parameterizations of models with measurement error in the response variables. One must carefully consider the methods of data collection to rule out correlation between measurement error in the explanatory variables and measurement error in the response variables. Such correlations would appear as non-zero covariances between $e_{ij}$ and $\epsilon_{ij}$ terms in the models, and it will be seen in homework how this can sink the ship on a technical level.

Just to be clear, when data are collected by a common method in a common setting, errors of measurement will naturally be correlated with one another. For example, in a study investigating the connection between diet and athletic accomplishment in children, suppose the data all came from questionnaires filled out by parents. It would be very natural for some parents to exaggerate the healthfulness of the food they serve and also to exaggerate their children's athletic achievements. On the other extreme, some parents would immediately figure out the purpose of the study, and tell the interviewers what they want to hear. "My kids eat junk (I can't control them) and they are terrible in sports." Both these tendencies would produce a positive covariance between the measurement errors in the explanatory and response variables. And in the absence of other information, it would be impossible to tell whether a positive relationship between observable diet and athletic performance came from this, or from an actual relationship between the latent variables.

## 0.12 Instrumental Variables Again

In Section 0.5, the method of instrumental variables was introduced as a solution to the problems that arise when explanatory variables that are missing from the model cause non-zero covariances between the error term and variables that are in the model. We will now see that instrumental variables can help with measurement error too.

Recall Example 0.5.1 in Section 0.5; see page 36. The interest was in the relationship of income to credit card debt. In the imaginary study, data were collected on real estate agents in a variety of towns and cities. In addition to income ($X_i$) and credit card debt ($Y_i$), we had an instrumental variable ($Z_i$) — the median selling price of a home in the agent's region. With the instrumental variable, everything worked out beautifully. The parameters were just identifiable, with nine covariance structure equations in nine unknown parameters.

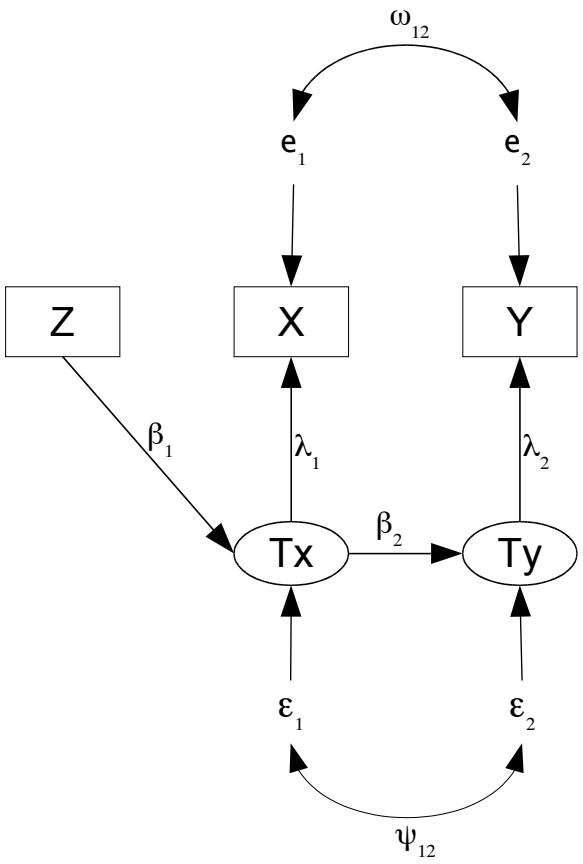
The problem is that both income and debt are undoubtedly measured with error, and there are almost surely other unmeasured variables that affect them both. Figure 20 represents a more realistic model. Omitted variables affecting both true $X$ and true $Y$ give rise to covariance $\psi_{12}$ between the error terms $\epsilon_1$ and $\epsilon_2$. Common omitted variables are also affecting measurement of $X$ and measurement of $Y$, which are both likely to be self-report. This gives rise to the covariance $\omega_{12}$ between the measurement error terms $e_1$ and $e_2$. The regression coefficients $\lambda_1$ and $\lambda_2$ linking true income ($Tx$) to observed income ($X$) and true credit card debt ($Ty$) are positive, but unknown and unlikely to equal one. We now have six covariance structure equations in eleven unkowns, and still it's not realistic enough, because housing prices are only estimated.

The model shown in Figure 21 is easier to defend, but impossible to estimate. By a mysterious process possibly involving multiple variables, the publicly available median resale price of a home is dynamically related to a latent variable or set of variables that positively affect the real estate agent's income.

Fortunately, an instrumental variable only has to be *correlated* with the explanatory variable. As long as we are confident that the covariance between resale price and income is positive (and we are) everything will be okay. Figure 22 acknowledges our ignorance of the exact process by which which median resale price to connected to income, representing the connection with an *un-analyzed covariance* represented by a curved, double-headed arrow. Since the model no longer explicitly posits that true latent income is *affected* by any variable in the model, the operation of common omitted variables on $Tx$ and $Ty$ is now represented by a curved, double-headed arrow connecting $Tx$ and $\epsilon$.

The model of Figure 22 is fairly realistic, but on first examination it does not look promising. There are six covariance structure equations in 11 unknowns. This model fails the parameter count rule, which is poison. The explanatory variable is correlated with the error term, which is another flavour of poison. In addition, errors of measurement are correlated, which is yet another form of poison. However, we have an instrumental variable. Let's calculate the covariance matrix of the observable variables, bearing in

Figure 20: $Z$ is median price of resale home, $X$ is income, $Y$ is credit card debt




mind that $\beta$ is the parameter of primary interest. Showing part of the calculation,

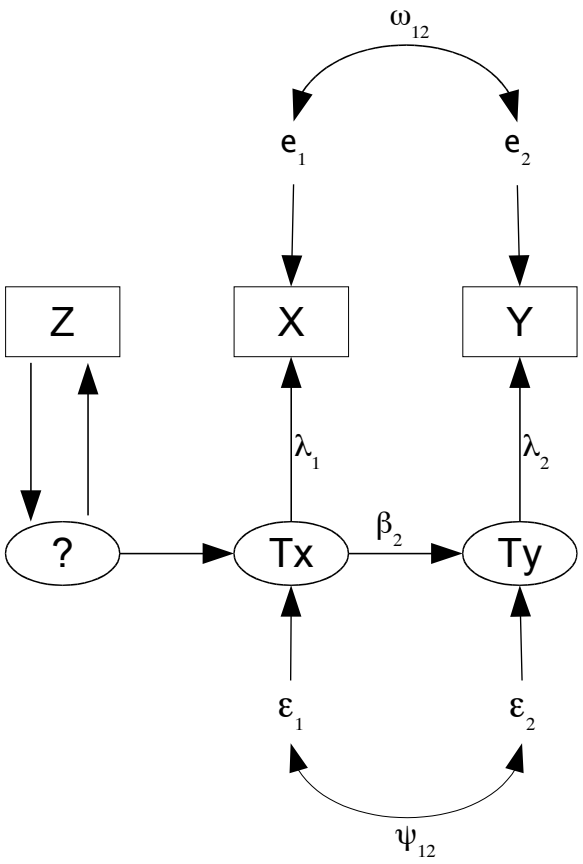
$$\begin{aligned} Cov(Z,Y) &= Cov(Z, \lambda_2 Ty + e_2) \\ &= Cov(Z, \lambda_2(\beta Tx + \epsilon) + e_2) \\ &= \lambda_2 \beta Cov(Z, Tx) + \lambda_2 Cov(Z, \epsilon) + Cov(Z, e_2) \\ &= \lambda_2 \beta \phi_{12} + 0 + 0 \end{aligned}$$

The full covariance matrix is

$$cov \begin{pmatrix} Z \\ X \\ Y \end{pmatrix} = \begin{pmatrix} \phi_{11} & \lambda_1\phi_{12} & \beta\lambda_2\phi_{12} \\ \cdot & \lambda_1^2\phi_{22} + \omega_{11} & \beta\lambda_1\lambda_2\phi_{22} + c\lambda_1\lambda_2 + \omega_{12} \\ \cdot & \cdot & \beta^2\lambda_2^2\phi_{22} + 2\,\beta c\lambda_2^2 + \lambda_2^2\psi + \omega_{22} \end{pmatrix}.$$

The primary parameter $\beta$ is not identifiable, but $\phi_{12}$ (the covariance between median home price and real estate agent income) is positive, and $\lambda_2$ (the link between true income and reported income) is also greater than zero. So the sign of $\beta$ is identifiable from $\sigma_{13}$, the null hypothesis $H_0 : \beta = 0$ is testable by simply testing whether $\sigma_{13}$ is different from zero, and it is possible to answer the basic question of the study.

Figure 21: More realistic, but impossible to estimate

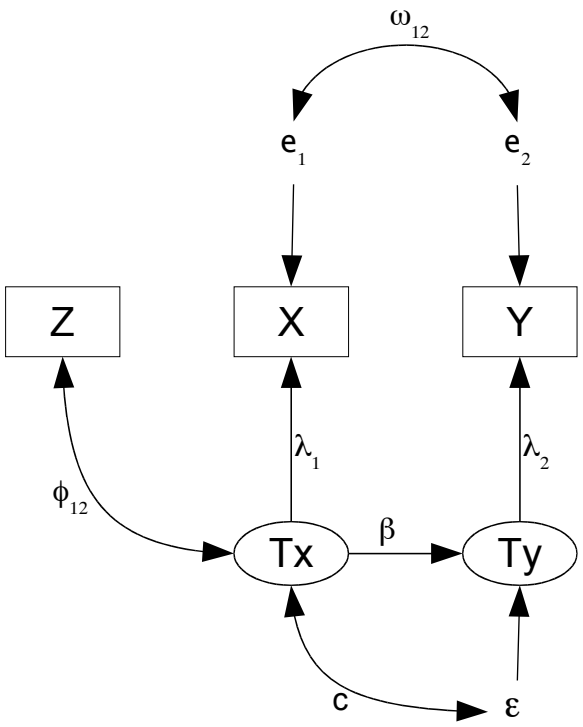



It's a miracle. Instrumental variables can help with measurement error and omitted variables at the same time. If there is measurement error, the regression coefficients of interest are not identifiable and cannot be estimated consistently, but their signs can. Often, that's all you really need to know. A matrix version is available. The usual rule in Econometrics is at least one instrumental variable for each explanatory variable. As you will see in homework, the main technical requirement is that the $p \times p$ matrix of covariances between $\mathbf{X}$ and $\mathbf{Z}$ must have an inverse.

Zero covariance between the instrumental variable and error terms is critical. Since non-zero covariances arise naturally from omitted variables, this means instrumental variables need to come from another world, and are related to to $\mathbf{x}$ for reasons that are *separate* from why $\mathbf{x}$ is related to $\mathbf{y}$. For example, consider the question of whether academic ability contributes to higher salary. Study adults who were adopted as children. $x$ is academic ability, $y$ is salary at age 40, $W$ is measured IQ at 40, and the instrumental variable $z$ is birth mother's IQ score.

The method of instrumental variables is a solution to the problems of omitted variables and measurement error, but it's a partial solution. Good instrumental variables are not

Figure 22: An improved model of income and credit card debt




easy to find. They will almost certainly not be in a data set casually collected for other purposes. Advance planning is needed.

In many textbook examples of instrumental variables, the instrumental variable arguably has a causal impact on the corresponding explanatory variable. That is, one can argue for a straight arrow running from $Z$ to $X$. Here is a nice example from the *Wikipedia* article on "natural experiments" []. The idea behind a natural experiment is that nature, rather than the investigator, assigns the study participants to treatment conditions. And, while the assignment may not be exactly random, it is at least unlikely to be connected to plausible confounding variables. Here's the story, quoted from the Wikipedia.

> One of the best-known early natural experiments was the 1854 Broad Street cholera outbreak in London, England. On 31 August 1854, a major outbreak of cholera struck Soho. Over the next three days, 127 people near Broad Street died. By the end of the outbreak 616 people died. The physician John Snow identified the source of the outbreak as the nearest public water pump, using a map of deaths and illness that revealed a cluster of cases around the pump.
>
> In this example, Snow discovered a strong association between the use of the water from the pump, and deaths and illnesses due to cholera. Snow found that the Southwark and Vauxhall Waterworks Company, which supplied water to districts with high attack rates, obtained the water from the Thames down-

> stream from where raw sewage was discharged into the river. By contrast, districts that were supplied water by the Lambeth Waterworks Company, which obtained water upstream from the points of sewage discharge, had low attack rates. Given the near-haphazard patchwork development of the water supply in mid-nineteenth century London, Snow viewed the developments as "an experiment ... on the grandest scale."

So, the explanatory variable $x$ was drinking and otherwise using water containing raw sewage, the response variable $y$ was getting cholera, and the instrumental variable $z$ was the company that supplied the water. The critical fact that makes it a good instrumental variable is the "... near-haphazard patchwork development of the water supply in mid-nineteenth century London." (We will gladly take their word for it.) Seemingly, the configuration of the water supply was so chaotic that it was unlikely to be related to other plausible influences on getting cholera, like social class and income. Thus, one can argue for the absence of any curved arrows connecting the instrumental variable to the error terms. From both a technical and common-sense viewpoint, that's what makes the whole thing work.

The Wikipedia article has several other good examples of natural experiments, and they are also good examples of instrumental variables. In fact, one could say that the ultimate instrumental variable is randomly assigned; in that case, it's guaranteed to come from another world, and if the experiment is otherwise well-controlled, connections between omitted variables and the treatment are entirely ruled out.

But for better or worse, we are concerned with cases where ethics or simple practical considerations dictate that we cannot control the values of the explanatory variables. Our data come from observational studies. If the data set contains good instrumental variables, many of our difficulties will disappear, but we cannot just manufacture them. We must discover and notice them as they naturally occur, and this requires a bit of good luck, as well as a sharp eye and flexible thinking.

## 0.13 Exercises for Chapter 0

- Exercises 0.2: Conditional and unconditional regression

1. Everybody knows that $Var(Y_i) = \sigma^2$ for a regression model, but that's really a conditional variance. Independently for $i = 1, \ldots, n$, let

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i,$$

   where $\epsilon_1, \ldots \epsilon_n$ are independent random variables with expected value zero and common variance $\sigma^2$, $E(X_{i,1}) = \mu_1$, $Var(X_{i,1}) = \sigma_1^2$, $E(X_{i,2}) = \mu_2$, $Var(X_{i,2}) = \sigma_2^2$, and $Cov(X_{i,1}, X_{i,2}) = \sigma_{12}$. Calculate $Var(Y_i)$; show your work.

2. Suppose that the model (3) has an intercept. How many integral signs are there in the second line of (6)? The answer is a function of $n$ and $p$.

3. The usual univariate multiple regression model with independent normal errors is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{X}$ is an $n \times p$ matrix of known constants, $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown constants, and $\boldsymbol{\epsilon}$ is multivariate normal with mean zero and covariance matrix $\sigma^2\mathbf{I}_n$, with $\sigma^2 > 0$ an unknown constant. But of course in practice, the explanatory variables are random, not fixed. Clearly, if the model holds *conditionally* upon the values of the explanatory variables, then all the usual results hold, again conditionally upon the particular values of the explanatory variables. The probabilities (for example, $p$-values) are conditional probabilities, and the $F$ statistic does not have an $F$ distribution, but a conditional $F$ distribution, given $\mathbf{X} = \mathbf{x}$.

   (a) Show that the least-squares estimator $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ is conditionally unbiased.

   (b) Show that $\widehat{\boldsymbol{\beta}}$ is also unbiased unconditionally.

   (c) A similar calculation applies to the significance level of a hypothesis test. Let $F$ be the test statistic (say for an extra-sum-of-squares $F$-test), and $f_c$ be the critical value. If the null hypothesis is true, then the test is size $\alpha$, conditionally upon the explanatory variable values. That is, $P(F > f_c|\mathbf{X} = \mathbf{x}) = \alpha$. Find the *unconditional* probability of a Type I error. Assume that the explanatory variables are discrete, so you can write a multiple sum.

- Exercises **??**: The Centering Rule

  Maybe refer to some exercises from the Appendix.

- Exercises 0.4: Omitted variables

1. In the following regression model, the independent variables $X_1$ and $X_2$ are random variables. The true model is

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i,$$

independently for $i = 1, \ldots, n$, where $\epsilon_i \sim N(0, \sigma^2)$.
The mean and covariance matrix of the independent variables are given by

$$E\begin{pmatrix} X_{i,1} \\ X_{i,2} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \text{and} \quad Var\begin{pmatrix} X_{i,1} \\ X_{i,2} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}$$

Unfortunately $X_{i,2}$, which has an impact on $Y_i$ and is correlated with $X_{i,1}$, is not part of the data set. Since $X_{i,2}$ is not observed, it is absorbed by the intercept and error term, as follows.

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i \\ &= (\beta_0 + \beta_2\mu_2) + \beta_1 X_{i,1} + (\beta_2 X_{i,2} - \beta_2\mu_2 + \epsilon_i) \\ &= \beta_0' + \beta_1 X_{i,1} + \epsilon_i'. \end{aligned}$$

The primes just denote a new $\beta_0$ and a new $\epsilon_i$. It was necessary to add and subtract $\beta_2\mu_2$ in order to obtain $E(\epsilon_i') = 0$. And of course there could be more than one omitted variable. They would all get swallowed by the intercept and error term, the garbage bins of regression analysis.

(a) What is $Cov(X_{i,1}, \epsilon_i')$?

(b) Calculate the variance-covariance matrix of $(X_{i,1}, Y_i)$ under the true model.

(c) Suppose we want to estimate $\beta_1$. The usual least squares estimator is

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (X_{i,1} - \overline{X}_1)(Y_i - \overline{Y})}{\sum_{i=1}^n (X_{i,1} - \overline{X}_1)^2}.$$

You may just use this formula; you don't have to derive it. Is $\widehat{\beta}_1$ a consistent estimator of $\beta_1$ for all points in the parameter space if the true model holds? Answer Yes or no and show your work. Remember, $X_2$ is not available, so you are doing a regression with one independent variable. You may use the consistency of the sample variance and covariance without proof.

(d) Are there *any* points in the parameter space for which $\widehat{\beta}_1$ is a consistent estimator when the true model holds?

2. Ordinary least squares is often applied to data sets where the independent variables are best modeled as random variables. In what way does the usual conditional linear regression model imply that (random) independent variables have zero covariance with the error term? Hint: Assume $\mathbf{X}_i$ as well as $\epsilon_i$ continuous. What is the conditional distribution of $\epsilon_i$ given $\mathbf{X}_i = \mathbf{x}_i) = 0$?

3. Show that $E(\epsilon_i | X_i = x_i) = 0$ for all $x_i$ implies $Cov(X_i, \epsilon_i) = 0$, so that a standard regression model without the normality assumption still implies zero covariance (though not necessarily independence) between the error term and explanatory variables.

- Exercises 0.6: Measurement error

1. Calculate expression (29) for the reliability, showing the details that were skipped. The point of this question (besides exercising your variance-covariance muscles and keeping you busy so you don't have a personal life) is to see whether you feel comfortable assuming $\mu = 0$ even though it may not be.

2. In a study of diet and health, suppose we want to know how much snack food each person eats, and we "measure" it by asking a question on a questionnaire. Surely there will be measurement error, and suppose it is of a simple additive nature. But we are pretty sure people under-report how much snack food they eat, so a model like $W = X + e$ with $E(e) = 0$ is hard to defend. Instead, let

$$W = \nu + X + e,$$

where $E(X) = \mu$, $E(e) = 0$, $Var(X) = \sigma_x^2$, $Var(e) = \sigma_e^2$, and $Cov(X, e) = 0$ The unknown constant $\nu$ could be called *measurement bias*. Calculate the

reliability of $W$ for this model. Is it the same as (29), or does $\nu \neq 0$ make a difference?

3. Continuing Exercise 2, suppose that two measurements of $W$ are available.

$$\begin{aligned} W_1 &= \nu_1 + X + e_1 \\ W_2 &= \nu_2 + X + e_2, \end{aligned}$$

where $E(X) = \mu$, $Var(X) = \sigma^2_T$, $E(e_1) = E(e_2) = 0$, $Var(e_1) = Var(e_2) = \sigma^2_e$, and $X$, $e_1$ and $e_2$ are all independent. Calculate $Corr(W_1, W_2)$. Does this correlation still equal the reliability?

4. Let $X$ be a latent variable, $W = X + e_1$ be the usual measurement of $X$ with error, and $G = X + e_2$ be a measurement of $X$ that is deemed "gold standard," but of course it's not completely free of measurement error. It's better than $W$ in the sense that $0 < Var(e_2) < Var(e_1)$, but that's all you can really say. This is a realistic scenario, because nothing is perfect. Accordingly, let

$$\begin{aligned} W &= X + e_1 \\ G &= X + e_2, \end{aligned}$$

where $E(X) = \mu$, $Var(X) = \sigma^2_x$, $E(e_1) = E(e_2) = 0$, $Var(e_1) = \sigma^2_1$, $Var(e_2) = \sigma^2_2$ and that $X$, $e_1$ and $e_2$ are all independent of one another. Prove that the squared correlation between $W$ and $G$ is strictly less than the reliability of $W$. Show your work.

The idea here is that the squared *population* correlation[51] between an ordinary measurement and an imperfect gold standard measurement is strictly less than the actual reliability of the ordinary measurement. If we were to estimate such a squared correlation by the corresponding squared *sample* correlation, all we would be doing is estimating a quantity that is not the reliability. On the other hand, we would be estimating a lower bound for the reliability — and this could be reassuring if it is a high number.

5. In this continuation of Exercise 4, show what happens when you calculate the squared *sample* correlation between a usual measurement and an imperfect gold standard, and let $n \to \infty$. It's just what you would think.

6. Suppose we have two equivalent measurements with uncorrelated measurement error:

$$\begin{aligned} W_1 &= X + e_1 \\ W_2 &= X + e_2, \end{aligned}$$

[51] When we do Greek-letter calculations, we are figuring out what is happening in the population from which a data set might be a random sample.

where $E(X) = \mu$, $Var(X) = \sigma_x^2$, $E(e_1) = E(e_2) = 0$, $Var(e_1) = Var(e_2) = \sigma_e^2$, and $X$, $e_1$ and $e_2$ are all independent. What if we were to measure the true score $X$ by adding the two imperfect measurements together? Would the result be more reliable?

(a) Let $S = W_1 + W_2$. Calculate the reliability of $S$. Is there any harm in assuming $\mu = 0$?

(b) Suppose you take $k$ independent measurements (in psychometric theory, these would be called equivalent test items). What is the reliability of $S = \sum_{i=1}^{k} W_i$? Show your work.

(c) What happens as the number of measurements $k \to \infty$?

This exercise establishes the well-known principle that longer tests tend to be more reliable. The measurement of practically anything can be improved by measuring it independently several times and then averaging the results — assuming this is possible.

7. Suppose we have two equivalent measurements with *correlated* measurement error:

$$\begin{aligned} W_1 &= X + e_1 \\ W_2 &= X + e_2, \end{aligned}$$

where $E(X) = \mu$, $Var(X) = \sigma_x^2$, $E(e_1) = E(e_2) = 0$, $Var(e_1) = Var(e_2) = \sigma_e^2$, and $e_1$ and $e_2$ are all independent of $X$ but $Cov(e_1, e_2) = \kappa$. Calculate $Corr(W_1, W_2)$; show your work. What is the relationship of your answer to the reliability if $\kappa > 0$ (which is typical of correlated measurement error)? The point of this question is that correlated measurement errors are more the rule than the exception in practice, and it's poison.

- Exercises 0.7: Ignoring measurement error

1. The following is perhaps the simplest example of what happens to regression when there is measurement error in the explanatory variable. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} Y_i &= X_i\beta + \epsilon_i \\ W_i &= X_i + e_i, \end{aligned}$$

where $E(X_i) = E(\epsilon_i) = 0$, $Var(X_i) = \sigma_x^2$, $Var(\epsilon_i) = \sigma_\epsilon^2$, $Var(e_i) = \sigma_e^2$, and $X_i$, $\epsilon_i$ and $e_i$ are all independent. Notice that $W_i$ is just $X_i$ plus a piece of random noise. This is a simple additive model of measuremnt error.

Unfortunately, we cannot observe the $X_i$ values. All we can see are the pairs $(X_i, W_i)$ for $i = 1, \ldots, n$. So we do what everybody does, and fit the *naive* (mis-specified, wrong) model

$$Y_i = W_i\beta + \epsilon_i$$

and estimate $\beta$ with the usual formula for regression through the origin. Where does $\widehat{\beta}_n$ go as $n \to \infty$? Show your work.

2. Recall the simulation study of inflated Type I error when independent variables are measured with error but one ignores it and uses ordinary regression anyway. We needed to produce correlated (latent, that is unobservable) independent variables from different distributions. Here's how we did it.

   (a) It is easy to simulate a collection of independent random variables from any distribution, and then standardize them to have expected value zero and variance one. Let $E(X) = \mu$ and $Var(X) = \sigma^2$. Now define $Z = \frac{X-\mu}{\sigma}$. Find

      i. $E(Z)$
      ii. $Var(Z)$

   (b) Okay, now let $R_1$, $R_2$ and $R_3$ be independent random variables from any distribution you like, but standardized to have expected value zero and variance one. Now let

$$\begin{aligned} W_1 &= \sqrt{1-\phi}\, R_1 + \sqrt{\phi}\, R_3 \text{ and} \\ W_2 &= \sqrt{1-\phi}\, R_2 + \sqrt{\phi}\, R_3. \end{aligned}$$

      Find

      i. $Cov(W_1, W_2)$
      ii. $Corr(W_1, W_2)$

   (c) This one is more efficient. Let $R_1$ and $R_2$ be independent random variables with expected value zero and variance one. Now let

$$\begin{aligned} W_1 &= \sqrt{\frac{1+\phi}{2}}\, R_1 + \sqrt{\frac{1-\phi}{2}}\, R_2 \\ W_2 &= \sqrt{\frac{1+\phi}{2}}\, R_1 - \sqrt{\frac{1-\phi}{2}}\, R_2 \end{aligned}$$

      Find

      i. $Cov(W_1, W_2)$
      ii. $Corr(W_1, W_2)$

   (d) Briefly state how you know the following. No proof is required.

      – If the $R$ variables are normal and $\phi = 0$, both methods yield $X_1$ and $X_2$ independent.
      – But if the $R$s are non-normal, then $\phi = 0$ only implies independence for the first method.

- Exercises 0.8: Modeling measurement error

1. Let $X_1, \ldots, X_n$ be a random sample from a normal distribution with mean $\theta_1$ and variance $\theta_2 + \theta_3$, where $-\infty < \theta_1 < \infty$, $\theta_2 > 0$ and $\theta_3 > 0$. Are the prameters of this model identifiable? Answer Yes or No and prove your answer. This is fast.
2. Let $X_1, \ldots, X_n$ be a random sample from a normal distribution with mean $\theta$ and variance $\theta^2$, where $-\infty < \theta < \infty$. Is $\theta$ identifiable? Answer Yes or No and justify your answer. This is even faster than the last one.
3. For this problem you may want to read about the *invariance principle* of maximum likelihood estimation in Appendix A. Consider the simple regression model
$$Y_i = \beta X_i + \epsilon_i,$$
where $\beta$ is an unknown constant, $X_i \sim N(0, \phi)$, $\epsilon_i \sim N(0, \psi)$ and the random variables $X_i$ and $\epsilon_i$ are independent. $X_i$ and $Y_i$ are observable variables.
   (a) What is the parameter vector $\boldsymbol{\theta}$ for this model? It has three elements.
   (b) What is the distribution of the data vector $(X_i, Y_i)^\top$? Of course the expected value is zero; obtain the covariance matrix in terms of $\boldsymbol{\theta}$ values. Show your work.
   (c) Now solve three equations in three unknowns to express the three elements of $\boldsymbol{\theta}$ in terms of $\sigma_{i,j}$ values.
   (d) Are the parameters of this model identifiable? Answer Yes or No and state how you know.
   (e) For a sample of size $n$, give the MLE $\widehat{\boldsymbol{\Sigma}}$. Your answer is a matrix containing three scalar formulas (or four formulas, if you write down the same thing for $\widehat{\sigma}_{1,2}$ and $\widehat{\sigma}_{2,1}$). Write your answer in terms of $X_i$ and $Y_i$ quantities. You are *not* being asked to derive anything. Just translate the matrix MLE into scalar form.
   (f) Use the invariance principle to obtain the formula for $\widehat{\beta}$ and simplify. Show your work.
   (g) Give the formula for $\widehat{\phi}$. Use the invariance principle.
   (h) Obtain the formula for $\widehat{\psi}$ and simplify. Use the invariance principle. Show your work.
4. Consider the regression model
$$\begin{aligned} Y_{i,1} &= \beta_1 X_i + \epsilon_{i,1} \\ Y_{i,2} &= \beta_2 X_i + \epsilon_{i,2}, \end{aligned}$$
where $X_i \sim N(0, \phi)$, and $X_i$ is independent of $\epsilon_{i,1}$ and $\epsilon_{i,2}$. The error terms $\epsilon_{i,1}$ and $\epsilon_{i,2}$ are bivariate normal, with mean zero and covariance matrix
$$\boldsymbol{\Psi} = \begin{pmatrix} \psi_{1,1} & \psi_{1,2} \\ \psi_{1,2} & \psi_{2,2} \end{pmatrix}.$$
The variables $X_i$, $Y_{i,1}$ and $Y_{i,2}$ are observable; there is no measurement error.

(a) What is the parameter vector $\boldsymbol{\theta}$ for this model? It has six elements.

(b) Calculate the covariance matrix of the observable variables; show your work.

(c) Are the parameters of this model identifiable? Answer Yes or No and justify your answer.

5. Here is a multivariate regression model with no intercept and no measurement error. Independently for $i = 1, \ldots, n$,

$$\mathbf{y}_i = \boldsymbol{\beta}\mathbf{X}_i + \boldsymbol{\epsilon}_i$$

where

$\mathbf{y}_i$ is an $q \times 1$ random vector of observable response variables, so the regression can be multivariate; there are $q$ response variables.

$\mathbf{X}_i$ is a $p \times 1$ observable random vector; there are $p$ explanatory variables. $\mathbf{X}_i$ has expected value zero and variance-covariance matrix $\boldsymbol{\Phi}$, a $p \times p$ symmetric and positive definite matrix of unknown constants.

$\boldsymbol{\beta}$ is a $q \times p$ matrix of unknown constants. These are the regression coefficients, with one row for each response variable and one column for each explanatory variable.

$\boldsymbol{\epsilon}_i$ is the error term of the latent regression. It is an $q \times 1$ random vector with expected value zero and variance-covariance matrix $\boldsymbol{\Psi}$, a $q \times q$ symmetric and positive definite matrix of unknown constants. $\boldsymbol{\epsilon}_i$ is independent of $\mathbf{X}_i$.

Are the parameters of this model identifiable? Answer Yes or No and show your work.

6. Consider the following simple regression through the origin with measurement error in both the explanatory and response variables. Independently for $i = 1, \ldots, n$,

$$\begin{aligned} Y_i &= \beta X_i + \epsilon_i \\ W_{i,1} &= X_i + e_{i,1} \\ W_{i,2} &= X_i + e_{i,2} \\ V_i &= Y_i + e_{i,3} \end{aligned}$$

where $X_i$ and $Y_i$ are latent variables, $\epsilon_i$, $e_{i,1}$, $e_{i,2}$, $e_{i,3}$ and $X_i$ and are independent normal random variables with expected value zero, $Var(X_i) = \phi$, $Var(\epsilon_i) = \psi$, and $Var(e_{i,1}) = Var(e_{i,2}) = Var(e_{i,3}) = \omega$. The regression coefficient $\beta$ is a fixed constant. The observable variables are $W_{i,1}, W_{i,1}$ and $V_i$.

(a) Calculate the variance-covariance matrix of the observable variables. Show your work.

(b) Write down the moment structure equations.

(c) Are the parameters of this model identifiable? Answer Yes or No and prove your answer.

7. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} Y_i &= \beta X_i + \epsilon_i \\ W_i &= X_i + e_i, \end{aligned}$$

where $E(X_i) = \mu \neq 0$, $E(\epsilon_i) = E(e_i) = 0$, $Var(X_i) = \phi$, $Var(\epsilon_i) = \psi$, $Var(e_i) = \omega$, and $X_i$, $e_i$ and$\epsilon_i$ are all independent. The variables $X_i$ is latent, while $W_i$ and $Y_i$ are observable.

(a) Does this model pass the test of the parameter count rule? Answer Yes or No and give the numbers.

(b) Is the parameter vector identifiable? Answer Yes or No and prove your answer. If the answer is No, give a simple example of two different sets of parameter values that yield the same (bivariate normal) distribution of the observable data.

(c) Let

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n W_i Y_i}{\sum_{i=1}^n W_i^2}.$$

Is $\widehat{\beta}_1$ a consistent estimator of $\beta$? Answer Yes or No and prove your answer.

(d) Let

$$\widehat{\beta}_2 = \frac{\sum_{i=1}^n Y_i}{\sum_{i=1}^n W_i}.$$

- Is $\widehat{\beta}_2$ a consistent estimator of $\beta$? Answer Yes or No and justify your answer.
- We know from Theorem 0.1 that consistent estimation is impossible when the parameter is not identifiable. Does this example contradict Theorem 0.1?

8. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} Y_i &= \beta X_i + \epsilon_i \\ W_{i,1} &= X_i + e_{i,1} \\ W_{i,2} &= X_i + e_{i,2}, \end{aligned}$$

where

- $X_i$ is a normally distributed *latent* variable with mean zero and variance $\phi > 0$
- $\epsilon_i$ is normally distributed with mean zero and variance $\psi > 0$
- $e_{i,1}$ is normally distributed with mean zero and variance $\omega_1 > 0$
- $e_{i,2}$ is normally distributed with mean zero and variance $\omega_2 > 0$

  - $X_i$, $\epsilon_i$, $e_{i,1}$ and $e_{i,2}$ are all independent of one another.

(a) What is the parameter vector $\boldsymbol{\theta}$ for this model?

(b) Does this problem pass the test of the parameter count rule? Answer Yes or No and give the numbers.

(c) Calculate the variance-covariance matrix of the observable variables. Show your work.

(d) Is the parameter vector identifiable? Answer Yes or No and prove your answer.

(e) Propose a consistent estimator of the parameter $\beta$, and show it is consistent.

- Exercises 0.9
-
-
-
-

# Chapter 1

# Introduction to Structural Equation Models

The design of this book is for Chapter 0 to be a self-contained discussion of regression with measurement error, while this chapter introduces the classical structural equation models in their full generality. So, this chapter may serve as a starting point for advanced readers. These advanced readers may belong to two species — quantitatively oriented social scientists who are already familiar with structural equation modeling, and statisticians looking for a quick introduction to the topic at an appropriate level.

Also, readers of Chapter 0 will have noticed that the study of a particular model typically involves a fair amount of symbolic calculation, particularly the calculation of covariance matrices in terms of model parameters. While these calculations often yield valuable insights, they become increasingly burdensome as the number of variables increases, particularly when more than one model must be considered.

The solution is to let a computer do it. So starting with this chapter, many calculations will be illustrated using Sage, an open source computer algebra package described in Appendix B. The Sage parts will be interleaved with the rest of the text rather than fully integrated. Typically, an example will include the result of a calculation without giving a lot of detail, and then at an appropriate place for a pause, the Sage code will be given. This will allow readers who are primarily interested in the ideas to skip material they may find tedious.

## 1.1 Overview

Structural equation models may be viewed as an extension of multiple regression. They generalize multiple regression in three main ways: there is usually more than one equation, a response variable in one equation can be an explanatory variable in another, and structural equation models can include latent variables.

**Multiple equations**: Structural equation models are usually based upon more than one regression-like equation. Having more than one equation is not really

unique; multivariate regression already does that. But you will see that structural equation models are more flexible than the usual multivariate linear model.

**Variables can be both explanatory and response**: This is an attractive feature. Consider a study of arthritis patients, in which joint pain and mobility are measured at several time points. Joint pain at one time period can lead to decreased physical activity during the same period, which then leads to more pain at the next time period. Level of physical activity at time $t$ is both a response variable and a response variable. Structural equation models are also capable of representing the back-and-forth nature of supply and demand in Economics. Many other examples will be given

**Latent variables**: Structural equation models may include random variables that cannot be directly observed, and also are not error terms. This capability (combined with relative simplicity) is their biggest advantage. It allows the statistican to admit that measurement error exists, and to incorporate it directly into the statistical model. The regression models with latent variables in Chapter 0 are special cases of structural equation models.

There are some ways that structural equation models are different from ordinary linear regression. These include random (rather than fixed) explanatory variable values, a bit of specialized vocabulary, and some modest changes in notation. Tests and confidence intervals are based on large-sample theory, even when normal distributions are assumed. Also, structural equation models have a substantive[1] as well as a statistical compontent; closely associated with this is the use of path diagrams to represent the connections between variables.

To the statistician, perhaps the most curious feature of structural equation models is that usually, the regression-like equations lack intercepts and the expected values of all random variables equal zero. This happens because the models have been re-parameterized in search of parameter identifiability. Details are given in the next section (Section A.6.1).

**Random explanatory variables** Chapter 0 discusses the advantages of the traditional regression model in which values of the explanatory variables are treated as fixed constants, and the model is considered to be *conditional* on those values. But once we admit that the variables we observe are contaminated by random measurement error, the virtues of a conditional model mostly disappear. So in the standard structural equation models, all variables are random variables.

**Vocabulary** Structural equation modeling has developed a specialized vocabulary, and except for the term "latent variable," much of it is not seen elsewhere in Statistics. But the terminology can help clarify things once you know it, and also it appears in software manuals and on computer output. Here are some terms and their definitions.

[1] Substantive means having to do with the subject matter. A good substantive model of water pollution would depend on concepts from Chemistry and Hydrodynamics.

- **Latent variable**: A random variable that cannot be directly observed, and also is not an error term.
- **Manifest variable**: An observable variable. An actual data set contains only values of the manifest variables. This book will mostly use the term "observable."
- **Exogenous variable**: In the regression-like equations of a structural equation model, the exogenous variabes are ones that appear *only* on the right side of the equals sign, and never on the left side in any equation. If you think of $Y$ being a function of $X$, this is one way to remember the meaning of **ex**ogenous. All error terms are exogenous variables.
- **Endogenous variable**: Endogenous variables are those that appear on the left side of at least one equals sign. Endogenous variables depend on the exogenous varables, and possibly other endogenous variables. Think of an arrow from an exogenous variable to an endogenous variable. The **end** of the arrow is pointing at the **end**ogenous variable.
- **Factor**: This term has a meaning that actually conflicts with its meaning in mainstream Statistics, particularly in experimental design. Factor analysis (not "factorial" analysis of variance!) is a set of statistical concepts and methods that grew up in Psychology. Factor analysis models are special cases of the general structural equation model. A *factor* is an underlying trait or characteristic that cannot be measured directly, like intelligence. It is a latent variable, period.

**Notation** Several different but overlapping models and accompanying notation systems are to be found in the many books and articles on structural equation modeling. The present book introduces a sort of hybrid notation system, in which the symbols for parameters are mosly taken from the structural equation modeling literature, while the symbols for random variables are based on common statistical usage. This is to make it easier for statisticians to follow. The biggest change from Chapter 0 is that the symbol $\boldsymbol{\beta}$ is no longer used for just any regression coefficient. It is reserved for links between latent endgenous variables and other latent endgenous variables.

## 1.2 A general two-stage model

Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} \mathbf{y}_i &= \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\ \mathbf{F}_i &= \begin{pmatrix} \mathbf{x}_i \\ \hline \mathbf{y}_i \end{pmatrix} \\ \mathbf{d}_i &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \end{aligned} \tag{1.1}$$

where

- $\mathbf{y}_i$ is a $q \times 1$ random vector.
- $\boldsymbol{\alpha}$ is a $q \times 1$ vector of constants.
- $\boldsymbol{\beta}$ is a $q \times q$ matrix of constants with zeros on the main diagonal.
- $\boldsymbol{\Gamma}$ is a $q \times p$ matrix of constants.
- $\mathbf{x}_i$ is a $p \times 1$ random vector with expected value $\boldsymbol{\mu}_x$ and positive definite covariance matrix $\boldsymbol{\Phi}_x$.
- $\boldsymbol{\epsilon}_i$ is a $q \times 1$ random vector with expected value zero and positive definite covariance matrix $\boldsymbol{\Psi}$.
- $\mathbf{F}_i$ ($F$ for Factor) is a partitioned vector with $\mathbf{x}_i$ stacked on top of $\mathbf{y}_i$. It is a $(p+q) \times 1$ random vector whose expected value is denoted by $\boldsymbol{\mu}_F$, and whose variance-covariance matrix is denoted by $\boldsymbol{\Phi}$.
- $\mathbf{d}_i$ is a $k \times 1$ random vector. The expected value of $\mathbf{d}_i$ will be denoted by $\boldsymbol{\mu}$, and the covariance matrix of $\mathbf{d}_i$ will be denoted by $\boldsymbol{\Sigma}$.
- $\boldsymbol{\nu}$ is a $k \times 1$ vector of constants.
- $\boldsymbol{\Lambda}$ is a $k \times (p+q)$ matrix of constants.
- $\mathbf{e}_i$ is a $k \times 1$ random vector with expected value zero and covariance matrix $\boldsymbol{\Omega}$.
- $\mathbf{x}_i$, $\boldsymbol{\epsilon}_i$ and $\mathbf{e}_i$ are independent.

Only $\mathbf{d}_1, \ldots, \mathbf{d}_n$ are observable. All the other random vectors are latent. But because $\boldsymbol{\Omega} = cov(\mathbf{e}_i)$ need not be strictly positive definite, error variances of zero are permitted. This way, it is possible for a variable to be both exogenous and observable.

The distributions of $\mathbf{x}_i$, $\boldsymbol{\epsilon}_i$ and $\mathbf{e}_i$ are either assumed to be independent and multivariate normal, or independent and unknown. When the distributions are normal, the parameter vector $\boldsymbol{\theta}$ consists of the unique elements of the parmeter matrices $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\mu}_x$, $\boldsymbol{\Phi}_x$, $\boldsymbol{\Psi}$, $\boldsymbol{\nu}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$. When the distributions are unknown, the parameter vector also includes the three unknown probability distributions.

The two parts of Model (1.1) are called the *Latent Variable Model* and the *Measurement Model*. The latent variable part is $\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i$, and the measurement part is $\mathbf{d}_i = \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i$. The bridge between the two parts is the process of collecting the latent exogenous vector $\mathbf{x}_i$ and the latent endogenous vector $\mathbf{y}_i$ into a "factor" $\mathbf{F}_i$. This is *not* a categorical explanatory variable, the usual meaning of factor in experimental design. The terminology comes from *factor analysis*, a popular multivariate method in the social sciences. Factor analysis is discussed in Chapters 2 and 3.

**Example 1.2.1** *The Brand Awareness study*

A major Canadian coffee shop chain is trying to break into the U.S. Market. They assess the following variables twice on a random sample of coffee-drinking adults. Each variable is measured first in an in-person interview, and then in a telephone call-back several days later, conducted by a different interviewer. Thus, errors of measurement for the two measurements of each variable are assumed to be independent. The variables are

- **Brand Awareness** ($X_1$): Familiarity with the coffee shop chain
- **Advertising Awareness** ($X_2$): Recall for advertising of the coffee shop chain
- **Interest in the product category** ($X_3$): Mostly this was how much they say they like coffee and doughnuts.
- **Purchase Intention** ($Y_1$): Expressed willingness to go to an outlet of the coffeeshop chain and make an order.
- **Purchase behaviour** ($Y_2$): Reported dollars spent at the chain during the 2 months following the interview.

All variables were measured on a scale from 0 to 100 except purchase behaviour, which is in dollars.

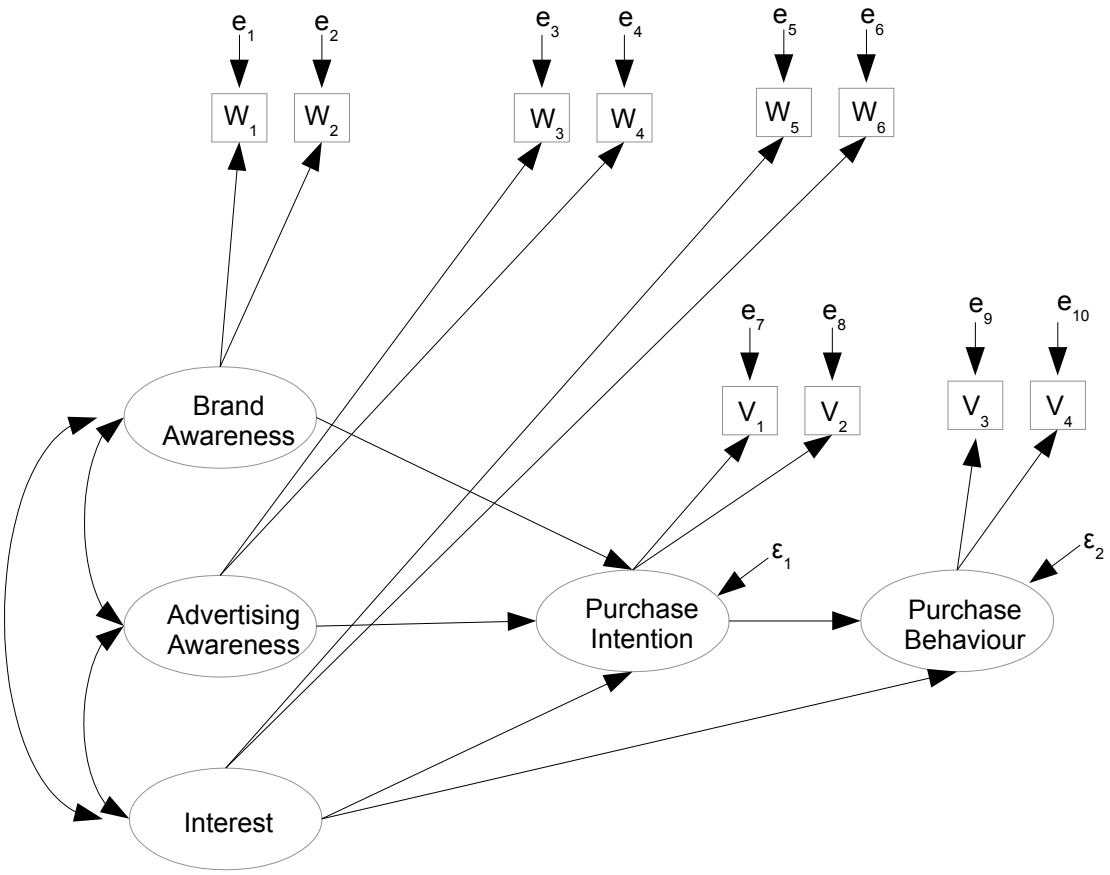
Figure 1.1 shows a path diagram for these data. It is a picture of how some variables are thought to influence other variables. The notation is standard. Straight arrows go from exogenous variables to endogenous variables, and possibly from endogenous variables to other endogenous variables. Correlations among exogenous variables are represented by two-headed curved arrows. Observable variables are enclosed by rectangles or squares, while latent variables are enclosed by ellipses or circles. Error terms are not enclosed by anything.

The path diagram in Figure 1.1 expresses some very definite assertions about consumer behaviour. For example, it says that brand awareness and advertising awareness affect actual purchase only through purchase intention, while interest in the product may have a direct effect on purchase behaviour, as well as an indirect effect through purchase intention — perhaps reflecting impulse purchases. Such claims may be right or they may be wrong, and some are testable. But the point is that the statistical model corresponding to the typical path diagram has a strong subject matter component, and actually is a sort of hybrid, occupying a position somewhere between the typical statistical model and an actual theory about the data.

It is always possible to argue about how the path diagram should look, and it is usually valuable as well. The more subject matter expertise that can be brought to the discussion, the better. Often, the contest between two or more competing pictures will be traceable to unresolved theoretical issues in the field. Will the data at hand allow a formal statistical test to decide between the models? If not, is it possible to design a study that will make such a comparison possible? Thus, the more technical statistical expertise that can be brought to the discussion, the better.

The measurement model — that is, the part relating the latent variables to the observable variables — should not escape scrutiny. The processes it represents are usually

Figure 1.1: The Brand Awareness Study



not the reason the data were collected, but high quality measurement is a key to the success of structural equation modeling.

Continuing with the Brand Awareness example, the model corresponding to Figure 1.1 may be written in scalar form as a system of simultaneous regression-like equations. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned}
Y_{i,1} &= \alpha_1 + \gamma_1 X_{i,1} + \gamma_2 X_{i,2} + \gamma_3 X_{i,3} + \epsilon_{i,1} \\
Y_{i,2} &= \alpha_2 + \beta Y_{i,1} + \gamma_4 X_{i,3} + \epsilon_{i,2} \\
W_{i,1} &= \nu_1 + \lambda_1 X_{i,1} + e_{i,1} \\
W_{i,2} &= \nu_2 + \lambda_2 X_{i,1} + e_{i,2} \\
W_{i,3} &= \nu_3 + \lambda_3 X_{i,2} + e_{i,3} \\
W_{i,4} &= \nu_4 + \lambda_4 X_{i,2} + e_{i,4} \\
W_{i,5} &= \nu_5 + \lambda_5 X_{i,3} + e_{i,5} \\
W_{i,6} &= \nu_6 + \lambda_6 X_{i,3} + e_{i,6} \\
V_{i,1} &= \nu_7 + \lambda_7 Y_{i,1} + e_{i,7} \\
V_{i,2} &= \nu_8 + \lambda_8 Y_{i,1} + e_{i,8} \\
V_{i,3} &= \nu_9 + \lambda_9 Y_{i,2} + e_{i,9} \\
V_{i,4} &= \nu_{10} + \lambda_{10} Y_{i,2} + e_{i,10},
\end{aligned} \tag{1.2}$$

where $E(X_{i,1} = \mu_{x1})$, $E(X_{i,2} = \mu_{x2})$, $E(X_{i,3} = \mu_{x3})$, the expected values of all error terms equal zero, $Var(X_{i,j}) = \phi_{jj}$ for $j = 1, 2, 3$, $Cov(X_{i,j}, X_{i,k}) = \phi_{jk}$, $Var(e_{i,j}) = \omega_j$ for $j = 1, \ldots, 10$, $Var(\epsilon_{i,1}) = \psi_1$, $Var(\epsilon_{i,2}) = \psi_2$, and all the error terms are independent of one another and of the $X_{i,j}$ variables.

If the two measurements of each variable were deemed similar enough, it would be possible to reduce the parameter space quite a bit, for example setting $\nu_1 = \nu_2$, $\lambda_1 = \lambda_2$, and $\omega_1 = \omega_2$. The same kind of thing could be done for the other latent variables. Also, the distributions could be assumed normal, or they could be left unspecified; in practice, those are the two choices.

Setting up the problem in matrix form, we have $p = 3$ latent exogenous variables, $q = 2$ latent endogenous variables, and $k = 10$ observable variables, all of which are endogenous in this example. Using parameter symbols from the scalar version, the equations of the latent variable model are

$$\begin{array}{ccccccccccc}
\mathbf{y}_i & = & \boldsymbol{\alpha} & + & \boldsymbol{\beta} & \mathbf{y}_i & + & \boldsymbol{\Gamma} & \mathbf{x}_i & + & \boldsymbol{\epsilon}_i \\
\begin{pmatrix} Y_{i,1} \\ Y_{i,2} \end{pmatrix} & = & \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} & + & \begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix} & \begin{pmatrix} Y_{i,1} \\ Y_{i,2} \end{pmatrix} & + & \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 \\ 0 & 0 & \gamma_4 \end{pmatrix} & \begin{pmatrix} X_{i,1} \\ X_{i,2} \\ X_{i,3} \end{pmatrix} & + & \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \end{pmatrix}
\end{array}$$

with

$$\boldsymbol{\Phi}_x = cov(\mathbf{x}_i) = \begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{12} & \phi_{22} & \phi_{23} \\ \phi_{13} & \phi_{23} & \phi_{33} \end{pmatrix} \text{ and } \boldsymbol{\Psi} = cov(\boldsymbol{\epsilon}_i) = \begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix}.$$

Collecting $\mathbf{x}_i$ and $\mathbf{y}_i$ into a single vector of "factors,"

$$\mathbf{F}_i = \begin{pmatrix} \mathbf{x}_i \\ \hline \mathbf{y}_i \end{pmatrix} = \begin{pmatrix} X_{i,1} \\ X_{i,2} \\ X_{i,3} \\ Y_{i,1} \\ Y_{i,2} \end{pmatrix}.$$

Finally, the equations of the measurement model are

$$\begin{array}{ccccccc} \mathbf{d}_i & = & \boldsymbol{\nu} & + & \boldsymbol{\Lambda} & \mathbf{F}_i & + & \mathbf{e}_i \\ \begin{pmatrix} W_{i,1} \\ W_{i,2} \\ W_{i,3} \\ W_{i,4} \\ W_{i,5} \\ W_{i,6} \\ V_{i,1} \\ V_{i,2} \\ V_{i,3} \\ V_{i,4} \end{pmatrix} & = & \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \\ \nu_5 \\ \nu_6 \\ \nu_7 \\ \nu_8 \\ \nu_9 \\ \nu_{10} \end{pmatrix} & + & \begin{pmatrix} \lambda_1 & 0 & 0 & 0 & 0 \\ \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & \lambda_3 & 0 & 0 & 0 \\ 0 & \lambda_4 & 0 & 0 & 0 \\ 0 & 0 & \lambda_5 & 0 & 0 \\ 0 & 0 & \lambda_6 & 0 & 0 \\ 0 & 0 & 0 & \lambda_7 & 0 \\ 0 & 0 & 0 & \lambda_8 & 0 \\ 0 & 0 & 0 & 0 & \lambda_9 \\ 0 & 0 & 0 & 0 & \lambda_{10} \end{pmatrix} & \begin{pmatrix} X_{i,1} \\ X_{i,2} \\ X_{i,3} \\ Y_{i,1} \\ Y_{i,2} \end{pmatrix} & + & \begin{pmatrix} e_{i,1} \\ e_{i,2} \\ e_{i,3} \\ e_{i,4} \\ e_{i,5} \\ e_{i,6} \\ e_{i,7} \\ e_{i,8} \\ e_{i,9} \\ e_{i,10} \end{pmatrix} \end{array}$$

with

$$\boldsymbol{\Omega} = cov(\mathbf{e}_i) = \begin{pmatrix} \omega_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \omega_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \omega_7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_{10} \end{pmatrix}$$

Given a verbal description of a data set, the student should be able to write down a path diagram, and translate freely between the path diagram, the model in scalar form and the model in matrix form. Three three ways of expressing the model are equivalent, and some software[2] will allow a model to be specified using only a built-in drawing program. This can be appealing to users who don't like equations and Greek letters, but for larger models the process can be very tedious.

## 1.3 Review of identifiability

The general two-stage model (1.1) of Section 1.2 is very general indeed — so much so, that its parameters are seldom identifiable without additional restrictions. Choosing these

[2]The ones I know of are Amos and JMP.

restrictions wisely is an essential part of structural equation modeling. In fact, it turns out that almost everything that makes structural equation modeling distinct from other large-sample statistical methods can be traced to issue of parameter identifiability. For the convenience of readers who are starting with Chapter 1, this section collects material on identifiability from Chapter 0. Readers of Chapter 0 are also encouraged to look it over. The presentation is intended to be terse. For more detail, please see Chapter 0.

**Definition 0.5** (Page 59) Suppose a statistical model implies $\mathbf{d} \sim P_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \Theta$. If no two points in $\Theta$ yield the same probability distribution, then the parameter $\boldsymbol{\theta}$ is said to be *identifiable*. On the other hand, if there exist $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ in $\Theta$ with $P_{\boldsymbol{\theta}_1} = P_{\boldsymbol{\theta}_2}$, the parameter $\boldsymbol{\theta}$ is *not identifiable.*

**Theorem 0.1** (Page 59) If the parameter vector is not identifiable, consistent estimation for all points in the parameter space is impossible.

**Definition 0.6** (Page 60) The parameter is said to be *identifiable* at a point $\boldsymbol{\theta}_0$ if no other point in $\Theta$ yields the same probability distribution as $\boldsymbol{\theta}_0$.

**Definition 0.7** (Page 60) The parameter is said to be *locally identifiable* at a point $\boldsymbol{\theta}_0$ if there is a neighbourhood of points surrounding $\boldsymbol{\theta}_0$, none of which yields the same probability distribution as $\boldsymbol{\theta}_0$.

**Definition 0.8** (Page 60) Let $g(\boldsymbol{\theta})$ be a function of the parameter vector. If $g(\boldsymbol{\theta}_0) \neq g(\boldsymbol{\theta})$ implies $P_{\boldsymbol{\theta}_0} \neq P_{\boldsymbol{\theta}}$ for all $\boldsymbol{\theta} \in \Theta$, then the function $g(\boldsymbol{\theta})$ is said to be identifiable at the point $\boldsymbol{\theta}_0$.

**Theorem 0.2** (Page 61) Let

$$\begin{aligned} y_1 &= f_1(x_1, \ldots, x_p) \\ y_2 &= f_2(x_1, \ldots, x_p) \\ \vdots & \qquad \vdots \\ y_q &= f_q(x_1, \ldots, x_p), \end{aligned}$$

If the functions $f_1, \ldots, f_q$ are analytic (posessing a Taylor expansion) and $p > q$, the set of points $(x_1, \ldots, x_p)$ where the system of equations has a unique solution occupies at most a set of volume zero in $\mathbb{R}^p$.

*Moment structure equations* give moments of the distribution of the observable data in terms of model parameters. In this course, moments are limited to expected values, variances and covariances. If it is possible to solve uniquely for the parameter vector in terms of the these quantities, then the parameter vector is identifiable. Even when a multivariate normal distribution is not assumed, in practice "identifiable" means identifiable from the moments — usually the variances and covariances.

**Rule 1** (The Parameter Count Rule, page 61) Suppose identifiability is to be decided based on a set of moment structure equations. If there are more parameters than equations, the parameter vector is identifiable on at most a set of volume zero in the parameter space.

## 1.4 Models: Original and Surrogate

### 1.4.1 Overview

It is taken for granted that even the best scientific models are not "true" in any ultimate sense. At best, they are approximations of how nature really works. And this is even more true of statistical models. As Box and Draper (1987, p. 424) put it, "Essentially all models are wrong, but some are useful." [11] In structural equation modeling, the models used in practice are usually not even the approximate versions that the scientist or statistician has in mind. Instead, they are re-parameterized versions of the intended models. This explains some features that may seem odd at first.

Figure 1.2: A sequence of re-parameterizations

$$\text{Truth} \approx \text{Original Model} \rightarrow \text{Surrogate Model 1} \rightarrow \text{Surrogate Model 2} \rightarrow \ldots$$

Figure 1.2 is a picture of the process[3]. Underlying everything is the true state of nature, the real process that gave rise to the observable data in our possession. We can scarcely even imagine what it is, but undoubtedly it's non-linear, and involves a great many unmeasured variables. So we start with a model based on the general two-stage model (1.1) of Section 1.2. It is not the truth and we know it's not the truth, but maybe it's not too bad. It's basically a collection of regression equations, complete with intercepts. Based on the usefulness of ordinary multiple regression, there is reason to hope it roughly approximates the truth in a useful way, at least within the range of the observed data.

As primitive as the original model may be compared to the real truth, its parameters are still not identifiable. So we re-parameterize, producing a new model whose parameters are *functions* of the parameters of the original model. Such a model will be called a *surrogate* model because it stands for the original model, and tries to do the job of the original model. Like a surrogate mother, it may not be as good as a the real thing, but it will have to do.

As indicated in Figure 1.2, re-parameterization may happen in more than one step. For the classical structural equation models presented in this book, the first re-parameterization results in a *centered* surrogate model with no intercepts, and all expected values equal to zero. The model equations may look a bit strange at first glance, but it is much more

[3]Thanks to Michael Li for this way of expressing the idea.

convenient if we even don't even have to look at symbols for vectors of parameters that we can't estimate uniquely anyway.

Typically, the parameters of the centered surrogate model are still not identifiable, and there is another re-parameterization, leading to a second level surrogate model. The process can continue. At each step, the parameter vector of the new model is a function of of the parameters of the preceding model, and typically the function is not one-to-one. Otherwise, identifiability would not change. At each stage, the dimension of the new parameter space is less, so the re-parameterization represents a restriction, or collapsing of the original parameter space. The end result is a model whose parameters are identifiable functions of the original parameter vector. The goal is for those functions to be as informative as possible about the parameters of the original model.

Two features of the original model deserve special mention. The first is that usually, the original model is already a restricted version of Model (1.1), even before it is re-parameterized to produce a surrogate model. The restrictions in question arise from substantive modeling considerations rather than from a search for identifiability. So, in the Brand Awareness example of Section 1.2, the parameter matrices have many elements fixed at zero. These represent theoretical assertions about consumer psychology. They may be helpful in making the remaining free parameters identifiable, but that is not their justification.

A second notable feature of the original model is that expected values are non-zero in general, and all the equations are regression-like equations with intercepts, and with slopes that do not necessarily equal one. Any deviation from this standard needs to be justified on substantive grounds, not on grounds of simplicity or convenience. Otherwise, it's a surrogate model and not an original model. The distinction is important, because most structural equation models used in practice are surrogate models, and a good way to understand them is to trace the connection between their parameters and the parameters of the original models from which they are are derived.

Consider a simple additive model for measurement error, like (28) on page 40:

$$W = X + e.$$

Immediately it is revealed as a surrogate model, because there is no intercept and the slope is set to one – a choice that would be hard to justify on modeling grounds most of the time. For example, $X$ might be actual calories consumed during the past week, and $W$ might be number of reported calories based on answers to a questionnaire. Undoubtedly, the true relationship between these variables is non-linear. In an original (though not exactly true) model, the relationship would be approximated by

$$W = \nu + \lambda X + e.$$

With this example in mind, it is clear that most of the models given in Chapter 0 (and *all* the models in Chapter 0 with identifiable parameters) are actually surrogate models. This might be a bit unsettling because you did not realize that you were being tricked, or it might be reassuring because some models that struck you as unrealistic may actually be better than they seem.

### 1.4.2 The centered surrogate model

The first stage of re-parameterization may be done in full generality. The argument begins with a demonstration that the means and intercepts of the original model are not identifiable. Please bear in mind that as a practical consideration, "identifiable" means identifiable from the moments – the expected values and variance-covariance matrix of the observable data.

Starting with the latent variable part of the two-stage original model (1.1), it is helpful to write the endogenous variables solely as functions of the exogenous variables, and not of each other.

$$\begin{aligned}
& \mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\
\Leftrightarrow \quad & \mathbf{y}_i - \boldsymbol{\beta}\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\
\Leftrightarrow \quad & \mathbf{I}\mathbf{y}_i - \boldsymbol{\beta}\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\
\Leftrightarrow \quad & (\mathbf{I} - \boldsymbol{\beta})\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\
\Leftrightarrow \quad & (\mathbf{I} - \boldsymbol{\beta})^{-1}(\mathbf{I} - \boldsymbol{\beta})\mathbf{y}_i = (\mathbf{I} - \boldsymbol{\beta})^{-1}(\boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}) \\
\Leftrightarrow \quad & \mathbf{y}_i = (\mathbf{I} - \boldsymbol{\beta})^{-1}(\boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i) \qquad (1.3)
\end{aligned}$$

The preceding calculation assumes that the matrix $\mathbf{I} - \boldsymbol{\beta}$ has an inverse. Surprisingly, the existence of $(\mathbf{I} - \boldsymbol{\beta})^{-1}$ is guaranteed by the model. The proof hinges on the specifications that $\mathbf{x}_i$ and $\boldsymbol{\epsilon}_i$ are independent, and that $\boldsymbol{\Psi} = cov(\boldsymbol{\epsilon}_i)$ is positive definite.

**Theorem 1.1** *Model (1.1) implies the existence of* $(\mathbf{I} - \boldsymbol{\beta})^{-1}$.

**Proof** $\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i$ yields $(\mathbf{I} - \boldsymbol{\beta})\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i$. Suppose $(\mathbf{I} - \boldsymbol{\beta})^{-1}$ does not exist. Then the rows of $\mathbf{I} - \boldsymbol{\beta}$ are linearly dependent, and there is a $q \times 1$ non-zero vector of constants $\mathbf{a}$ with $\mathbf{a}^\top(\mathbf{I} - \boldsymbol{\beta}) = 0$. So,

$$\begin{aligned}
0 &= \mathbf{a}^\top(\mathbf{I} - \boldsymbol{\beta})\mathbf{y}_i = \mathbf{a}^\top\boldsymbol{\alpha} + \mathbf{a}^\top\boldsymbol{\Gamma}\mathbf{x}_i + \mathbf{a}^\top\boldsymbol{\epsilon}_i \\
\Rightarrow Var(0) &= Var(\mathbf{a}^\top\boldsymbol{\Gamma}\mathbf{x}_i) + Var(\mathbf{a}^\top\boldsymbol{\epsilon}_i) \\
\Rightarrow 0 &= \mathbf{a}^\top\boldsymbol{\Gamma}\boldsymbol{\Phi}_x\boldsymbol{\Gamma}^\top\mathbf{a} + \mathbf{a}^\top\boldsymbol{\Psi}\mathbf{a}.
\end{aligned}$$

But the quantity on the right side is strictly positive, because while $\boldsymbol{\Gamma}\boldsymbol{\Phi}_x\boldsymbol{\Gamma}^\top = cov(\boldsymbol{\Gamma}\mathbf{x}_i)$ is only guaranteed to be non-negative definite, $\boldsymbol{\Psi}$ is strictly positive definite according to the model. Thus, the assumption that $\mathbf{I} - \boldsymbol{\beta}$ is singular leads to a contradiction. This shows that $(\mathbf{I} - \boldsymbol{\beta})^{-1}$ must exist if the model holds. ■

Sometimes, the surface defined by $|\mathbf{I} - \boldsymbol{\beta}| = 0$ is interior to the parameter space, and yet cannot belong to the parameter space because of the other model specifications. Thus it forms an unexpected hole in the parameter space. The pinwheel Model () on page whatever provides an example.

Now that the existence of $(\mathbf{I} - \boldsymbol{\beta})^{-1}$ is established, Expression (1.3) may be used to calculate expected values, variances and covariances. Expressing the results of routine calculations as partitioned matrices,

$$
\begin{aligned}
\boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F &= E(\mathbf{F}_i) = \left(\frac{E(\mathbf{x}_i)}{E(\mathbf{y}_i)}\right) = \left(\frac{\boldsymbol{\mu}_x}{(\mathbf{I}-\boldsymbol{\beta})^{-1}(\boldsymbol{\alpha}+\boldsymbol{\Gamma}\boldsymbol{\mu}_x)}\right) \\
\boldsymbol{\mu} &= E(\mathbf{d}_i) = \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F \\
\boldsymbol{\Phi} &= cov(\mathbf{F}_i) = \left(\begin{array}{c|c} cov(\mathbf{x}_i) & cov(\mathbf{x}_i,\mathbf{y}_i) \\ \hline & cov(\mathbf{y}_i) \end{array}\right) = \left(\begin{array}{c|c} \boldsymbol{\Phi}_x & \boldsymbol{\Phi}_x\boldsymbol{\Gamma}^\top(\mathbf{I}-\boldsymbol{\beta})^{-1\,T} \\ \hline & (\mathbf{I}-\boldsymbol{\beta})^{-1}\left(\boldsymbol{\Gamma}\boldsymbol{\Phi}_x\boldsymbol{\Gamma}^\top+\boldsymbol{\Psi}\right)(\mathbf{I}-\boldsymbol{\beta})^{-1\,T} \end{array}\right) \\
\boldsymbol{\Sigma} &= cov(\mathbf{d}_i) = \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^\top + \boldsymbol{\Omega}
\end{aligned}
\tag{1.4}
$$

The parameter matrices may be divided into three categories: those appearing only in $\boldsymbol{\mu} = E(\mathbf{d}_i)$, those appearing only in $\boldsymbol{\Sigma} = cov(\mathbf{d}_i)$, and those appearing in both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$.

| | |
|---|---|
| Appearing only in $\boldsymbol{\mu}$ | $\boldsymbol{\mu}_x, \boldsymbol{\alpha}, \boldsymbol{\nu}$ |
| Appearing only in $\boldsymbol{\Sigma}$ | $\boldsymbol{\Phi}_x, \boldsymbol{\Psi}, \boldsymbol{\Omega}$ |
| Appearing in both | $\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Lambda}$ |

Clearly, the parameters appearing only in $\boldsymbol{\mu}$ must be identified from the $k$ mean structure equations or not at all. Even assuming the best case scenario in which $\boldsymbol{\beta}, \boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ can be identified from $\boldsymbol{\Sigma}$ and thus may be considered known, this requires the solution of $k$ equations in $k + p + q$ unknowns. Since the equations are linear, there is no need to invoke the parameter count rule[4]. For every fixed set of $(\boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Lambda})$ values, infinitely many sets $(\boldsymbol{\mu}_x, \boldsymbol{\alpha}, \boldsymbol{\nu})$ yield the same vector of expected values $\boldsymbol{\mu}$. Thus, the means and intercepts in the model are not identifiable.

Not much is lost, because usually the matrices $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ are of primary interest, and these (or useful functions of them) may potentially be recovered from $\boldsymbol{\Sigma}$. So the standard solution is to re-parameterize, replacing the parameter set $(\boldsymbol{\Phi}_x, \boldsymbol{\Psi}, \boldsymbol{\Omega}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{\mu}_x, \boldsymbol{\alpha}, \boldsymbol{\nu})$ with $(\boldsymbol{\Phi}_x, \boldsymbol{\Psi}, \boldsymbol{\Omega}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{\kappa})$, where $\boldsymbol{\kappa} = \boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F$. Then $\boldsymbol{\kappa}$ is treated as a nuisance parameter to be estimated with the vector of sample means where technically necessary, but otherwise ignored.

A useful way to express the re-parameterization is to re-write the equations of Model (1.1), centering all the random vectors. Starting with the latent variable part,

$$
\begin{aligned}
\mathbf{y}_i &= (\mathbf{I}-\boldsymbol{\beta})^{-1}(\boldsymbol{\alpha}+\boldsymbol{\Gamma}\mathbf{x}_i+\boldsymbol{\epsilon}_i) \\
&= (\mathbf{I}-\boldsymbol{\beta})^{-1}(\boldsymbol{\alpha}+\boldsymbol{\Gamma}\mathbf{x}_i-\boldsymbol{\Gamma}\boldsymbol{\mu}_x+\boldsymbol{\Gamma}\boldsymbol{\mu}_x+\boldsymbol{\epsilon}_i) \\
\Leftrightarrow\quad \mathbf{y}_i-(\mathbf{I}-\boldsymbol{\beta})^{-1}(\boldsymbol{\alpha}+\boldsymbol{\Gamma}\boldsymbol{\mu}_x) &= (\mathbf{I}-\boldsymbol{\beta})^{-1}(\boldsymbol{\Gamma}(\mathbf{x}_i-\boldsymbol{\mu}_x)+\boldsymbol{\epsilon}_i) \\
\Leftrightarrow\quad \overset{c}{\mathbf{y}}_i &= (\mathbf{I}-\boldsymbol{\beta})^{-1}(\boldsymbol{\Gamma}\overset{c}{\mathbf{x}}_i+\boldsymbol{\epsilon}_i) \\
\Leftrightarrow\quad (\mathbf{I}-\boldsymbol{\beta})\overset{c}{\mathbf{y}}_i &= \boldsymbol{\Gamma}\overset{c}{\mathbf{x}}_i+\boldsymbol{\epsilon}_i \\
\Leftrightarrow\quad \overset{c}{\mathbf{y}}_i &= \boldsymbol{\beta}\overset{c}{\mathbf{y}}_i+\boldsymbol{\Gamma}\overset{c}{\mathbf{x}}_i+\boldsymbol{\epsilon}_i,
\end{aligned}
$$

[4] A system of linear equations with more unknowns than equations has either infinitely many solutions or none at all. The option of no solutions is ruled out because the pair $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is actually the image of one particular set of parameter matrices in the parameter space. More details about mappings between the parameter space and the moment space are given in Chapter 6.

where putting a $c$ above a random vector means it has been centered by subtracting off its expected value. Automatically we have

$$\overset{c}{\mathbf{F}}_i = \mathbf{F}_i - \boldsymbol{\mu}_F = \begin{pmatrix} \overset{c}{\mathbf{x}}_i \\ \hline \overset{c}{\mathbf{y}}_i \end{pmatrix}.$$

For the measurement part of the model,

$$\begin{array}{rrcl} & \mathbf{d}_i & = & \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i \\ & & = & \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i - \boldsymbol{\Lambda}\boldsymbol{\mu}_F + \boldsymbol{\Lambda}\boldsymbol{\mu}_F + \mathbf{e}_i \\ \Leftrightarrow & \mathbf{d}_i - (\boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F) & = & \boldsymbol{\Lambda}(\mathbf{F}_i - \boldsymbol{\mu}_F) + \mathbf{e}_i \\ \Leftrightarrow & \overset{c}{\mathbf{d}}_i & = & \boldsymbol{\Lambda}\overset{c}{\mathbf{F}}_i + \mathbf{e}_i. \end{array}$$

Thus, a centered version of Model (1.1) is 100% equivalent to the original. A *surrogate* for Model (1.1) is obtained by simply dropping the letter $c$ over the random vectors, and writing

$$\begin{array}{rcl} \mathbf{y}_i & = & \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\ \mathbf{F}_i & = & \begin{pmatrix} \mathbf{x}_i \\ \hline \mathbf{y}_i \end{pmatrix} \\ \mathbf{d}_i & = & \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \end{array} \tag{1.5}$$

where $E(\mathbf{x}_i) = 0$, and all other specifications are as in Model (1.1). This will be called the *Centered Surrogate Model*. It is a good substitute for the original because

- It hides the nuisance parameters $\boldsymbol{\mu}_x$, $\boldsymbol{\alpha}$ and $\boldsymbol{\nu}$, which can't be identified anyway, and are essentially discarded by a re-parameterization.
- The remaining parameter matrices are identical to those of the original model.
- The covariance matrix $\boldsymbol{\Sigma}$ of the observable data (given in Expression 1.4) is identical to that of the original model.
- Special cases of $\boldsymbol{\Sigma}$ that are used in applications easier to calculate.

It must be emphasized that (1.5) is not a realistic model for almost any actual data set, because most variables don't have zero expected value[5]. Rather, it's a substitute for a re-parameterized version of the original Model (1.1), one that's more convenient to work with. This explains why structural equation models are usually written in centered form, with zero means and no intercepts, and why some structural equation modeling software does not even allow for models with means and intercepts.

[5]Some authors suggest that the observable data have been centered by subtracting off *sample* means, so that they do have expected value zero. That would explain why $\boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F = 0$, but not why $\boldsymbol{\mu}_F$ is necessarily equal to zero.

### 1.4.3 An additional re-parameterization

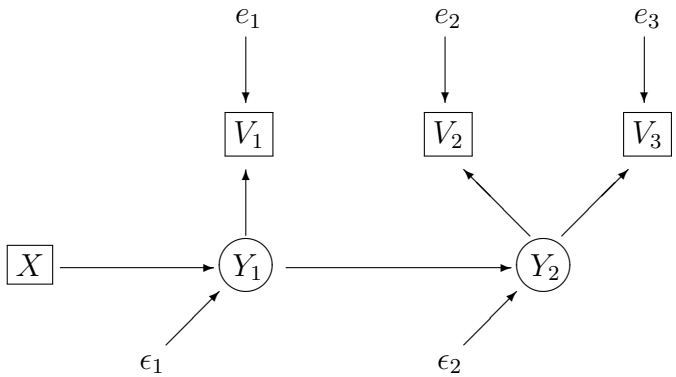
In general, the parameters of the centered surrogate model are still not identifiable. In most cases, even after restricting the parameters based on modeling considerations, further technical restrictions are necessary to obtain a model whose parameters are identifiable. Like centering, these restrictions should be viewed as re-parameterizations, and the models that result should be viewed as surrogates for the original model. But unlike centering, which does not affect the parameters appearing in the covariance matrix, the second level of re-parameterization affects the *meaning* of the remaining parameters. General principles will be developed in later chapters, but here is a simple example to illustrate the idea.

**Example 1.4.1** *Blood Pressure*

Patients with high blood pressure are randomly assigned to different dosages of a blood pressure medication. There are many different dosages, so dosage may be treated as a continuous variable. Because the exact dosage is known, this exogenous variable is observed without error. After one month of taking the medication, the level of the drug in the patient's bloodstream is measured once (with error, of course), by an independent lab. Then, two measurements of the patient's blood pressure are taken in the doctor's office. The measurements are taken on different days and by different technicians, but with exacly the same equipment and following exactly the same measurement protocol. Thus, the two blood pressure readings are thought to be equivalent as well as having independent measurement errors.

Figure 1.3 shows a path diagram of the model, with $X$ representing drug dosage, $Y_1$ representing true blood level of the drug, and $Y_2$ representing the patient's average resting blood pressure.

Figure 1.3: Blood pressure path model




The original model for this problem may be written in scalar form as follows. Inde-

pendently for $i = 1, \ldots, n$,

$$\begin{aligned} Y_{i,1} &= \alpha_1 + \gamma X_i + \epsilon_{i,1} \\ Y_{i,2} &= \alpha_2 + \beta Y_{i,1} + \epsilon_{i,2} \\ V_{i,1} &= \nu_1 + \lambda_1 Y_{i,1} + e_{i,1} \\ V_{i,2} &= \nu_2 + \lambda_2 Y_{i,2} + e_{i,2} \\ V_{i,3} &= \nu_2 + \lambda_2 Y_{i,2} + e_{i,3}, \end{aligned} \tag{1.6}$$

where $E(X_i) = \mu_x$, $Var(X_i) = \phi$, all error terms are independent with expected values equal to zero, $Var(\epsilon_{i,1}) = \psi_1$, $Var(\epsilon_{i,2}) = \psi_2$, $Var(e_{i,1}) = \omega_1$, and $Var(e_{i,2}) = Var(e_{i,3}) = \omega_2$. The equal intercepts, slopes and intercepts for $V_2$ and $V_3$ are modeling restrictions, based on the belief that $V_2$ and $V_3$ really are equivalent measurements.

Again, this is the original model. In a typical application, a surrogate model would be presented, both to the reader and to the software. It would be in centered form, with the coefficients $\lambda_1$ and $\lambda_2$ both set equal to one. There might be a brief reference to "setting the scales" of the latent variables[6]. Here is a more detailed account of what is going on.

How does the surrogte model arise from the original model? The first step is to re-parameterize by a change of variables in which each variable is transformed by subtracting off its expected value, and then any notational evidence if the transformation is suppressed. The result is a centered surrogate model like (1.5). Before further re-parameterization, let us verify that the parameters of the centered model are not identifiable. It passes the test of the parameter count rule, because the covariance matrix contains ten parameters and has ten unique elements. So there are ten covariance structure equations in ten unknowns.

The covariance matrix $\boldsymbol{\Sigma} = [\sigma_{ij}]$ of the observable variables $\mathbf{d}_i = (X_i, V_{i,1}, V_{i,2}, V_{i,3})^\top$ is

$$\begin{pmatrix} \phi & \gamma\lambda_1\phi & \beta\gamma\lambda_2\phi & \beta\gamma\lambda_2\phi \\ & (\gamma^2\phi + \psi_1)\lambda_1^2 + \omega_1 & (\gamma^2\phi + \psi_1)\beta\lambda_1\lambda_2 & (\gamma^2\phi + \psi_1)\beta\lambda_1\lambda_2 \\ & & (\beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2)\lambda_2^2 + \omega_2 & (\beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2)\lambda_2^2 \\ & & & (\beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2)\lambda_2^2 + \omega_2 \end{pmatrix}. \tag{1.7}$$

The model imposes three three equality constraints on the covariance matrix: $\sigma_{13} = \sigma_{14}$, $\sigma_{23} = \sigma_{24}$ and $\sigma_{33} = \sigma_{34}$. This effectively reduces the number of covariance structure equations by three, so that to show identifiability it would be necessary to solve seven equations in ten unknowns[7]. By the parameter count rule, a unique solution is impossible except possibly on a set of volume zero in the parameter space. So the parameter vector is not identifiable.

[6] See for example Bollen, get reference from language paper.

[7] This idea is a bit subtle. The $\sigma_{ij}$ quantities should be viewed as images of a *single, fixed* point $\boldsymbol{\theta}_0$ in the parameter space. So if the model implies $\sigma_{13} = \sigma_{14}$ because they both equal $\beta\gamma\lambda_2\phi$, it means that $\sigma_{13}$ and $\sigma_{14}$ both represent the same real number. At this point, parameter symbols like $\beta$ and $\gamma$ represent fixed constants too, because they are elements of $\boldsymbol{\theta}_0$. But then when the attempt is made to recover $\boldsymbol{\theta}_0$ from $\boldsymbol{\Sigma}(\boldsymbol{\theta}_0)$ by solving equations, parameter symbols like $\beta$ and $\gamma$ are treated as variables, while the $\sigma_{ij}$ quantities remain fixed constants. Chapter 6 discusses mappings back and forth between the parameter space and the moment space.

If this argument is not entirely convincing, the table below gives a numerical example of two different parameter vectors (with $\gamma$, $\beta$, $\lambda_1$ and $\lambda_2$ all non-zero) that yield the same covariance matrix.

| | $\gamma$ | $\beta$ | $\lambda_1$ | $\lambda_2$ | $\psi_1$ | $\psi_2$ | $\phi$ | $\omega_1$ | $\omega_2$ |
|---|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{\theta}_1$ | 2 | 4 | 1 | 1 | 4 | 16 | 1 | 1 | 1 |
| $\boldsymbol{\theta}_2$ | 1 | 2 | 2 | 4 | 1 | 1 | 1 | 1 | 1 |

Both parameter vectors yield the covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 2 & 8 & 8 \\ 2 & 9 & 32 & 32 \\ 8 & 32 & 145 & 144 \\ 8 & 32 & 144 & 145 \end{pmatrix}.$$

By Definition 0.5, the parameter vector is not identifiable.

The next step is to re-examine the model equations in (surrogate) centered form,

$$\begin{aligned} Y_{i,1} &= \gamma X_i + \epsilon_{i,1} \\ Y_{i,2} &= \beta Y_{i,1} + \epsilon_{i,2} \\ V_{i,1} &= \lambda_1 Y_{i,1} + e_{i,1} \\ V_{i,2} &= \lambda_2 Y_{i,2} + e_{i,2} \\ V_{i,3} &= \lambda_2 Y_{i,2} + e_{i,3} \end{aligned} \tag{1.8}$$

and carry out the standard re-parameterization that yields $\lambda_1 = \lambda_2 = 1$, purchasing identifiability. Expressing the re-parameterization as a *change of variables* will make it easier to trace the connection between the parameters of the original model and those of the re-parameterized model. First note that on modeling grounds, we are sure that $\lambda_1 > 0$ and $\lambda_2 > 0$.

Let $Y'_{i,1} = \lambda_1 Y_{i,1}$ and $Y'_{i,2} = \lambda_2 Y_{i,2}$. The primes just denote a new (transformed) random variable. Then from the first equation of (1.8),

$$\begin{aligned} Y'_{i,1} &= (\lambda_1\gamma)X_i + \lambda_1\epsilon_{i,1} \\ &= \gamma' X_i + \epsilon'_{i,1}. \end{aligned}$$

From the second equation of (1.8),

$$\begin{aligned} Y'_{i,2} &= \lambda_2\beta Y_{i,1} + \lambda_2\epsilon_{i,2} \\ &= \lambda_2\beta\frac{\lambda_1}{\lambda_1}Y_{i,1} + \lambda_2\epsilon_{i,2} \\ &= \left(\frac{\lambda_2\beta}{\lambda_1}\right)Y'_{i,1} + \lambda_2\epsilon_{i,2} \\ &= \beta' Y'_{i,1} + \epsilon'_{i,2}. \end{aligned}$$

Using $Y'_{i,1} = \lambda_1 Y_{i,1}$ and $Y'_{i,2} = \lambda_2 Y_{i,2}$, and putting it all together, the equations of the second level surrogate model are

$$\begin{aligned} Y'_{i,1} &= \gamma' X_i + \epsilon'_{i,1} \\ Y'_{i,2} &= \beta' Y'_{i,1} + \epsilon'_{i,2} \\ V_{i,1} &= Y'_{i,1} + e_{i,1} \\ V_{i,2} &= Y'_{i,2} + e_{i,2} \\ V_{i,3} &= Y'_{i,2} + e_{i,3}, \end{aligned} \tag{1.9}$$

where

$$\begin{aligned} \gamma' &= \lambda_1 \gamma \\ \psi'_1 &= Var(\epsilon'_{i,1}) = \lambda_1^2 \psi_1 \\ \beta' &= \frac{\lambda_2 \beta}{\lambda_1} \\ \psi'_2 &= Var(\epsilon'_{i,2}) = \lambda_2^2 \psi_2 \\ \lambda'_1 &= 1 \\ \lambda'_2 &= 1. \end{aligned} \tag{1.10}$$

The only parameters of the original model that are unaffected are $\omega_1$ and $\omega_2$.

The primes are now suppressed, resulting in a model that looks like (1.8) with $\lambda_1 = \lambda_2 = 1$. The parameters of this model have the same names as some parameters of the original model, but actually they are *functions* of those parameters and other parameters ($\lambda_1$ and $\lambda_2$, in this case) that have been made invisible by the re-parameterization. In terms of the new parameters, the covariance matrix $\mathbf{\Sigma}$ is

$$\begin{pmatrix} \phi & \gamma\phi & \beta\gamma\phi & \beta\gamma\phi \\ \gamma\phi & \gamma^2\phi + \omega_1 + \psi_1 & (\gamma^2\phi + \psi_1)\beta & (\gamma^2\phi + \psi_1)\beta \\ \beta\gamma\phi & (\gamma^2\phi + \psi_1)\beta & \beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 & \beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2 \\ \beta\gamma\phi & (\gamma^2\phi + \psi_1)\beta & \beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2 & \beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 \end{pmatrix}. \tag{1.11}$$

It is easy to solve for the new parameters in terms of the variances and covariances $\sigma_{ij}$, showing that the functions of the original parameters given in (1.7) are identifiable.

Moreover, because the covariance matrix (1.11) is just the covariance matrix (1.7) written in a different notation, the second level surrogate model (1.9) imposes the same constraints on the covariance matrix that the original and centered surrogate models do. These include the equality constraints $\sigma_{13} = \sigma_{14}$, $\sigma_{23} = \sigma_{24}$ and $\sigma_{33} = \sigma_{34}$. As described in Chapter 7, treating these constraints as a null hypothesis provides a way of testing model correctness. Rejection of that null hypothesis would cast doubt on the original model.

The *meanings* of the parameters of the surrogate model are clear from the identities in (1.10). The crucial parameters $\gamma$ and $\beta$ are multiplied by constants that are not just unknown, they are *un-knowable* except for being positive. Thus, it will be possible to make reasonable inference about whether these regression coefficients are positive, negative or

zero. But parameter estimation as such is a meaningless exercise. It is useful only as an intermediate step in the construction of hypothesis tests.

Actually, not much is lost here. It may be impossible to estimate the the parameters of interest[8], but recall Figure 1.2. The straight-line relationships of the original model are at best approximations of the non-linear functions that occur in nature. So one may hope that conclusions about the signs of regression coefficients will apply to whether the true relationship is monotone increasing or monotone decreasing. By the way, this hope is all you ever have with linear regression, as well.

So on the surface, setting $\lambda_1 = \lambda_2 = 1$ looks like either an arbitrary restriction of the parameter space, or a measurement model that is very difficult to defend. But in fact it is a very good re-parameterization, resulting in a surrogate model whose parameters are not only identifiable, but also reflect what can be known about the parameters of the original model. It is very helpful to express the re-parameterization in terms of a change of variables, because that reveals how the apparent suppression of $\lambda_1$ and $\lambda_2$ caused them to appear in the remaining model parameters. This was not at all obvious.

Fortunately, re-parameterizations like this usually do not need to be carried out explicitly. It is common practice to write the model in centered form from the beginning, set one factor loading[9] for each latent variable equal to one, and then check parameter identifiability. This is fine, provided that the process is understood as a re-parameterization with cascading effects on the coefficients linking the latent variables to one another and to the other observable variables in the model.

As alternative to setting factor loadings equal to one, the centered surrogate model may be re-parameterized so that the variances of transformed latent variables are equal to one. That is, if $F_j$ is a latent variable with variance $\phi_{jj}$, the change of variables is $F_j' = \sqrt{\phi_{jj}} F_j$. This device has advantages and disadvantages. Further discussion is deferred until Chapter 3, which focuses upon the measurement model that links latent to observable variables.

### 1.4.4 The blood pressure example with Sage

Sage is an open source symbolic mathematics software package. Use of such software can greatly ease the computational burden of structural equation modeling. This section assumes the introduction to Sage in Appendix B. Like all the Sage material, it may be skipped without loss of continuity. Since this is the first example in the textbook proper, it contains quite a bit of extra detail.

Writing the equations of the centered surrogate model in matrix form, the latent

[8] One might hope that in a different re-parameterization, $\gamma$ and $\beta$ might appear unaltered as parameters in the new model. But the numerical example shows that $\gamma$ and $\beta$ are not identifiable, and hence by Theorem 0.1, consistent estimation of them is out of the question.

[9] This terminology anticipates Chapters 2 and 3. A factor loading is a coefficient linking a latent variable to an observable variable.

variable part is

$$\begin{array}{ccccccccc} \mathbf{y}_i & = & \boldsymbol{\beta} & \mathbf{y}_i & + & \boldsymbol{\Gamma} & \mathbf{x}_i & + & \boldsymbol{\epsilon}_i \\ \begin{pmatrix} Y_{i,1} \\ Y_{i,2} \end{pmatrix} & = & \begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix} & \begin{pmatrix} Y_{i,1} \\ Y_{i,2} \end{pmatrix} & + & \begin{pmatrix} \gamma \\ 0 \end{pmatrix} & \begin{pmatrix} X_i \end{pmatrix} & + & \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \end{pmatrix}, \end{array}$$

and the measurement part of the model is

$$\begin{array}{ccccccc} \mathbf{d}_i & = & \boldsymbol{\Lambda} & \mathbf{F}_i & + & \mathbf{e}_i \\ \begin{pmatrix} X_i \\ V_{i,1} \\ V_{i,2} \\ V_{i,3} \end{pmatrix} & = & \begin{pmatrix} 1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \\ 0 & 0 & \lambda_2 \end{pmatrix} & \begin{pmatrix} X_i \\ Y_{i,1} \\ Y_{i,2} \\ X_{i,3} \end{pmatrix} & + & \begin{pmatrix} e_{i,1} \\ e_{i,2} \\ e_{i,3} \\ e_{i,4} \end{pmatrix}. \end{array}$$

For the measurement model equations to make sense, it is necessary for the distribution of $e_{i,1}$ to be degenerate at zero; that is, $Pr\{e_{i,1} = 0\} = 1$. This will be accomplished by setting $Var(e_{i,1}) = 0$.

The covariance matrix $\boldsymbol{\Sigma} = cov(\mathbf{d}_i)$ is the same under the original model and the centered surrugate model. To calculate it, first download the `sem` package.

```
sem = 'http://www.utstat.toronto.edu/ brunner/openSEM/sage/sem.sage'
load(sem)
```

evaluate

Then set up the parameter matrices $\boldsymbol{\Phi}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\beta}$, $\boldsymbol{\Psi}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$. Because these matrices contain so many zeros, the `ZeroMatrix` function is used quite a bit to create symbolic matrices that initially contain nothing but zeros. Then, non-zero elements are assigned using `var` statements. First comes $\boldsymbol{\Phi}$, which is $1 \times 1$.

```
# Set up matrices: p = 1, q = 2, k = 4
# Remember, matrix indices start with zero
PHIx = ZeroMatrix(1,1); PHIx[0,0] = var('phi'); show(PHIx)
```

evaluate

$$\begin{pmatrix} \phi \end{pmatrix}$$

The matrix $\boldsymbol{\Gamma}$ is $2 \times 1$.

```
GAMMA = ZeroMatrix(2,1); GAMMA[0,0] = var('gamma'); show(GAMMA)
```

evaluate

$$\begin{pmatrix} \gamma \\ 0 \end{pmatrix}$$

The matrix $\boldsymbol{\beta}$ is $2 \times 2$.

```
BETA = ZeroMatrix(2,2); BETA[1,0] = var('beta'); show(BETA)
```

evaluate

$$\begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix}$$

The $2 \times 2$ matrix $\mathbf{\Psi}$ can be created directly with the `DiagonalMatrix` function; the default symbol is a $\psi$.

```
PSI = DiagonalMatrix(2); show(PSI)
```

evaluate

$$\begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix}$$

The matrix $\mathbf{\Lambda}$ is $4 \times 3$.

```
LAMBDA = ZeroMatrix(4,3); LAMBDA[0,0] = 1 ; LAMBDA[1,1] = var('lambda1')
LAMBDA[2,2] = var('lambda2') ; LAMBDA[3,2] = var('lambda2')
show(LAMBDA)
```

evaluate

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \\ 0 & 0 & \lambda_2 \end{pmatrix}$$

The matrix $\mathbf{\Omega} = cov(\mathbf{e}_i)$ has $Var(e_{i,1}) = 0$, so that the observable variable $X_i$ can also appear in the latent variable model.

```
OMEGA = ZeroMatrix(4,4); OMEGA[1,1] = var('omega1')
OMEGA[2,2] = var('omega2'); OMEGA[3,3] = var('omega2')
show(OMEGA)
```

evaluate

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \omega_1 & 0 & 0 \\ 0 & 0 & \omega_2 & 0 \\ 0 & 0 & 0 & \omega_2 \end{pmatrix}$$

Following the two-stage model formulation, the next step is to calculate $\mathbf{\Phi} = cov(\mathbf{F}_i)$. Then $\mathbf{\Phi}$ will be used as an ingredient in the calculation of $\mathbf{\Sigma}$.

```
# Calculate PHI = cov(F)
PHI = PathCov(Phi=PHIx,Beta=BETA,Gamma=GAMMA,Psi=PSI)
show(PHI)
```

evaluate

$$\begin{pmatrix} \phi & \gamma\phi & \beta\gamma\phi \\ \gamma\phi & \gamma^2\phi+\psi_1 & (\gamma^2\phi+\psi_1)\beta \\ \beta\gamma\phi & (\gamma^2\phi+\psi_1)\beta & \beta^2\gamma^2\phi+\beta^2\psi_1+\psi_2 \end{pmatrix}$$

Now, $\boldsymbol{\Sigma}$ is calculated from $\boldsymbol{\Phi}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$, yielding Expression (1.7). I used Sage to generate the LaTeXcode for the matrix by double-clicking on the object in the Sage worksheet, and then manually deleted the lower triangular part of the matrix so it would fit better on the page. It was still a lot better than typesetting the matrix myself.

```
# Calculate SIGMA = cov(D)
SIGMA = FactorAnalysisCov(Lambda=LAMBDA,Phi=PHI,Omega=OMEGA)
show(SIGMA)
```

evaluate

$$\begin{pmatrix} \phi & \gamma\lambda_1\phi & \beta\gamma\lambda_2\phi & \beta\gamma\lambda_2\phi \\ \gamma\lambda_1\phi & (\gamma^2\phi+\psi_1)\lambda_1^2+\omega_1 & (\gamma^2\phi+\psi_1)\beta\lambda_1\lambda_2 & (\gamma^2\phi+\psi_1)\beta\lambda_1\lambda_2 \\ \beta\gamma\lambda_2\phi & (\gamma^2\phi+\psi_1)\beta\lambda_1\lambda_2 & (\beta^2\gamma^2\phi+\beta^2\psi_1+\psi_2)\lambda_2^2+\omega_2 & (\beta^2\gamma^2\phi+\beta^2\psi_1+\psi_2)\lambda_2^2 \\ \beta\gamma\lambda_2\phi & (\gamma^2\phi+\psi_1)\beta\lambda_1\lambda_2 & (\beta^2\gamma^2\phi+\beta^2\psi_1+\psi_2)\lambda_2^2 & (\beta^2\gamma^2\phi+\beta^2\psi_1+\psi_2)\lambda_2^2+\omega_2 \end{pmatrix}$$

To generate the example of two numerically different parameter sets that yield the same $\boldsymbol{\Sigma}$, I looked at the equations in (1.10) to find distinct $\boldsymbol{\theta}$ vectors corresponding to the same $\boldsymbol{\theta}'$. There was a bit of trial and error, and Sage made it really convenient to do the numerical calculations. A Sage object like a matrix may be treated as a *function* of the symbolic variables that appear in it.

```
SIGMA(gamma=2,beta=4,lambda1=1,lambda2=1,psi1=4,psi2=16,
phi=1,omega1=1,omega2=1)
```

evaluate

$$\begin{pmatrix} 1 & 2 & 8 & 8 \\ 2 & 9 & 32 & 32 \\ 8 & 32 & 145 & 144 \\ 8 & 32 & 144 & 145 \end{pmatrix}$$

```
SIGMA(gamma=1,beta=2,lambda1=2,lambda2=4,psi1=1,psi2=1,
phi=1,omega1=1,omega2=1)
```

evaluate

$$\begin{pmatrix} 1 & 2 & 8 & 8 \\ 2 & 9 & 32 & 32 \\ 8 & 32 & 145 & 144 \\ 8 & 32 & 144 & 145 \end{pmatrix}$$

The same Sage capability was used to generate Expression (1.11), the re-parameterized $\mathbf{\Sigma}$ matrix under the second-level surrogate model. Rather than starting from the surrogate model equations (1.9) and re-doing the whole calculation, I just evaluated the $\mathbf{\Sigma}$ of (1.7) at $\lambda_1 = \lambda_2 = 1$.

```
SIGMA(lambda1=1,lambda2=1)
```

evaluate

$$\begin{pmatrix} \phi & \gamma\phi & \beta\gamma\phi & \beta\gamma\phi \\ \gamma\phi & \gamma^2\phi + \omega_1 + \psi_1 & (\gamma^2\phi + \psi_1)\beta & (\gamma^2\phi + \psi_1)\beta \\ \beta\gamma\phi & (\gamma^2\phi + \psi_1)\beta & \beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 & \beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2 \\ \beta\gamma\phi & (\gamma^2\phi + \psi_1)\beta & \beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2 & \beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 \end{pmatrix}$$

The covariance structure equations may now be solved by inspection, verifying identifiability of the parameters in the re-parameterized model. But it is instructive to solve the equations using Sage. The necessary ingredients are a list of equations and a list of unknown parameters for which to solve.

The `sem` package has the specialized function `Parameters` for extracting parameters from matrices, so they don't all need to be re-typed. It works on the original parameter matrices, not on computed matrices like $\mathbf{\Phi}$ or $\mathbf{\Sigma}$. For example, the $4 \times 3$ matrix $\mathbf{\Lambda}$ contains just two parameters, $\lambda_1$ and $\lambda_2$.

```
Parameters(LAMBDA) # Don't need these - just an example
```

evaluate

$$(\lambda_1, \lambda_2)$$

```
param = [phi,beta,gamma] # Start with this
param.extend(Parameters(PSI))
param.extend(Parameters(OMEGA))
param
```

evaluate

$$(\phi, \beta, \gamma, \psi_1, \psi_2, \omega_1, \omega_2)$$

Notice how the list `param` has been `extended` by adding the contents of $\mathbf{\Psi}$ and $\mathbf{\Omega}$. For big matrices with lots of parameters, this is a real convenience.

The next step is to set up the equations to solve. The Sage `solve` function needs the same number of equations as unknowns, so giving it the full set of 10 equations in 7 unknowns will not work. But we'll set up all 10 equations anyway to see what happens.

```
# Now set up equations to solve
S  = SIGMA(lambda1=1,lambda2=1) # Sigma under surrogate model
S2 = SymmetricMatrix(4,'sigma')
eqns = [] # Empty list
for i in range(4):              # i goes from 0 to 3
    for j in range(i+1):        # j goes from 0 to i
        item = S[i,j]==S2[i,j]  # An equation
        eqns.append(item)       # Append to list of equations
eqns # Not easy to look at, but there is a scroll bar
```

evaluate

$$(\phi = \sigma_{11}, \gamma\phi = \sigma_{12}, \gamma^2\phi + \omega_1 + \psi_1 = \sigma_{22}, \beta\gamma\phi = \sigma_{13}, (\gamma^2\phi + \psi_1)\beta = \sigma_{23}, \beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 = \sigma_{33},$$

The object `eqns` is a *list* of equations; you can tell it's a list because it's enclosed in brackets. As the comment statement says, it's not very easy to look at, but there is a scroll bar. So in a Sage environment, you can examine the output that runs off the page in this document. Here's a more convenient way to look at the covariance structure equations.

```
for item in eqns: item
```

evaluate

$$\phi = \sigma_{11}$$
$$\gamma\phi = \sigma_{12}$$
$$\gamma^2\phi + \omega_1 + \psi_1 = \sigma_{22}$$
$$\beta\gamma\phi = \sigma_{13}$$
$$(\gamma^2\phi + \psi_1)\beta = \sigma_{23}$$
$$\beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 = \sigma_{33}$$
$$\beta\gamma\phi = \sigma_{14}$$
$$(\gamma^2\phi + \psi_1)\beta = \sigma_{24}$$
$$\beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2 = \sigma_{34}$$
$$\beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 = \sigma_{44}$$

It would seem easy to ask Sage to solve these ten equations in seven unknowns. It's easy to ask, but the answer is not what we're looking for.

```
solve(eqns,param)
```

evaluate

[]

That little rectangle is a left square bracket followed by a right square bracket; that is, it's an empty list (empty set), meaning that the system of equations has no general solution. This happens because, for example, the fourth equation in the list says $\beta\gamma\phi = \sigma_{13}$, while the seventh equation says $\beta\gamma\phi = \sigma_{14}$. To Sage, $\sigma_{13}$ and $\sigma_{14}$ are just numbers, and there is no reason to assume they are equal. Thus there is no *general* solution.

Actually, because we think of the $\sigma_{ij}$ values as arising from a single, fixed point in the parameter space, we recognize $\sigma_{13} = \sigma_{14}$ (and also $\sigma_{23} = \sigma_{24}$ and $\sigma_{33} = \sigma_{44}$) as realities – distinctive features that the model imposes on the covariance matrix $\mathbf{\Sigma}$. But Sage can't know this unless we tell it, and I don't know how to do that. It's easiest to just eliminate the redundant equations.

```
extra = [9,7,6] # Redundant equations, starting with index zero
for item in extra: show(eqns[item])
```

evaluate

$\beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 = \sigma_{44}$
$(\gamma^2\phi + \psi_1)\beta = \sigma_{24}$
$\beta\gamma\phi = \sigma_{14}$

Removing the the extra equations from the list and then taking a look ...

```
for item in extra: eqns.remove(eqns[item])
for item in eqns: item
```

evaluate

$\phi = \sigma_{11}$
$\gamma\phi = \sigma_{12}$
$\gamma^2\phi + \omega_1 + \psi_1 = \sigma_{22}$
$\beta\gamma\phi = \sigma_{13}$
$(\gamma^2\phi + \psi_1)\beta = \sigma_{23}$
$\beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 = \sigma_{33}$
$\beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2 = \sigma_{34}$

Now it is possible to solve the remaining seven equations in seven unknowns. The solution will be easier to use in later calculations if it is obtained in the form of a *dictionary*. To see if the solution is unique, first check the *length* of the list of dictionaries returned by `solve`.

```
# Return solution as list of dictionaries
solist = solve(eqns,param,solution_dict=True)
len(solist)
```

evaluate

1

There is only one item in the list of dictionaries; it's item zero. The key of the dictionary is the parameter, and the value is the solution, which for us will be some function of the $\sigma_{ij}$ quantities. Dictionary entries take the form Key-Colon-Value. Dictionaries are inherently unordered.

```
sol = solist[0]; sol   # Item 0 of the list; there's just one.
```

evaluate

$$\left\{\phi : \sigma_{11}, \psi_1 : \frac{\sigma_{11}\sigma_{12}\sigma_{23}-\sigma_{12}^2\sigma_{13}}{\sigma_{11}\sigma_{13}}, \beta : \frac{\sigma_{13}}{\sigma_{12}}, \omega_2 : \sigma_{33}-\sigma_{34}, \gamma : \frac{\sigma_{12}}{\sigma_{11}}, \omega_1 : -\frac{\sigma_{12}\sigma_{23}-\sigma_{13}\sigma_{22}}{\sigma_{13}}, \psi_2 : \frac{\sigma_{12}\sigma_{34}-\sigma_{13}\sigma_{23}}{\sigma_{12}}\right\}$$

The dictionary format makes it convenient to refer to the solution for a parameter — for example, the solution for $\psi_2$.

```
sol[psi2]
```

evaluate

$$\frac{\sigma_{12}\sigma_{34}-\sigma_{13}\sigma_{23}}{\sigma_{12}}$$

Dictionaries are hard to look at when they have a lot of items. Here is one way to take a quick look at a solution. Dictionary entries are expressed as *tuples* of the form (Parameter, Solution). Since the `for` loop below is going through the list of parameters, the output is in that order.

```
for item in param:
    item, sol[item]
```

evaluate

$$(\phi, \sigma_{11})$$
$$\left(\beta, \frac{\sigma_{13}}{\sigma_{12}}\right)$$
$$\left(\gamma, \frac{\sigma_{12}}{\sigma_{11}}\right)$$
$$\left(\psi_1, \frac{\sigma_{11}\sigma_{12}\sigma_{23}-\sigma_{12}^2\sigma_{13}}{\sigma_{11}\sigma_{13}}\right)$$
$$\left(\psi_2, \frac{\sigma_{12}\sigma_{34}-\sigma_{13}\sigma_{23}}{\sigma_{12}}\right)$$
$$\left(\omega_1, -\frac{\sigma_{12}\sigma_{23}-\sigma_{13}\sigma_{22}}{\sigma_{13}}\right)$$
$$(\omega_2, \sigma_{33}-\sigma_{34})$$

That's okay for a quick look, and the syntax is intuitive. Equations are nicer, though. In the following, realize that nothing is getting *assigned*. Rather, `item==sol[item]` just causes that equation to be displayed.

```
for item in param: item==sol[item]
```

evaluate

$$\phi = \sigma_{11}$$
$$\beta = \frac{\sigma_{13}}{\sigma_{12}}$$
$$\gamma = \frac{\sigma_{12}}{\sigma_{11}}$$
$$\psi_1 = \frac{\sigma_{11}\sigma_{12}\sigma_{23} - \sigma_{12}^2\sigma_{13}}{\sigma_{11}\sigma_{13}}$$
$$\psi_2 = \frac{\sigma_{12}\sigma_{34} - \sigma_{13}\sigma_{23}}{\sigma_{12}}$$
$$\omega_1 = -\frac{\sigma_{12}\sigma_{23} - \sigma_{13}\sigma_{22}}{\sigma_{13}}$$
$$\omega_2 = \sigma_{33} - \sigma_{34}$$

The dictionary `sol` gives parameters in terms of the $\sigma_{ij}$ values. It can also be useful to have a dictionary that goes in the other direction, where the input is in terms $\sigma_{ij}$ and the output is in terms of the model parameters. The function `SigmaOfTheta` sets up such a dictionary; see Appendix B or try `SigmaOfTheta?` in a Sage environment for more detail. In the following, the dictionary is in terms of the *original* (not surrogate) model parameters.

```
# Original covariance matrix as a function of theta
theta = SigmaOfTheta(SIGMA)
# theta is a dictionary
# For example, sigma12 = gamma lambda1 phi
sigma12(theta)
```

evaluate

$$\gamma\lambda_1\phi$$

Such a dictionary can be used to evaluate big, messy functions of $\mathbf{\Sigma}$, including the solutions in the dictionary `sol`.

```
# What is the solution for psi2 (that's psi2-prime) in terms of
# ORIGINAL model parameters?
sol[psi2](theta)
```

evaluate

$$-\frac{\left(\gamma^2\phi+\psi_1\right)\beta^2\gamma\lambda_1\lambda_2^2\phi - \left(\beta^2\gamma^2\phi+\beta^2\psi_1+\psi_2\right)\gamma\lambda_1\lambda_2^2\phi}{\gamma\lambda_1\phi}$$

```
Simplify(_) # Underscore refers to the last item
```

evaluate

$\lambda_2^2\psi_2$

Where in the original parameter space is $\psi_1'$ identifiable? These are the points in the parameter space where the denominator of the solution (that's $\sigma_{11}\sigma_{13}$) is non-zero. Evaluating the denominator as a function of the model parameters $\boldsymbol{\theta}$,

```
# Where is psi1-prime identifiable?
denominator(sol[psi2])(theta)
```

evaluate

$\beta\gamma\lambda_2\phi^2$

Thus, $\beta$, $\gamma$ and $\lambda_2$ must all be non-zero in order for $\psi_1' = \lambda_1^2\psi_1$ to be identifiable. This is the end of the `Sage` example.

### 1.4.5 Yet another type of surrogate model

In some structural equation models, variables that are obviously measured with error are assumed to be observable. Invariably, the assumption is adopted so that the parameters of the resulting model will be identifiable. Since it is practically impossible to measure anything without error, almost every model that assumes error-free measurement is either dangerously[10] unrealistic, or a surrogate for some model that is more reasonable.

For an example, we will turn to Section 0.11 of Chapter 0, where extra response variables were used to identify the parameters of regression models with measurement error in the explanatory variables. Consider a centered version of model (53) on page 106.

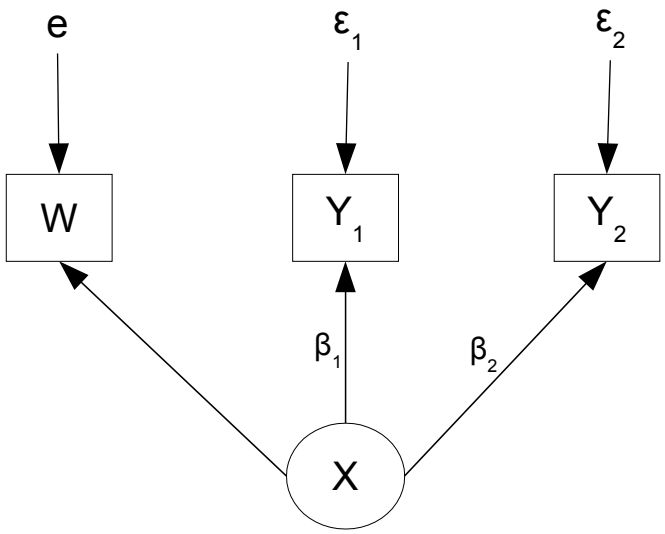
$$\begin{aligned} W_i &= X_i + e_i \\ Y_{i,1} &= \beta_1 X_i + \epsilon_{i,1} \\ Y_{i,2} &= \beta_2 X_i + \epsilon_{i,2} \end{aligned} \tag{1.12}$$

The path diagram is shown in Figure **??**. To give this some content, consider the question of whether smoking cigarettes can help you lose weight. We will limit the study to young adults who smoke at least occasionally, and who do not exercise regularly. Suppose that the latent variable $X_i$ is amount of smoking, $W_i$ is *reported* number of cigarettes smoked daily, $Y_{i,1}$ is body mass index[11], and $Y_{i,2}$ is resting heart rate. Interest is in the connection

[10] Section 0.7 in Chapter 0 points out the disastrous effects of ignoring measurement error in multiple regression, and it is natural to expect similar things to happen in a more general setting. Except possibly for experimentally manipulated exogenous variables, assuming perfect measurement is not something to be done lightly.

[11] Weight in kilograms divided by squared height in meters. Big numbers mean you are heavier for your height.

Figure 1.4: Path diagram of the surrogate model for credit card debt




between amount of smoking and body mass index (BMI), represented by $\beta_1$. Heart rate (known to be increased by smoking) is an extra response variable.

Notice that in $W_i = X_i + e_i$, the factor loading for equals one; this means that it's a surrogate model. As described starting on page 106, the parameters of this model are identifiable — but it's far from realistic. Body mass index surely cannot be measured without error, because height and weight are measured with error. As for resting heart rate, it will vary over the time of day, and also with things like ambient noise level and recent exertion.

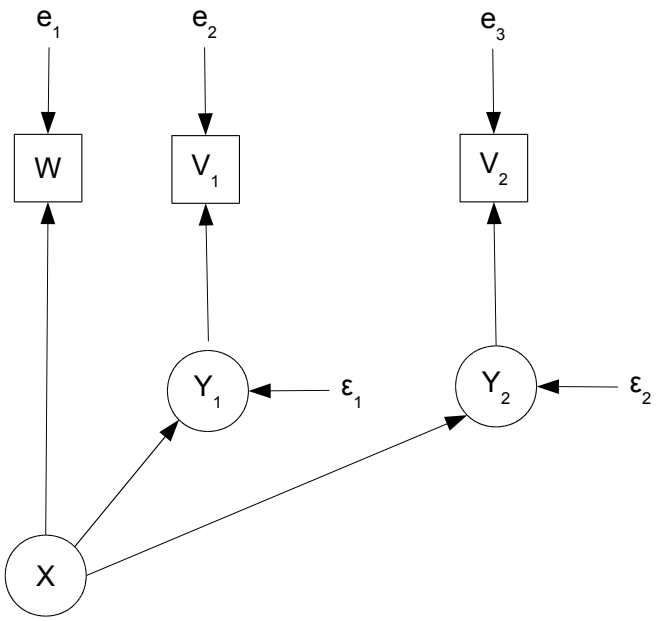
Figure 1.5 depicts a somewhat more reasonable model for the smoking example, and it is proposed as the original model. In this model, $Y_{i,1}$ is true body mass index, while $V_{i,1}$ is the measured version. $Y_{i,2}$ is true average resting heart rate, while $V_{i,2}$ is the snapshot measured with error that appears in the data file. The equations of the proposed original model are

$$\begin{aligned} W_i &= \nu_1 + \lambda_1 X_i + e_{i,1} \\ Y_{i,1} &= \alpha_1 + \beta_1 X_i + \epsilon_{i,1} \\ Y_{i,2} &= \alpha_2 + \beta_2 X_i + \epsilon_{i,2} \\ V_{i,1} &= \nu_2 + \lambda_2 Y_{i,1} + e_{i,2} \\ V_{i,2} &= \nu_3 + \lambda_3 Y_{i,2} + e_{i,3}, \end{aligned} \tag{1.13}$$

where $Var(X_i) = \phi$, $Var(e_{i,1}) = \omega_1$, $Var(e_{i,2}) = \omega_2$, $Var(e_{i,3}) = \omega_3$, $Var(\epsilon_{i,1}) = \psi_1$ and $Var(\epsilon_{i,2}) = \psi_2$. As the path diagram indicates, all error terms are independent of $X_i$ and of one another. Because $W_i$, $V_{i,1}$ and $V_{i,2}$ are direct measurements of the corresponding latent variables, it is safe to assume that the factor loadings $\lambda_1$, $\lambda_2$ and $\lambda_3$ are all positive.

Centering the variables and setting all three factor loadings to one yields a second

Figure 1.5: Path diagram of the original model for credit card debt




level surrogate model that preserves the signs of $\beta_1$ and $\beta_2$, though not their actual values. There are now eight parameters, but still only six covariance structure equations. By the parameter count rule, the parameters of this model cannot be identified. However,

$$\begin{aligned} V_{i,1} &= Y_{i,1} + e_{i,2} \\ &= (\beta_1 X_i + \epsilon_{i,1}) + e_{i,2} \\ &= \beta_1 X_i + (\epsilon_{i,1} + e_{i,2}) \\ &= \beta_1 X_i + \epsilon'_{i,1}. \end{aligned}$$

Re-labelling $V_{i,1}$ as $Y'_{i,1}$, we have the model equation $Y'_{i,1} = \beta_1 X_i + \epsilon'_{i,1}$, with $Var(\epsilon'_{i,1}) = \psi'_1 = \psi_1 + \omega_2$. The same procedure yields $Y'_{i,2} = \beta_2 X_i + \epsilon'_{i,2}$, with $Var(\epsilon'_{i,2}) = \psi'_2 = \psi_2 + \omega_3$.

Dropping the primes as usual to hide the evidence of our strange activities, we arrive once more at the model equations (1.12). All along, this model was a surrogate for the original model of Figure 1.5 and Equations (1.13). It never really assumed that credit card debt and vehicle value were observable. Rather, the change of variables $\epsilon'_{i,1} = \epsilon_{i,1} + e_{i,2}$ was carried out to obtain the re-parameterization $\psi'_1 = \psi_1 + \omega_2$, and the change of variables $\epsilon'_{i,2} = \epsilon_{i,2} + e_{i,3}$ was carried out to obtain the re-parameterization $\psi'_2 = \psi_2 + \omega_3$. Notationally, the result looks like a model with error-free measurement of $Y_{i,1}$ and $Y_{i,2}$ — but in this case appearances are deceiving. Surrogate models are never to be taken literally.

The beginning of Section 0.7 of Chapter 0 suggested that in multiple regression, measurement error in *response* variables may be safely ignored, and the result was a useful surrogate model. The same principle applies here. In general, suppose that an endogenous

variable $Y_{i,j}$ in the latent variable model is a *purely* endogenous variable, in the sense that there are no arrows from $Y_{i,j}$ to any other latent variable. In addition, suppose that $Y_{i,j}$ is measured with error in a single observable variable $V_{i,j}$, so that after centering,

$$\begin{aligned} Y_{i,j} &= \mathbf{r}_j^\top \mathbf{x}_i + \epsilon_{i,j} \\ V_{i,j} &= \lambda_j Y_{i,j} + e_{i,j}, \end{aligned}$$

where $\mathbf{r}_j = \mathbf{r}_j(\boldsymbol{\beta}, \boldsymbol{\Gamma})$ denotes row $j$ of the matrix $(\mathbf{I} - \boldsymbol{\beta})^{-1}\boldsymbol{\Gamma}$; see Expression (1.3) on page 146. In addition, suppose that $\epsilon_{i,j}$ and $e_{i,j}$ are independent of one another and of all other exogenous variables in the model, with $Var(\epsilon_{i,j}) = \psi_j$ and $Var(e_{i,j}) = \omega_j$.

At this point it would be possible and legitimate to implicitly re-parameterize by setting $\lambda_j = 1$ as in the Credit Card Debt example. As an alternative, the absorption of the un-knowable factor loading will be accomplished by the re-parameterization that combines $\psi_j$ and $\omega_j$, all in one step.

$$\begin{aligned} V_{i,j} &= \lambda_j Y_{i,j} + e_{i,j} \\ &= \lambda_j(\mathbf{r}_j^\top \mathbf{x}_i + \epsilon_{i,j}) + e_{i,j} \\ &= (\lambda_j \mathbf{r}_j)^\top \mathbf{x}_i + (\lambda_j \epsilon_{i,j} + e_{i,j}) \\ &= \mathbf{r}_j'^\top \mathbf{x}_i + \epsilon'_{i,j}, \end{aligned}$$

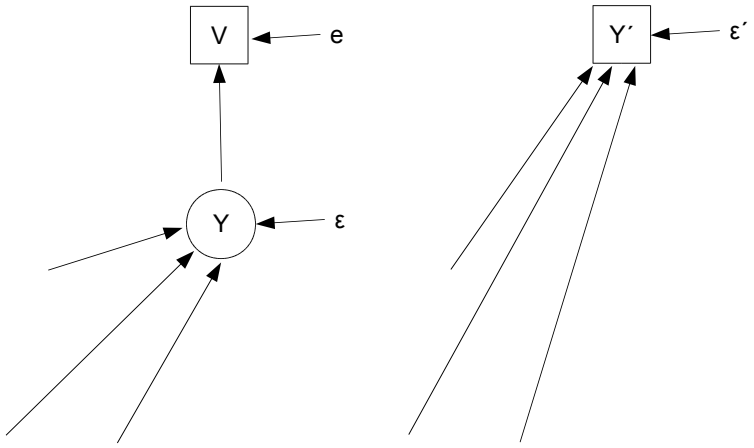
with $Var(\epsilon'_{i,j}) = \psi'_j = \lambda_j^2 \psi_j + \omega_j$. The $\beta$ and $\gamma$ parameters in $\mathbf{r}_j$ are also re-expressed in this step. Now $V_{i,j}$ may be called $Y'_{i,j}$ without doing any harm. The result is a new model in which

- The parameters are *functions* of the parameters in the original model.
- The dimension of the parameter space is two less, so the new parameter vector should be easier to identify.
- The meaning of the new parameters is clear. The $\beta$ and $\gamma$ parameters in $\mathbf{r}_j$ are positive multiples of what they were before, while any *separate* meaning that $\psi_j$ and $\omega_j$ may have had is lost. They were probably not knowable anyway.
- After dropping the primes, it *looks* like $Y_{i,j}$ is measured without error, but that is an illusion. No such claim was ever intended.

The situation is shown graphically in Figure 1.6. When a latent endogenous variable does not affect any other latent variables and is expressed by only one observable variable, it is acceptable to drop the latent variable from the model, and run all the arrows directly to the observable variable.

**Comments** Virtually all structural equation models used in practice are surrogate models, and most of them have the features described here. While the re-parameterizations

Figure 1.6: Direct path to the observed variable




are very standard, the terms "original model" and "surrogate model" are not. I made them up, and they will not be found elsewhere[12].

Experts in the field undoubtedly know that what's happening is a series of re-parameterizations, but this is often not acknowledged in textbooks. Instead, the process is presented as a harmless restriction of the parameter space, adopted in order to identify the parameters. I think it's really helpful to point out how the re-parameterizations are accomplished by change-of-variable operations. This reveals effects on other parameters in the model (not just the ones that seem to be restricted), and makes it possible to specify the *meanings* of the new parameters in terms of the parameters of the original model.

## 1.5 Maximum likelihood

In most structural equation modeling software, the default method of parameter estimation is numerical maximum likelihood[13]. The exogenous variables and error terms are assumed multivariate normal, and consequently the joint distribution of the observable variables is multivariate normal too. It will be seen in theorem **??** that when the normal assumption is clearly wrong, maximum likelihood estimates based on normality are still consistent. They are also asymptotically normal under conditions that are widely accepted. This makes bootstrap standard errors potentially very useful when the assumption of normality is questionable. Bootstrapping in lavaan is easy, and theoretically based robust standard errors are also available.

[12]That is, unless others find the terminology useful and it catches on. It's always possible, I suppose.

[13]The reader is referred to Section A.6.3 in Appendix A for material on maximum likelihood and related concepts.

### 1.5.1 Estimation

Let $\mathbf{d}_1, \ldots, \mathbf{d}_n$ be a random sample from a $k$-dimensional multivariate normal distribution with expected value $\boldsymbol{\mu}$ and varance-covariance matrix $\boldsymbol{\Sigma}$. The likelihood is

$$\begin{aligned} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \prod_{i=1}^{n} \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}(2\pi)^{\frac{k}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{d}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{d}_i - \boldsymbol{\mu})\right\} \\ &= |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-nk/2} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{d}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{d}_i - \boldsymbol{\mu})\right\} \\ &= |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-nk/2} \exp -\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) + (\overline{\mathbf{d}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{d}} - \boldsymbol{\mu})\right\}, \end{aligned}$$

where $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i - \overline{\mathbf{d}})(\mathbf{d}_i - \overline{\mathbf{d}})^\top$ is the sample variance-covariance matrix.

Let $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ be a vector of parameters from a structural equation model; $\boldsymbol{\Theta}$ is the parameter space. For example, $\boldsymbol{\theta}$ could be the the unique elements in the parameter matrices in the original Model (1.1), restricted only by modeling considerations. Then the likelihood is a function of $\boldsymbol{\theta}$ through $\boldsymbol{\mu} = \mu(\boldsymbol{\theta})$ and $\boldsymbol{\Sigma} = \Sigma(\boldsymbol{\theta})$, as given in Expressions (1.4).

Maximizing the likelihood over $\boldsymbol{\theta}$ is equivalent to minimizing the minus log likelihood

$$\begin{aligned} -\ell(\boldsymbol{\theta}) &= \frac{n}{2}\log|\Sigma(\boldsymbol{\theta})| + \frac{nk}{2}\log(2\pi) + \frac{n}{2}tr(\widehat{\boldsymbol{\Sigma}}\Sigma(\boldsymbol{\theta})^{-1}) \\ &\quad + \frac{n}{2}\left(\overline{\mathbf{d}} - \boldsymbol{\mu}(\boldsymbol{\theta})\right)^\top \Sigma(\boldsymbol{\theta})^{-1}\left(\overline{\mathbf{d}} - \boldsymbol{\mu}(\boldsymbol{\theta})\right) \end{aligned} \tag{1.14}$$

For any set of observed data values, the minus log likelihood defines a high-dimensional surface floating over the parameter space $\boldsymbol{\Theta}$. The maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ is the point in $\boldsymbol{\Theta}$ where the surface is lowest. One might try the calculus approach, partially differentiating the log likelihood and setting all the derivates to zero. This ypically yields a system of equations that nobody can solve, so it really does not help us locate the point where the minimum value ofccurs. To find the point numerically, choose a starting value as close to the answer as possible and move downhill. Choice of good starting values is important, because the likelihood surface can have many local maxima and minima, and other topological features that are "interesting," but not in a good way.

Ideally, the numerical search will terminate at the unique minimum of the function. Geometrically, the surface at that point will be level and concave up. Analytically, the gradient will be zero, and the eigenvalues of the Hessian matrix will all be positive. As described in Appendix A, the Hessian is the observed Fisher information matrix evaluated at $\widehat{\boldsymbol{\theta}}$, and its inverse is the approximate asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}$.

When the parameters are not identifiable, this procedure fails. The likelihood is constant on collections of *functions* of $\boldsymbol{\theta}$ that are identifiable. Typically, the numerical search reaches the bottom of a high-dimensional valley, and at the bottom of that valley is a contour (think of a winding, invisibly thin river) where the minus log likelihood is constant. The gradient is zero at any point on the surface of the river, but the surface is not concave up in every direction. It follows that the Hessian matrix has one or more

eigenvalues equal to zero. The determinant of the Hessian equals zero, and inverting it to approximate the asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}$ is impossible. In this situation, good software complains loudly[14].

**Re-parameterization** Since the parameters of the original Model (1.1) are not identifiable, directly fitting it by maximum likelihood is out of the question. Re-parameterization is necessary. Following Section A.6.1, the first step is to lose the expected values and intercepts. Let $\boldsymbol{\kappa} = \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F$, where the partitioned matrix

$$\boldsymbol{\mu}_F = \begin{pmatrix} \boldsymbol{\mu}_x \\ \hline (\mathbf{I}-\boldsymbol{\beta})^{-1}(\boldsymbol{\alpha}+\boldsymbol{\Gamma}\boldsymbol{\mu}_x) \end{pmatrix}.$$

Under this re-parameterization, the new parameter vector $\boldsymbol{\theta}'$ consists of $\boldsymbol{\kappa}$, plus all the parameters that appear in $\boldsymbol{\Sigma}$ — that is, the unique elements of $\boldsymbol{\Phi}_x, \boldsymbol{\Psi}, \boldsymbol{\Omega}, \boldsymbol{\beta}, \boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$.

Because the new parameter $\boldsymbol{\kappa}$ is exactly $\boldsymbol{\mu}(\boldsymbol{\theta})$, the minus log likelihood is minimal when $\boldsymbol{\kappa} = \overline{\mathbf{d}}$, regardless of the values of the remaining parameters. The second line of Expression (1.14) disappears, and the task is now to minimize the first line with respect to the parameters that appear in the covariance matrix.

The remaining parameters are still not identifiable in general. Further re-parameterization is necessary, and the re-parameterizations corresponding to standard surrogate models are often very helpful. The parameters of a good surrogate model are identifiable functions of the original model's parameters. After the centering step, re-parameterization is carried out by a set of change-of-variables operations involving only latent variables. As a result, the parameters of the original model appear in the covariance matrix only through functions of $\boldsymbol{\theta}$ that correspond to the parameters of the surrogate model. If the re-parameterizations are well chosen, the maximum of the likelihood under the surrogate model is identical to the maximum of the likelihood under the original model. If in addition, the likelihood function achieves its maximum at a point where the parameters of the surrogate model are identifiable, then the maximum will be unique. The minus log likelihod will be nicely concave up at this point in the parameter space of the re-parameterized model. The Hessian matrix (observed Fisher Information) will be positive definite, and its inverse will provide an approximate asymptotic covariance for the estimated parameters of the surrogate model. This is the main ingredient for $Z$-tests and Wald tests. The height of the minus log likelihood at the MLE is used in likelihood ratio tests.

Once the expected values and intercepts have been absorbed into $\boldsymbol{\kappa}$, we implicitly estimate the identifiable function $\boldsymbol{\kappa}$ with the vector of sample means $\overline{\mathbf{d}}$, and then forget about it, basing all inference upon the sample variance-covariance matrix. This is standard practice, but it raises a few issues. First, note that while $\boldsymbol{\kappa}$ is a function of the un-knowable parameters $\boldsymbol{\nu}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}_x$, it is also a function of $\boldsymbol{\beta}, \boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$. These last three matrices

[14]This encourages some naive users to simply run their structural equation modeling software without thinking very hard about identifiability, trusting that if the parameters are not identifiable, the search will blow up. Unfortunately, the search can blow up numerically for other reasons, and sometimes the symptoms can be very similar to those arising from lack of identifiability. It is much better to check identifiability mathematically, before trying to fit the model.

are often of primary interest. Might $\overline{\mathbf{d}}$ contain some information about them? Are we are throwing this information away?

The answer is no, provided that the intercept term $\boldsymbol{\nu}$ is not restricted by modeling considerations. Suppose that the first line of the minus log likelihood (1.14) is minimized, regardless of whether that minimum is unique. Now consider the effect of adjusting $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}$ or $\boldsymbol{\Lambda}$. The value of the first line will increase or remain the same. Now look at the second line, recalling that $\boldsymbol{\mu}(\boldsymbol{\theta}) = \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F$. Regardless of how the values of the other parameters change, $\boldsymbol{\nu}$ can always be adjusted so that $\overline{\mathbf{d}} - \boldsymbol{\mu}(\boldsymbol{\theta}) = \mathbf{0}$. This makes the second line equal to zero, which is as low as it can be. Therefore, the second line of (1.14) makes no contribution to the MLEs of parameters appearing in the covariance matrix $\boldsymbol{\Sigma}$ — that is, provided that $\boldsymbol{\nu}$ is unrestricted.

Since inference is to be be based on the covariance matrix, it saves mental effort to employ the centered surrogate model. But we never actually *fit* the centered surrogate model. We cannot, because the change of variables involves subtracting expected values from the observed data, and those expected values (elements of $\boldsymbol{\mu}(\boldsymbol{\theta}) = \boldsymbol{\kappa}$) are unknown. On the other hand, it is possible to fit an *approximate* centered model by using the vector of sample means in place of $\boldsymbol{\mu}(\boldsymbol{\theta})$. That is,

$$\overset{c}{\mathbf{d}}_i = \mathbf{d}_i - \mu(\boldsymbol{\theta}) \approx \mathbf{d}_i - \overline{\mathbf{d}}$$

by the Law of Large Numbers. The approximation will be very good for large samples. Letting $\overset{c}{\mathbf{d}}_i$ refer to $\mathbf{d}_i - \overline{\mathbf{d}}$ for now, the model is that $\overset{c}{\mathbf{d}}_1, \ldots \overset{c}{\mathbf{d}}_n$ are a random sample from a multivariate normal distribution with expected value zero and covariance matrix $\Sigma(\boldsymbol{\theta})$. The observations are not quite independent because the same random quantity $\overline{\mathbf{d}}$ is subtracted from each one, but the covariances go to zero as $n \to \infty$. The likelihood function is

$$\begin{aligned} L(\boldsymbol{\Sigma}) &= \prod_{i=1}^{n} \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}(2\pi)^{\frac{k}{2}}} \exp\left\{-\frac{1}{2}\, \overset{c}{\mathbf{d}}_i^{\top} \boldsymbol{\Sigma}^{-1}\, \overset{c}{\mathbf{d}}_i\right\} \\ &= |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-nk/2} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{d}_i - \overline{\mathbf{d}})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{d}_i - \overline{\mathbf{d}})\right\} \\ &= |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-nk/2} \exp -\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})\right\}. \end{aligned}$$

The minus log likelihood is just the first line of (1.14). So, estimating $\boldsymbol{\kappa} = \mu(\boldsymbol{\theta})$ with $\overline{\mathbf{d}}$ and setting it aside is the same as fitting the approximate centered surrogate model. Either way, the intercepts and expected values disappear.

### 1.5.2 Hypothesis testing

**z-tests** The maximum likelihood estimates are asymptotically normal under general conditions, so that for a scalar parameter $\theta_j$,

$$z = \frac{\widehat{\theta}_j - \theta_j}{s_{\theta_j}} \tag{1.15}$$

has an approximate standard normal distribution for large samples, where $s_{\theta_j}$ is the standard error (estimated standard deviation) of $\widehat{\theta}_j$, obtained by taking the square root of a diagonal element the estimated asymptotic covariance matrix. There are various good ways to estimate the asymptotic covariance matrix[15]. Squaring the $z$ statistic yields a Wald chi-square statistic with one degree of freedom. Wald tests are the topic of the next brief section.

**Wald tests** As described in Section A.6.7 of Appendix A, a linear null hypothesis of the form $H_0 : \mathbf{L}\boldsymbol{\theta} = \mathbf{h}$ can be tested using the statistic

$$W_n = (\mathbf{L}\widehat{\boldsymbol{\theta}}_n - \mathbf{h})^\top (\mathbf{L}\widehat{\mathbf{V}}_n\mathbf{L}^\top)^{-1}(\mathbf{L}\widehat{\boldsymbol{\theta}}_n - \mathbf{h}). \tag{1.16}$$

Under the null hypothesis, $W_n$ has an approximate chi-squared distribution with $r$ degrees of freedom, where $r$ is the number of rows in the matrix $\mathbf{L}$. In the formula, $\widehat{\mathbf{V}}_n$ is the estimated asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}$; see footnote 15.

**Likelihood ratio tests** As described more fully in Section A.6.8 of Appendix A, a large-sample likelihood ratio test of a linear (or under some circumstances, non-linear) null hypothesis may be based on the test statistic

$$\begin{aligned} G^2 &= -2\log\left(\frac{L(\widehat{\boldsymbol{\theta}}_0)}{L(\widehat{\boldsymbol{\theta}})}\right) \\ &= 2\left(\ell(\widehat{\boldsymbol{\theta}}) - \ell(\widehat{\boldsymbol{\theta}}_0)\right), \end{aligned} \tag{1.17}$$

where $L(\cdot)$ is the likelihood function, $\ell(\cdot)$ is the log likelihood, $\widehat{\boldsymbol{\theta}}$ is the unrestricted maximum likelihood estimate, and $\widehat{\boldsymbol{\theta}}_0$ is the maximum likelihood estimate restricted by the null hypothesis. The second line says that the test statistic is just the difference between two log likelihoods. If the null hypothesis is true, then the approximate large-sample distribution of $G^2$ is chi-squared with $r$ degrees of freedom, where $r$ is the number of equalities specified by the null hypothesis.

### 1.5.3 Testing model correctness

The typical structural equation model implies a covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ with properties that are not necessarily true of covariance matrices in general. For example, the original and surrogate model for the Blood Pressure example yields the covariance matrix (1.7) on page 150. In this matrix, $\sigma_{13} = \sigma_{14}$, $\sigma_{23} = \sigma_{24}$ and $\sigma_{33} = \sigma_{34}$; these same constraints are implied by the surrogate model. The double measurement regression Model (47) and

[15]For a classical estimate that depends on multivariate normality of the data, one can use the inverse of the estimated Fisher information – either $\boldsymbol{\mathcal{I}}(\widehat{\boldsymbol{\theta}})$ or $\boldsymbol{\mathcal{J}}(\widehat{\boldsymbol{\theta}})$ from Section A.6.6 in Appendix A. Robust estimators like the ones described in Section 5 provide alternatives that do not assume multivariate normality.

the instrumental variables Model (57) also induce equality constraints on their covariance matrices; see pages 87 and 115 respectively for details.

In all such cases, the model implies that certain polynomials in $\sigma_{ij}$ are equal to zero. These constraints are satisfied by $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ for any $\boldsymbol{\theta}$ in the parameter space, including $\widehat{\boldsymbol{\theta}}$. This means that the matrix $\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$ (the reproduced covariance matrix) automatically satisfies the constraints as well.

With probability one, $\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$ will not be exactly equal to $\widehat{\boldsymbol{\Sigma}}$, but if the model is correct it should be fairly close. This is the idea behind Jöreskog's (1967) classical likelihood ratio test for goodness of model fit [35]. The null hypothesis is that the equality constraints implied by the model are true[16], and the alternative is that $\boldsymbol{\Sigma}$ is completely unconstrained except for being symmetric and positive definite. Note that since a well-chosen surrogate model implies the same constraints as the original model, this test of model correctness applies equally to the original and the surrogate model. It is far more convenient to carry out model fitting using the surrogate model.

Assuming that substantive modeling considerations do not restrict the intercept parameter $\boldsymbol{\nu}$ in the general Model (1.1)[17], the likelihood ratio test statistic is written

$$\begin{aligned} G^2 &= -2\log\frac{L\Big(\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\Big)}{L(\widehat{\boldsymbol{\Sigma}})} \\ &= -2\log\frac{|\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})|^{-n/2}(2\pi)^{-nk/2}\exp-\frac{n}{2}\Big\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1})\Big\}}{|\widehat{\boldsymbol{\Sigma}}|^{-n/2}(2\pi)^{-nk/2}\exp-\frac{n}{2}\Big\{tr(\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Sigma}}^{-1})\Big\}} \\ &= n\Big(\log|\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})| + tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}| - k\Big) \\ &= n\Big(tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}| - k\Big) \end{aligned} \tag{1.18}$$

This statistic is quite easy to compute given $\widehat{\boldsymbol{\theta}}$. In fact, it is common for software to directly minimize the "objective function" or "loss function"

$$b(\boldsymbol{\theta}) = tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}) - k - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}| \tag{1.19}$$

instead of the minus log likelihood[18], and then just multiply the final result by $n$ to get the likelihood ratio test statistic $G^2$. An advantage of doing it this way is that the numerical performance of the minimization is not affected by the sample size.

[16] This is not what he says, but it clarifies what he does say.

[17] This might not be a completely safe assumption. For example, if two measurements of a latent variable are truly equivalent, they will have the same means as well as the same variances and the same covariances with other variables. Overlooking this kind of thing would result in a modest loss of power in the goodness of fit test.

[18] If you are a history buff, compare (1.19) to formula (6) on p. 446 in Jöreskog's (1978) classic article [37] in *Psychometrika*. Astonishingly, this is almost the same as Formula (6) (same equation number) on p. 446 (same page number) in [35], another classic article by Jöreskog in *Psychometrika* (Jöreskog, 1967). The 1967 paper is limited to the special case of factor analysis.

The test statistic $G^2$ is referred to a chi-squared distribution with degrees of freedom equal to the number of model-induced equality constraints on $\boldsymbol{\Sigma}$. When $G^2$ is larger than the critical value, the null hypothesis that the constraints hold is rejected, casting doubt on the model.

To count the constraints, first assume that the parameter vector is identifiable, and that there are more moment structure equations than unknown parameters. If the number of parameters is equal to the number of moment structure equations, the model is called *saturated*, and this way of testing model fit does not work.

Suppose there are $m$ moments (typically covariances or correlations), and $r$ unknown parameters in the vector $\boldsymbol{\theta}$, with $m > r$. The degrees of freedom are $m - r$. To see why this might hold, suppose that exactly $r$ of the the moment structure equations can be solved for the $r$ unknown parameters. Substituting the solution into the $m - r$ unused equations gives $m - r$ equalities involving only $\sigma_{ij}$ quantities. These correspond to the constraints. Notice that while this is a test of the constraints that the model induces on the covariance matrix $\boldsymbol{\Sigma}$, the test statistic can be calculated and degrees of freedom can be determined without knowing exactly what the constraints are.

If a model fails the $G^2$ goodness of fit test, it is common to search for a model that does fit. Sometimes, the reason for lack of fit can be revealed by *residuals* formed by subtracting the elements of $\widehat{\boldsymbol{\Sigma}}$ from those of $\Sigma(\boldsymbol{\theta})$. Approximate formulas for standardization are available. Once the model fits, likelihood ratio tests for full versus reduced models can be obtained by subtracting $G^2$ statistics, with degrees of freedom equal to the number of additional constraints implied by the reduced model.

The likelihood ratio test for goodness of fit is useful, but as a test of model correctness it is incomplete. This is because structural equation models imply two types of constraint on $\boldsymbol{\Sigma}$: equality constraints and inequality constraints. For example, in proving identifiability for the instrumental variables Model (57) on page 112, the solution (61) includes $\omega = \sigma_{11} - \frac{\sigma_{13}\sigma_{14}}{\sigma_{34}}$. Because $\omega$ is a variance, this means $\sigma_{11} > \frac{\sigma_{13}\sigma_{14}}{\sigma_{34}} \implies \sigma_{11}\sigma_{34} > \sigma_{13}\sigma_{14}$, an inequality constraint that is obviously not true of $4 \times 4$ covariance matrices in general. The typical structural equation model imposes many inequality constraints on the covariance matrix.

In general, moment structure equations map the parameter space into a *moment space*, which for the classical surrogate models is a space of $k \times k$ positive definite matrices. As the numerical maximum likelihood search moves $\boldsymbol{\theta}$ through the parameter space, $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ moves along through a lower-dimensional subset of the moment space where the equality constraints are satisfied, generally behaving as if it were attracted to $\widehat{\boldsymbol{\Sigma}}$.

While $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is forced to obey the equality constraints, it need not obey the inequality constraints. If the true value of $\boldsymbol{\Sigma}$ is such that an inequality constraint is not satisfied (which means the model is wrong), then it is quite possible for $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ to cross the boundary of an inequality constraint. This means that $\boldsymbol{\theta}$ leaves the parameter space. Maximum likelihood estimates that are outside the parameter space make everyone uncomfortable, if they are noticed. In factor analysis, this phenomenon is called a "Heywood case;" see page 226.

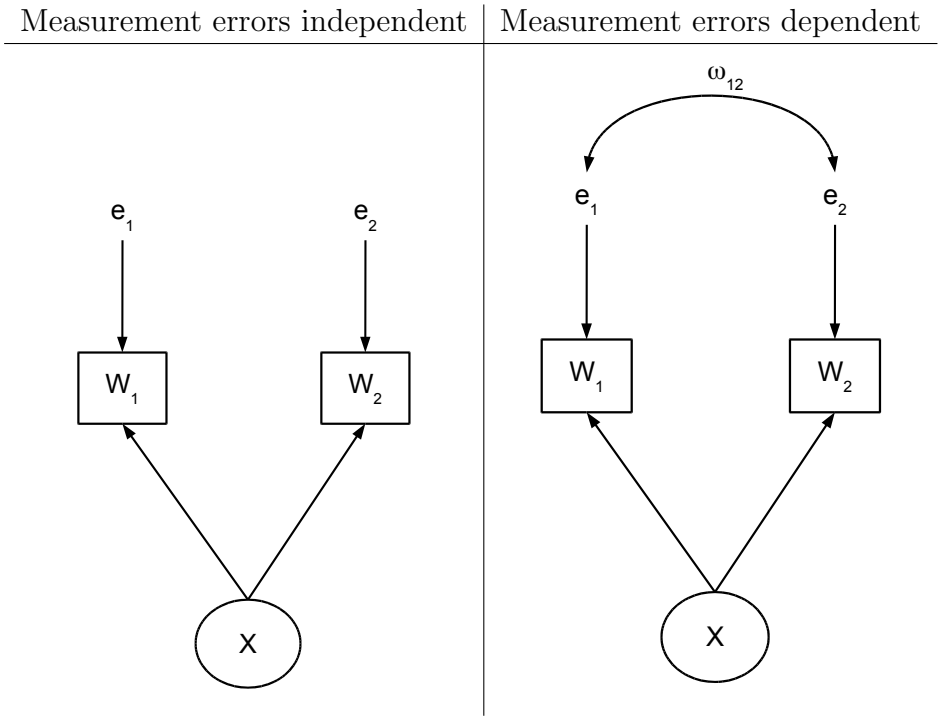
**Example 1.5.1** *A negative variance estimate*

Here is a very simple example. Suppose we have two measurements of a latent variable, like academic ability. The surrogate model equations are, independently for $i = 1, \ldots, n$,

$$\begin{aligned} W_{i,1} &= X_i + e_{i,1} \\ W_{i,2} &= X_i + e_{i,2}, \end{aligned}$$

where all expected values are zero, $Var(X_i) = \phi$, $Var(e_{i,1}) = \omega_1$, and $Var(e_{i,2}) = \omega_2$. According to the model, the exogenous variables $e_{i,1}$, $e_{i,2}$ and $X_i$ are all independent. A path diagram is shown in the left panel of Figure 1.7. The covariance matrix of the

Figure 1.7: Two measurements of a latent variable




observable variables $(W_{i,1}, W_{i,2})^\top$ is

$$\begin{pmatrix} \omega_1 + \phi & \phi \\ \phi & \omega_2 + \phi \end{pmatrix} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix}.$$

The model is saturated, with three linear covariance structure equations in three unknown parameters. The solutions are

$$\begin{aligned} \phi &= \sigma_{12} \\ \omega_1 &= \sigma_{11} - \sigma_{12} \\ \omega_2 &= \sigma_{22} - \sigma_{12}, \end{aligned} \tag{1.20}$$

so that the parameters are just identifiable. The model imposes no equality constraints on $\boldsymbol{\Sigma}$, and it is untestable with the classical test of fit. However, since the model parameters are all variances, the equations (1.20) reveal three inequality constraints: $\sigma_{12} > 0$, $\sigma_{11} > \sigma_{12}$ and $\sigma_{22} > \sigma_{12}$.

By the invariance principle, explicit formulas for the maximum likelihood estimates $\widehat{\phi}$, $\widehat{\omega}_1$ and $\widehat{\omega}_2$ are obtained by simply putting hats on the Greek letters in (1.20). To see what could go wrong, suppose that the observable variables $W_{i,1}$ and $W_{i,2}$ have other, unmeasured common influences in addition to $X_i$, like test anxiety or something. As discussed in Section 0.4 on omitted variables in regression, the result would be a positive covariance between $e_{i,1}$ and $e_{i,2}$. We will denote $cov(e_{i,1}, e_{i,2})$ by $\omega_{12}$. The resulting path diagram is shown in the right panel of Figure 1.7. The covariance matrix of the observable variables is now

$$\begin{pmatrix} \omega_1 + \phi & \phi + \omega_{12} \\ \phi + \omega_{12} & \omega_2 + \phi \end{pmatrix} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix}.$$

This second model could well be more realistic than the first, even though the parameters are not identifiable. There is no doubt that it's easier to assume zero covariance between error terms than to guarantee it in practice.

Let's say that the second model is correct, but we fit the first model anyway. The model we are fitting says that $\sigma_{12} = \phi$, when in fact $\sigma_{12} = \phi + \omega_{12}$. Assuming the incorrect model, the maximum likelihood estimate of $\omega_1$ is $\widehat{\omega}_1 = \widehat{\sigma}_{11} - \widehat{\sigma}_{12}$. But under the correct model,

$$\begin{aligned} \widehat{\omega}_1 &= \widehat{\sigma}_{11} - \widehat{\sigma}_{12} \\ &\stackrel{a.s.}{\to} \sigma_{11} - \sigma_{12} \\ &= (\omega_1 + \phi) - (\phi + \omega_{12}) \\ &= \omega_1 - \omega_{12}. \end{aligned}$$

Recall that $\omega_1 = Var(e_{i,1})$. For the estimate of this variance to be negative for large samples, all that's required is $\omega_{12} > \omega_1$. Is this possible (while keeping the covariance matrix of $(e_{i,1}, e_{i,2})^\top$ positive definite)? Most assuredly. Here's a numerical example.

$$\begin{pmatrix} \omega_1 & \omega_{12} \\ \omega_{12} & \omega_2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}.$$

The point here is that structural equation models imply inequality constraints on the elements of $\boldsymbol{\Sigma}$, the covariance matrix of the observable variables. Model incorrectness can result in violation of these constraints, and cause numerical maximum likelihood to leave the parameter space. This is a valuable way to diagnose problems with the model. Of course negative variance estimates are easiest to notice. Chapter 7 treats model diagnostics in more detail.

## 1.6 The Brand Awareness Study Re-visted

We return to the Brand Awareness Example 1.2, given in Section 1.2. A major Canadian coffee shop chain is trying to break into the U.S. Market. They assess the following vari-

ables twice on a random sample of coffee-drinking adults. The two measurements of each variable are conducted at different times by different interviewers asking somewhat different questions, in such a way that the errors of measurement may be assumed independent. The latent variables are

$X_1$: Brand Awareness: True familiarity with the coffee shop chain.

$X_2$: Advertising Awareness: Recall for advertising of the coffee shop chain.

$X_3$: True interest in the product category: Mostly this is how much they really like doughnuts.

$Y_1$: Purchase Intention: True willingness to go to an outlet of the coffeeshop chain and make an order.

$Y_2$: Purchase behaviour: True number of dollars spent at the chain during the 2 months following the interview.

There are two observed versions of each latent variable, all based on self-report. All observed variables were measured on a scale from 0 to 100 except purchase behaviour, which is in dollars.

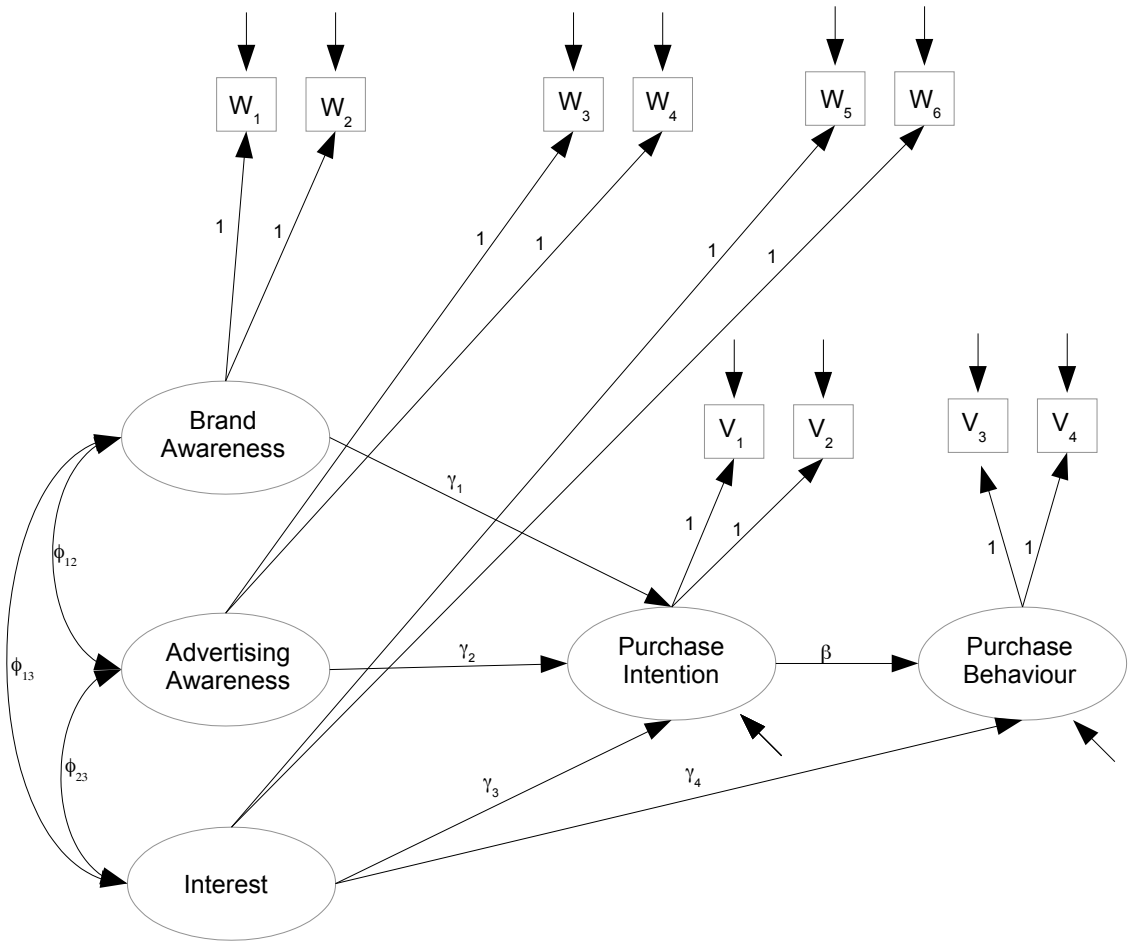
Figure 1.8 shows the path diagram for a surrogate model. It is more detailed than Figure 1.1 on page 140, in that symbols are indicted on the arrows. You can tell it's a surrogate model because of the symbol "1" on the arrows linking latent to observed variables. The model asserts that all measurement here is double measurement.

The model equations in (1.2) on page 141 are the equations of the original model. The equations of the centered surrogate model corresponding to Figure 1.8 are

$$\begin{aligned}
Y_{i,1} &= \gamma_1 X_{i,1} + \gamma_2 X_{i,2} + \gamma_3 X_{i,3} + \epsilon_{i,1} \\
Y_{i,2} &= \beta Y_{i,1} + \gamma_4 X_{i,3} + \epsilon_{i,2} \\
W_{i,1} &= X_{i,1} + e_{i,1} \\
W_{i,2} &= X_{i,1} + e_{i,2} \\
W_{i,3} &= X_{i,2} + e_{i,3} \\
W_{i,4} &= X_{i,2} + e_{i,4} \\
W_{i,5} &= X_{i,3} + e_{i,5} \\
W_{i,6} &= X_{i,3} + e_{i,6} \\
V_{i,1} &= Y_{i,1} + e_{i,7} \\
V_{i,2} &= Y_{i,1} + e_{i,8} \\
V_{i,3} &= Y_{i,2} + e_{i,9} \\
V_{i,4} &= Y_{i,2} + e_{i,10},
\end{aligned} \tag{1.21}$$

where all expected values equal zero, $Var(X_{i,j}) = \phi_{jj}$ for $j = 1, 2, 3$, $Cov(X_{i,j}, X_{i,k}) = \phi_{jk}$, $Var(e_{i,j}) = \omega_j$ for $j = 1, \ldots, 10$, $Var(\epsilon_{i,1}) = \psi_1$, $Var(\epsilon_{i,2}) = \psi_2$. All the error terms are independent of one another and of the $X_{i,j}$ variables.

Figure 1.8: Brand Awareness Model One




Before fitting any structural equation model, one should verify that the parameters are identifiable. Later chapters this text develop a set of standard rules that would allow us to do the check by just examining the path diagram in Figure 1.8. These rules are summarized in (someplace; I have not written it yet). For now, we will do the job from first principles.

The general two-stage model of Section 1.2 is designed to facilitate two-stage proofs of identifiability. Disregarding intercepts and expected values as usual and assuming other details in the model specification (1.1),

- The measurement model is $\mathbf{d}_i = \mathbf{\Lambda F}_i + \mathbf{e}_i$, with $cov(\mathbf{F}_i) = \mathbf{\Phi}$ and $cov(\mathbf{e}_i) = \mathbf{\Omega}$.
- The latent variable model is $\mathbf{y}_i = \boldsymbol{\beta}\mathbf{y}_i + \mathbf{\Gamma x}_i + \boldsymbol{\epsilon}_i$, with $cov(\mathbf{x}_i) = \mathbf{\Phi}_x$ and $cov(\epsilon_i) = \mathbf{\Psi}$.
- The models are linked by $\mathbf{F}_i = \begin{pmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{pmatrix}$.

Denoting the common covariance matrix of the data vectors by $cov(\mathbf{d}_i) = \boldsymbol{\Sigma}$, the task is to show that all the Greek-letter model parameters can be recovered from $\boldsymbol{\Sigma}$. The two-stage strategy is

1. Referring to the measurement model, write $\boldsymbol{\Sigma}$ as a function of the parameter matrices $\boldsymbol{\Lambda}$, $\boldsymbol{\Phi}$ and $\boldsymbol{\Omega}$. Then solve for $\boldsymbol{\Lambda}$, $\boldsymbol{\Phi}$ and $\boldsymbol{\Omega}$ in terms of $\boldsymbol{\Sigma}$, showing they are identifiable.
2. Referring to the latent variable model, write $\boldsymbol{\Phi} = cov(\mathbf{F}_i)$ as a function of $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\Phi}_x$ and $\boldsymbol{\Psi}$. Then solve for $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\Phi}_x$ and $\boldsymbol{\Psi}$ in terms of $\boldsymbol{\Phi}$. Since $\boldsymbol{\Phi}$ is already shown to be a function of $\boldsymbol{\Sigma}$ in the first stage, this means that the latent variable model parameters are also functions of $\boldsymbol{\Sigma}$, and they are identified.

**Double Measurement** For the brand awareness example, the measurement part of the model is a special case of the measurement model for double measurement regression in section 0.10.3 of Chapter 0. The measurements come in two independent sets, which may be denoted $\mathbf{d}_{i,1}$ and $\mathbf{d}_{i,2}$. The full set of observable data is the partitioned random vector

$$\mathbf{d}_i = \begin{pmatrix} \mathbf{d}_{i,1} \\ \hline \mathbf{d}_{i,2} \end{pmatrix}, \text{ where } \mathbf{d}_{i,1} = \begin{pmatrix} W_{i,1} \\ W_{i,3} \\ W_{i,5} \\ V_{i,1} \\ V_{i,3} \end{pmatrix} \text{ and } \mathbf{d}_{i,2} = \begin{pmatrix} W_{i,2} \\ W_{i,4} \\ W_{i,6} \\ V_{i,2} \\ V_{i,4} \end{pmatrix}.$$

The double measurement model equations are

$$\begin{aligned} \mathbf{d}_{i,1} &= \mathbf{F}_i + \mathbf{e}_{i,1} \\ \mathbf{d}_{i,2} &= \mathbf{F}_i + \mathbf{e}_{i,2}, \end{aligned} \tag{1.22}$$

where the vector of latent variables $\mathbf{F}_i$ has zero covariance with $\mathbf{e}_{i,1}$ and $\mathbf{e}_{i,2}$, $cov(\mathbf{e}_{i,1}) = \boldsymbol{\Omega}_1$, $cov(\mathbf{e}_{i,2}) = \boldsymbol{\Omega}_2$ and $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) = \mathbf{O}$. Thus we have a partitioned covariance matrix for the measurement errors:

$$cov(\mathbf{d}_i) = \boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{\Omega}_1 & \mathbf{O} \\ \mathbf{O} & \boldsymbol{\Omega}_2 \end{pmatrix}.$$

For the model of Figure 1.8, the matrices $\boldsymbol{\Omega}_1$ and $\boldsymbol{\Omega}_2$ happen to be diagonal, but what's important is independence of measurement errors between sets, not within.

Using the notation $\boldsymbol{\Sigma}_{1,1} = cov(\mathbf{d}_{i,1})$, $\boldsymbol{\Sigma}_{2,2} = cov(\mathbf{d}_{i,1})$ and $\boldsymbol{\Sigma}_{1,2} = cov(\mathbf{d}_{i,1}, \mathbf{d}_{i,2})$ (so that $\boldsymbol{\Sigma}$ is also a partitioned matrix), we have

$$\begin{aligned} \boldsymbol{\Sigma}_{1,1} &= \boldsymbol{\Phi} + \boldsymbol{\Omega}_1 \\ \boldsymbol{\Sigma}_{2,2} &= \boldsymbol{\Phi} + \boldsymbol{\Omega}_2 \\ \boldsymbol{\Sigma}_{1,2} &= \boldsymbol{\Phi}, \end{aligned}$$

Solving for the parameter matrices is immediate, yielding

$$\begin{aligned} \boldsymbol{\Phi} &= \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Omega}_1 &= \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Omega}_2 &= \boldsymbol{\Sigma}_{2,2} - \boldsymbol{\Sigma}_{1,2}. \end{aligned} \tag{1.23}$$

That establishes identifiability for the double measurement model in general, including this particular model for the brand awareness data. Identifiability of the double measurement model is so useful that it will be documented as a formal parameter identifiability rule.

**Rule 2a:** The Double Measurement Rule. *The parameters of the double measurement model (1.22) are identifiable. There are two sets of measurements. Each latent variable is measured twice, and all factor loadings equal one. Measurement errors may be correlated within sets, but not between sets.*

For the current Brand Awareness model, the double measurement rule establishes stage one of the two-stage proof. In the second stage, we recover the parameters of the latent variable model from $\boldsymbol{\Phi}$, which has already been identified. First of all, $\boldsymbol{\Phi}_x$, the covariance matrix of the latent exogenous variables $(X_{i,1}, X_{i,2}, X_{i,3})^\top$, is part of $\boldsymbol{\Phi}$ – so it's identified. Then, look at the first equation in (1.21), or at the path diagram. It's just a regression, so by (16) on page 26, all the parameters are identifiable from the covariance matrix of $(X_{i,1}, X_{i,2}, X_{i,3}, Y_{i,1})^\top$. That is, we have identified $\gamma_1, \gamma_2, \gamma_3$ and $\psi_1$. The second line of (1.21) is also just a regression, and the parameters $\gamma_4, \beta$ and $\psi_2$ are identified from the covariance matrix of the variables involved. This completes the second stage. All the parameters in the model are identifiable.

We proceed to fit the model with `lavaan`. Familiarity with the material in section 0.10.2 starting on page 66 is assumed. The R job begins by loading `lavaan`, and then reading and documenting the data.

```
> # Brand awareness
>
> rm(list=ls()); options(scipen=999)
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
> coffee = read.table("http://www.utstat.toronto.edu/~brunner/openSEM/data/timmy1.data.txt")
> head(coffee)
  w1 w2 w3 w4 w5 w6 v1 v2 v3 v4
1 40 23 26 21 48 38 22 22 15 15
2 45 24 29 23 49 48 26 13  8 13
3 29 21 21 13 42 37 18 12 13 13
4 38 26 18 19 47 42 20  9 12 10
5 47 31 30 18 48 52 26 16 22 16
6 31 24 18 13 39 40 20 12 16 18
>
> # Observed variables
> #   w1 = Brand Awareness 1
> #   w2 = Brand Awareness 2
> #   w3 = Ad Awareness 1
> #   w4 = Ad Awareness 2
```

```
> #   w5 = Interest 1
> #   w6 = Interest 2
> #   v1 = Purchase Intention 1
> #   v2 = Purchase Intention 2
> #   v3 = Purchase Behaviour 1
> #   v4 = Purchase Behaviour 2
> # Latent variables
> #   L_BrAw  = True brand awareness
> #   L_AdAw  = True advertising awareness
> #   L_Inter = True interest in the product category
> #   L_PI    = True purchase intention
> #   L_PBeh  = True purchase behaviour
```

Next, we define and fit the model. `lavaan` returns the R prompt without any complaints or warnings.

```
> torus1 =
+ '
+ # Latent variable model
+     L_PI ~ gamma1*L_BrAw + gamma2*L_AdAw + gamma3*L_Inter
+     L_PBeh    ~ gamma4*L_Inter + beta*L_PI
+ # Measurement model (simple double measurement)
+     L_BrAw  =~ 1*w1 + 1*w2
+     L_AdAw  =~ 1*w3 + 1*w4
+     L_Inter =~ 1*w5 + 1*w6
+     L_PI    =~ 1*v1 + 1*v2
+     L_PBeh  =~ 1*v3 + 1*v4
+ # Variances  and covariances
+     # Exogenous latent variables
+         L_BrAw ~~ phi11*L_BrAw   # Var(L_BrAw)         = phi11
+         L_BrAw ~~ phi12*L_AdAw   # Cov(L_BrAw,L_AdAw)  = phi12
+         L_BrAw ~~ phi13*L_Inter  # Cov(L_BrAw,L_Inter) = phi13
+         L_AdAw ~~ phi22*L_AdAw   # Var(L_AdAw)         = phi22
+         L_AdAw ~~ phi23*L_Inter  # Cov(L_AdAw,L_Inter) = phi23
+         L_Inter ~~ phi33*L_Inter # Var(L_Inter)        = phi33
+     # Errors in the latent model (epsilons)
+         L_PI ~~ psi1*L_PI        # Var(epsilon1) = psi1
+         L_PBeh ~~ psi2*L_PBeh    # Var(epsilon2) = psi2
+     # Measurement errors
+         w1 ~~ omega1*w1          # Var(e1)  = omega1
+         w2 ~~ omega2*w2          # Var(e2)  = omega2
+         w3 ~~ omega3*w3          # Var(e3)  = omega3
+         w4 ~~ omega4*w4          # Var(e4)  = omega4
+         w5 ~~ omega5*w5          # Var(e5)  = omega5
+         w6 ~~ omega6*w6          # Var(e6)  = omega6
+         v1 ~~ omega7*v1          # Var(e7)  = omega7
+         v2 ~~ omega8*v2          # Var(e8)  = omega8
```

```
+         v3 ~~ omega9*v3          # Var(e9)  = omega9
+         v4 ~~ omega10*v4         # Var(e10) = omega10
+ # Bounds (Variances are positive)
+     phi11 > 0; phi22 > 0; phi33 > 0
+     psi1 > 0; psi2 > 0
+     omega1 > 0; omega2 > 0; omega3 > 0; omega4 > 0; omega5 > 0
+     omega6 > 0; omega7 > 0; omega8 > 0; omega9 > 0; omega10 > 0
+ ' # End of model torus1
>
> fit1 = lavaan(torus1, data=coffee)
>
```

Looking just at the fit of the model,

```
> show(fit1)
lavaan 0.6-7 ended normally after 113 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         23
  Number of inequality constraints                  15

  Number of observations                           200

Model Test User Model:

  Test statistic                                77.752
  Degrees of freedom                                32
  P-value (Chi-square)                           0.000
```

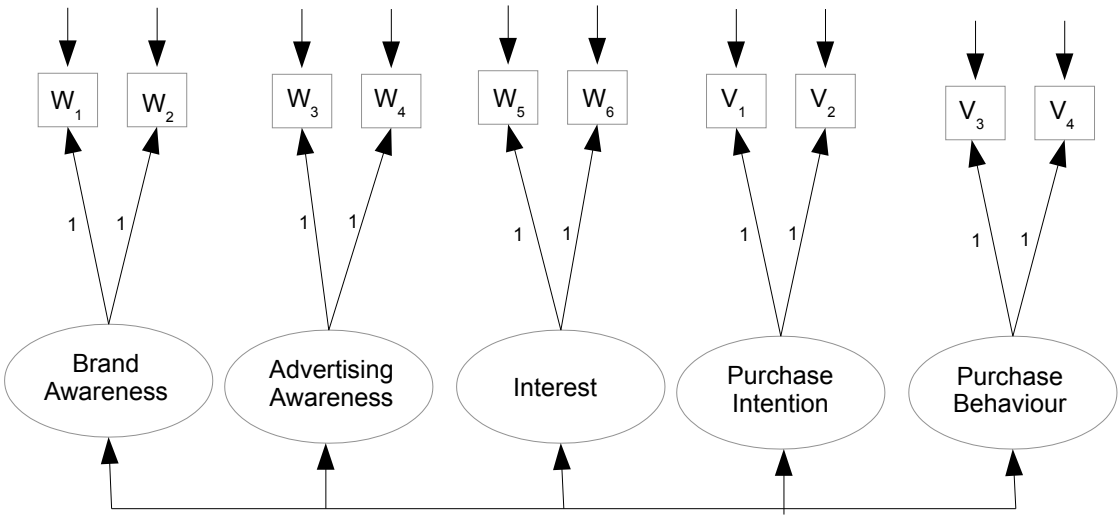
By the likelihood ratio test, the model does not fit[19]. A close look at the output of `summary` and `partable` reveals nothing out of the ordinary. We need determine why the model did not fit, and fix it if possible. To do this, a divide and conquer strategy can be helpful. We'll split the problem into parts, and look first at the measurement model. Figure 1.9 shows a model in which the structure in the latent variable model is discarded, and the measurement model is preserved. Note the shorthand way of expressing all possible covariances among the latent variables. By the first stage of the two-stage proof of identifiability, all the parameters of this model are identifiable.

The model is fully specified in the model string `torus2`. It's very explicit, but naming all the variances and covariances makes it tedious to type.

```
> torus2 =
+ '
```

[19] In this example, I follow my usual practice of relying on the likelihood ratio test to determine whether a model fits adequately. This choice is not very popular among practitioners of structural equation modelling, because standard models so often fail the test when applied to real data. See Chapter 7.

Figure 1.9: Brand Awareness Model Two




```
+ # Measurement model (still simple double measurement)
+     L_BrAw  =~ 1*w1 + 1*w2
+     L_AdAw  =~ 1*w3 + 1*w4
+     L_Inter =~ 1*w5 + 1*w6
+     L_PI    =~ 1*v1 + 1*v2
+     L_PBeh  =~ 1*v3 + 1*v4
+ # Variances and covariances
+     # Latent variables
+         L_BrAw ~~ phi11*L_BrAw      # Var(L_BrAw)          = phi11
+         L_BrAw ~~ phi12*L_AdAw      # Cov(L_BrAw, L_AdAw)  = phi12
+         L_BrAw ~~ phi13*L_Inter     # Cov(L_BrAw, L_Inter) = phi13
+         L_BrAw ~~ phi14*L_PI        # Cov(L_BrAw, L_PI)    = phi14
+         L_BrAw ~~ phi15*L_PBeh      # Cov(L_BrAw, L_PBeh)  = phi15
+
+         L_AdAw ~~ phi22*L_AdAw      # Var(L_AdAw)          = phi22
+         L_AdAw ~~ phi23*L_Inter     # Cov(L_AdAw, L_Inter) = phi23
+         L_AdAw ~~ phi24*L_PI        # Cov(L_AdAw, L_PI)    = phi24
+         L_AdAw ~~ phi25*L_PBeh      # Cov(L_AdAw, L_PBeh)  = phi25
+
+         L_Inter ~~ phi33*L_Inter    # Var(L_Inter)         = phi33
+         L_Inter ~~ phi34*L_PI       # Cov(L_Inter, L_PI)   = phi34
+         L_Inter ~~ phi35*L_PBeh     # Cov(L_Inter, L_PBeh) = phi35
+
+         L_PI ~~ phi44*L_PI          # Var(L_PI)            = phi44
+         L_PI ~~ phi45*L_PBeh        # Cov(L_PI, L_PBeh)    = phi45
+
```

```
+         L_PBeh ~~ phi55*L_PBeh      # Var(L_PBeh)          = phi55
+     # Measurement errors
+         w1 ~~ omega1*w1           # Var(e1)  = omega1
+         w2 ~~ omega2*w2           # Var(e2)  = omega2
+         w3 ~~ omega3*w3           # Var(e3)  = omega3
+         w4 ~~ omega4*w4           # Var(e4)  = omega4
+         w5 ~~ omega5*w5           # Var(e5)  = omega5
+         w6 ~~ omega6*w6           # Var(e6)  = omega6
+         v1 ~~ omega7*v1           # Var(e7)  = omega7
+         v2 ~~ omega8*v2           # Var(e8)  = omega8
+         v3 ~~ omega9*v3           # Var(e9)  = omega9
+         v4 ~~ omega10*v4          # Var(e10) = omega10
+ # Bounds (Variances are positive)
+     phi11 > 0; phi22 > 0; phi33 > 0; phi44 > 0; phi55 > 0
+     omega1 > 0; omega2 > 0; omega3 > 0; omega4 > 0; omega5 > 0
+     omega6 > 0; omega7 > 0; omega8 > 0; omega9 > 0; omega10 > 0
+ ' # End of model torus2
>
> fit2 = lavaan(torus2, data=coffee)
```

There has to be a better way, and there is. In the model `torus2b`, only the measurement model is specified.

```
> torus2b =
+ '
+ # Measurement model (still simple double measurement)
+     L_BrAw  =~ 1*w1 + 1*w2
+     L_AdAw  =~ 1*w3 + 1*w4
+     L_Inter =~ 1*w5 + 1*w6
+     L_PI    =~ 1*v1 + 1*v2
+     L_PBeh  =~ 1*v3 + 1*v4
+ # Leave off everything else and see what happens.
+ ' # End of model torus2b
```

The `lavaan` function chokes on this, because it requires more detail. However, the `cfa` function (for confirmatory factor analysis – see Chapter 3) assumes by default that all the latent variables have non-zero covariances, and does not require the user to name them[20].

```
> fit2b = cfa(torus2b, data=coffee)
```

That's a lot better. The models `torus2` and `torus2b` are 100% equivalent, except that the parameters in `torus2` have labels. The fit (that is, lack of fit) is identical.

[20] Actually, the `lavaan` function will name your parameters for you too. Syntax like `L_PI ~ gamma1*L_BrAw + gamma2*L_AdAw + gamma3*L_Inter` looks like you are transcribing a model equation, but technically those Greek letter names are just optional labels for the regression parameters, which have their own internal names.

```
> show(fit2)
lavaan 0.6-7 ended normally after 124 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         25
  Number of inequality constraints                  15

  Number of observations                           200

Model Test User Model:

  Test statistic                                76.380
  Degrees of freedom                                30
  P-value (Chi-square)                           0.000
> show(fit2b)
lavaan 0.6-7 ended normally after 139 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         25

  Number of observations                           200

Model Test User Model:

  Test statistic                                76.380
  Degrees of freedom                                30
  P-value (Chi-square)                           0.000
```

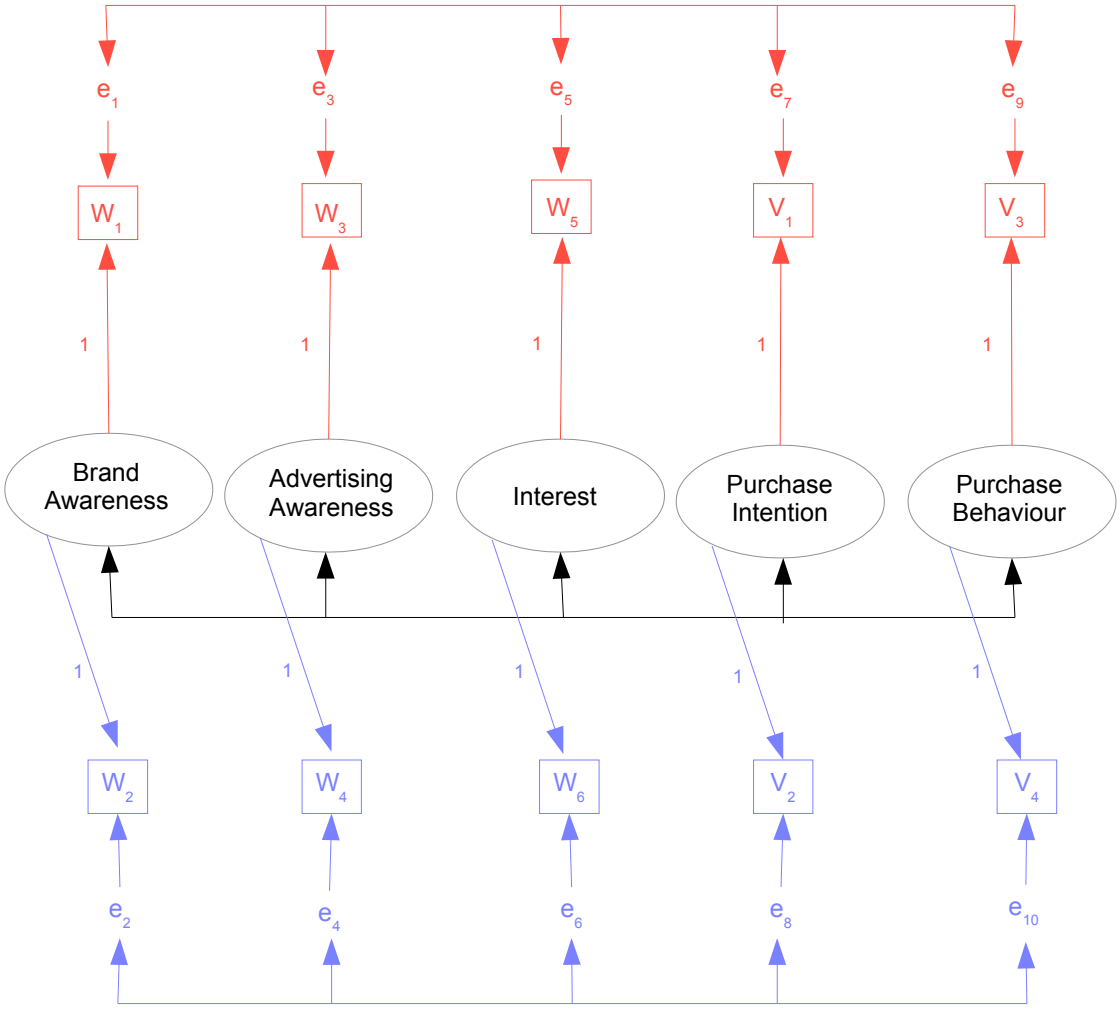
The measurement model does not fit[21], and we need to fix it. Now, the model asserts a kind of double measurement, but it's a restricted kind in which all the measurement errors are all independent. Maybe independence does not hold, and that's causing the lack of fit.

In the proof of identifiability for this example, the measurement model had two sets of measurements, with errors of measurement potentially correlated within sets but not between sets. The proposal here is just to put in the non-zero covariances between sets, so identifiability has already been established. Figure 1.10 shows the resulting model. Measurement set one is red, and measurement set two is blue.

In the model string `torus3`, the non-zero covariances among measurement error terms

[21] It's a bit tempting to observe that the difference between the models `torus1` and `torus2` is that `torus1` imposes some structure in the relationships among the latent variables. In fact, it can be shown that the *only* difference between the two models is the lack of some arrows in `torus1`. So it would seem that one could test the difference between the two models with a likelihood ratio test, and thereby assess the fit of the latent variable model. That's not a good idea, though. When a full model does not fit the data, testing for difference between full and restricted models can be very misleading.

Figure 1.10: Brand Awareness Model Three




are specified without explicitly naming the parameters. This saves a fair amount of typing.

```
> torus3 =
+ '
+ # Measurement model (still simple double measurement)
+     L_BrAw  =~ 1*w1 + 1*w2
+     L_AdAw  =~ 1*w3 + 1*w4
+     L_Inter =~ 1*w5 + 1*w6
+     L_PI    =~ 1*v1 + 1*v2
+     L_PBeh  =~ 1*v3 + 1*v4
+ # Add covariances between measurement error terms, without naming them
+     w1 ~~ w3; w1 ~~ w5; w1 ~~ v1; w1 ~~ v3
+               w3 ~~ w5; w3 ~~ v1; w3 ~~ v3
```

```
+                          w5 ~~ v1; w5 ~~ v3
+                                    v1 ~~ v3
+     w2 ~~ w4; w2 ~~ w6; w2 ~~ v2; w2 ~~ v4
+                w4 ~~ w6; w4 ~~ v2; w4 ~~ v4
+                          w6 ~~ v2; w6 ~~ v4
+                                    v2 ~~ v4
+ ' # End of model torus3
```

When we try to fit this nice model, there is trouble.

```
> fit3 = cfa(torus3, data=coffee)
Warning message:
In lav_object_post_check(object) :
  lavaan WARNING: the covariance matrix of the residuals of the observed
                variables (theta) is not positive definite;
                use lavInspect(fit, "theta") to investigate.
```

The phrase "residuals of the observed variables" refers to the measurement error terms. These are denoted by $e_{i,1}, \ldots, e_{i,10}$ in (1.21). Presumably they are called "residuals" because of the analogy between residuals and error terms in regression. Following the suggestion to try `lavInspect`,

```
> lavInspect(fit3, "theta")
   w1     w2     w3     w4     w5     w6     v1     v2     v3     v4
w1 10.617
w2  0.000 10.477
w3  2.700  0.000 11.704
w4  0.000 -1.726  0.000 11.263
w5  1.246  0.000  0.475  0.000  8.786
w6  0.000 -3.239  0.000 -1.904  0.000  5.053
v1  3.208  0.000  2.999  0.000  3.933  0.000 13.013
v2  0.000 -2.484  0.000 -1.490  0.000 -3.382  0.000  6.854
v3  0.555  0.000 -0.485  0.000  1.049  0.000  0.875  0.000  4.699
v4  0.000 -1.408  0.000 -1.756  0.000 -0.663  0.000 -1.499  0.000  3.911
```

Note how the covariances between even-numbered variables and odd-numbered variables are all zero. This is definitely the estimated covariance matrix of $(e_{i,1}, \ldots, e_{i,10})^\top$. An application of `eigen(lavInspect(fit3, "theta"))$values` reveals one negative eigenvalue, so the matrix is not positive definite, and the numerical search for the MLE has left the parameter space. It is nice that `lavaan` checks for this.

It is possible that the numerical search left the parameter space because the model is wrong, but it's also possible that the problem was caused by sub-optimal starting values. Method-of-moments estimates make excellent starting values. As usual, if identifiability has been established by obtaining explicit solutions to the covariance structure equations, then putting hats on the solutions yields method-of-moments estimates. Using the solution (1.23), estimates for the brand awareness data are calculated as follows.

```
> # Checking why torus3 left the parameter space.
> # Obtain MOM estimates for use as starting values.
>
> d1 = as.matrix(coffee[,c(1,3,5,7,9)]) # Measurement set one
> d2 = as.matrix(coffee[,c(2,4,6,8,10)]) # Measurement set two
> Phi_hat = cov(d1,d2); Phi_hat
          w2       w4        w6        v2        v4
w1 10.186131 6.670427 15.123116 11.928618  8.162688
w3  6.655075 8.684598 12.766332 11.339975  6.893844
w5  7.627940 6.536859 16.409548 10.881683  6.290829
v1  8.347940 7.563392 16.891960 15.024598 10.119975
v3  4.674573 3.738015  7.650754  6.998216 17.746859
```

This matrix isn't symmetric, so it's not in the parameter space. That's easy to fix.

```
> # Make it symmetric
> Phi_hat = (Phi_hat + t(Phi_hat) )/2; Phi_hat
          w2       w4        w6        v2        v4
w1 10.186131 6.662751 11.375528 10.138279  6.418631
w3  6.662751 8.684598  9.651595  9.451683  5.315930
w5 11.375528 9.651595 16.409548 13.886822  6.970791
v1 10.138279 9.451683 13.886822 15.024598  8.559095
v3  6.418631 5.315930  6.970791  8.559095 17.746859
> eigen(Phi_hat)$values # Is it positive definite?
[1] 50.164191 12.097980  2.925981  1.668071  1.195511
```

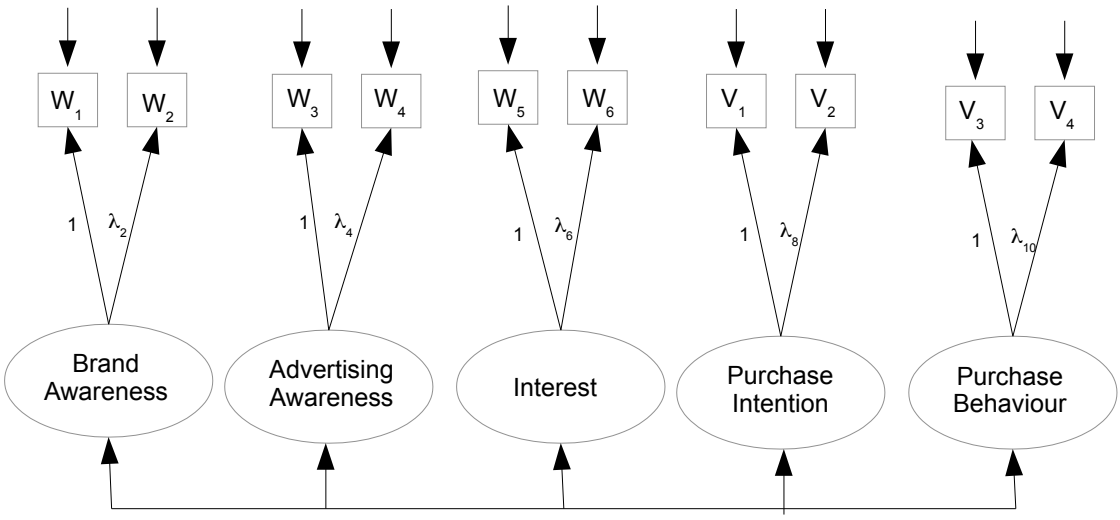
So $\widehat{\mathbf{\Phi}}$ is okay. Computing and testing the estimated covariance matrices of the error terms,

```
> Omega1_hat = cov(d1) - Phi_hat
> Omega2_hat = cov(d2) - Phi_hat
> eigen(Omega1_hat)$values # Is Omega1_hat positive definite?
[1] 26.402687  9.301147  8.288868  5.106178  2.868356
> eigen(Omega2_hat)$values # Is Omega2_hat positive definite?
[1] 12.867799 11.828405  9.847771  4.712254 -3.393667
```

The method-of-moments estimate $\widehat{\mathbf{\Omega}}_2$ is not positive definite. If we used it as a source of starting values, we would be starting the numerical search for the MLE outside of the parameter space. This is not going to be helpful. My conclusion is that this model is incompatible with the data, and it's time to consider another one.

Recall that the two measurements of each latent variable are *different*. One of the interviews is in-person, and the other is by telephone call-back. Maybe they're not really equivalent. Perhaps one in each set (say number two, the call-backs) should have a coefficient not equal to one. Figure 1.11 illustrates the model. We are back to independent error terms for the present. Proof of identifiability is deferred until (one of those two-variable rules).
Fitting the model,

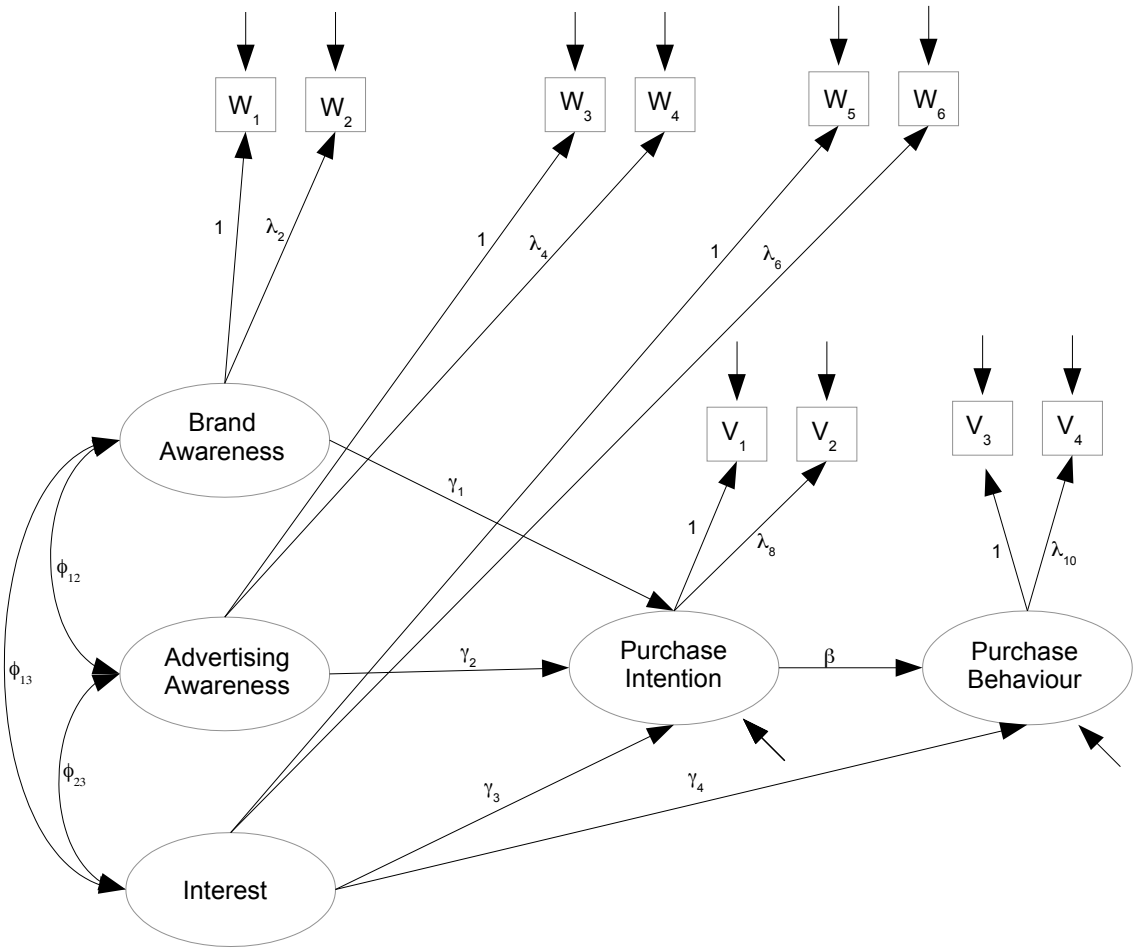
Figure 1.11: Brand Awareness Model Four




```
> torus4 =
+ '
+ # Measurement model (still simple double measurement)
+     L_BrAw  =~ 1*w1 + lambda2*w2
+     L_AdAw  =~ 1*w3 + lambda4*w4
+     L_Inter =~ 1*w5 + lambda6*w6
+     L_PI    =~ 1*v1 + lambda8*v2
+     L_PBeh  =~ 1*v3 + lambda10*v4
+ ' # End of model torus4
> fit4 = cfa(torus4, data=coffee)
> show(fit4)
lavaan 0.6-7 ended normally after 161 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         30

  Number of observations                           200

Model Test User Model:

  Test statistic                                17.837
  Degrees of freedom                                25
  P-value (Chi-square)                           0.849
```

The measurement model fits! Now combine it with the latent variable model, as shown in Figure 1.12.

Figure 1.12: Brand Awareness Model Five




It is easy to edit model string `torus1` to put the $\lambda_j$ parameters in the measurement model. Showing just the first part of the model string,

```
> torus5 =
+ '
+ # Latent variable model
+     L_PI ~ gamma1*L_BrAw + gamma2*L_AdAw + gamma3*L_Inter
+     L_PBeh    ~ gamma4*L_Inter + beta*L_PI
+ # Measurement model
+     L_BrAw  =~ 1*w1 + lambda2*w2
+     L_AdAw  =~ 1*w3 + lambda4*w4
+     L_Inter =~ 1*w5 + lambda6*w6
+     L_PI    =~ 1*v1 + lambda8*v2
+     L_PBeh  =~ 1*v3 + lambda10*v4
```

Fitting the model,

```
> fit5 = lavaan(torus5, data=coffee)
Warning messages:
1: In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING:
    Could not compute standard errors! The information matrix could
    not be inverted. This may be a symptom that the model is not
    identified.
2: In lav_object_post_check(object) :
  lavaan WARNING: covariance matrix of latent variables
                is not positive definite;
                use lavInspect(fit, "cov.lv") to investigate.
```

The parameters of this model are definitely identifiable, so that's not the problem. The search has left the parameter space, and since the measurement model fits, the source of the trouble must be in the fit of the latent variable model. The output of `summary` contains some clues. Let us examine it one piece at a time.

```
> summary(fit5)
lavaan 0.6-7 ended normally after 2096 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         28
  Number of inequality constraints                  15

  Number of observations                           200

Model Test User Model:

  Test statistic                                31.127
  Degrees of freedom                                27
  P-value (Chi-square)                           0.266

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured
```

It used a lot of iterations (2,096), which can be an indication that the numerical search wandered off into nowhere. For comparison, `fit4` (the good measurement model with $\lambda_2, \lambda_4, \ldots, \lambda_{10}$) found a good solution in 161 iterations, and `fit3` (the full double measurement model) found a solution outside the parameter space in 193 iterations, when the method-of-moments estimator was also outside the parameter space. The fit we are considering (`fit5`) actually passes the goodness of fit test, with $G^2 = 31.127, p = 0.266$. It's still unacceptable, though, because the solution is outside the parameter space.

Continuing to look at the output of `summary`,

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  L_BrAw =~
    w1                1.000
    w2      (lmb2)    0.535       NA
  L_AdAw =~
    w3                1.000
    w4      (lmb4)    0.552       NA
  L_Inter =~
    w5                1.000
    w6      (lmb6)    1.094       NA
  L_PI =~
    v1                1.000
    v2      (lmb8)    0.708       NA
  L_PBeh =~
    v3                1.000
    v4      (lm10)    1.034       NA
```

Comparing the estimates from the good measurement model,

```
> coef(fit4)
        lambda2          lambda4          lambda6          lambda8         lambda10
          0.530            0.543            1.090            0.708            1.029
         w1~~w1           w2~~w2           w3~~w3           w4~~w4           w5~~w5
          5.106           12.955            7.034           13.401            6.205
         w6~~w6           v1~~v1           v2~~v2           v3~~v3           v4~~v4
          6.134            8.322           10.301            4.440            3.993
  L_BrAw~~L_BrAw   L_AdAw~~L_AdAw L_Inter~~L_Inter       L_PI~~L_PI   L_PBeh~~L_PBeh
         19.135           15.914           14.980           21.128           17.155
  L_BrAw~~L_AdAw  L_BrAw~~L_Inter     L_BrAw~~L_PI   L_BrAw~~L_PBeh  L_AdAw~~L_Inter
         12.297           13.502           16.248            7.883           11.306
    L_AdAw~~L_PI   L_AdAw~~L_PBeh    L_Inter~~L_PI  L_Inter~~L_PBeh     L_PI~~L_PBeh
         15.070            6.144           15.564            6.533            9.619
```

Looking at just the first line, we see that the $\widehat{\lambda}_j$ from `fit5` are almost identical to the ones from `fit4`, which means that they are above suspicion. Continuing to look at the output of `summary(fit5)`,

```
Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  L_PI ~
    L_BrAw  (gmm1)   47.719       NA
    L_AdAw  (gmm2) -156.406       NA
    L_Inter (gmm3)   80.361       NA
  L_PBeh ~
    L_Inter (gmm4)   -0.156       NA
    L_PI    (beta)    0.570       NA
```

Now we see a problem. The estimates of $\gamma_1$, $\gamma_2$ and $\gamma_3$ are very large in absolute value. Consider that the observable versions of all the variables involved are on a scale from zero to one hundred, and that one of the coefficients linking the latent version to the observable version is set to one. This means that the latent variables are also approximately on a scale from zero to one hundred. $\widehat{\gamma}_1 = 47.719$ means that a one-point change in brand awareness is thought to produce a 47-point change in purchase intention. This is entirely unbelievable. Furthermore, the extremely large negative value of $\widehat{\gamma}_2$ means that a very small increase in advertising awareness produces produces a *decrease* in purchase intention that is off the scale. This is even worse. The first three estimates are all extremely suspect. In contrast, the next two, $\widehat{\gamma}_4$ and $\widehat{\beta}$, seem unremarkable.

Looking at the estimated variances and covariances,

```
Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  L_BrAw ~~
    L_AdAw  (ph12)   12.498       NA
    L_Inter (ph13)   13.407       NA
  L_AdAw ~~
    L_Inter (ph23)   11.621       NA

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    L_BrAw  (ph11)   18.730       NA
    L_AdAw  (ph22)    9.691       NA
    L_Inter (ph33)   14.851       NA
   .L_PI    (psi1)  260.320       NA
   .L_PBeh  (psi2)   12.623       NA
   .w1      (omg1)    5.511       NA
   .w2      (omg2)   12.959       NA
   .w3      (omg3)   13.263       NA
   .w4      (omg4)   15.139       NA
   .w5      (omg5)    6.335       NA
   .w6      (omg6)    6.158       NA
   .v1      (omg7)    8.341       NA
   .v2      (omg8)   10.301       NA
   .v3      (omg9)    4.524       NA
   .v4      (om10)    3.903       NA
```

The only thing that jumps out is the large value of $\widehat{\psi}_1$, the variance of the error term feeding into latent purchase intention. Looking back at Figure 1.12, it is clear that all the obvious signs of pathology are in the latent regression linking latent purchase intention to latent brand awareness, advertising awareness, and interest in the product.

Following the suggestion in the warning message, we take a look at the estimated variance-covariance matrix of the latent variables, which is not positive definite.

```
> lavInspect(fit5, "cov.lv")
        L_BrAw L_AdAw L_Intr L_PI   L_PBeh
```

```
L_BrAw  18.730
L_AdAw  12.498  9.691
L_Inter 13.407 11.621 14.851
L_PI    16.411 14.534 15.565 21.059
L_PBeh   7.261  6.469  6.554  9.572 17.054
```

At first, nothing seems obviously wrong; for example, all the estimated variances are positive. It's true that one of the eigenvalues is negative (I checked), but this is something we can trust `lavaan` to get right.

Comparison with `lavInspect(fit4, "cov.lv")` is really helpful. Recall that `fit4` was the successful fit of the measurement model, so this is the real MLE of the covariance matrix of the latent variables. It's shown in Table 1.1. The biggest difference between

Table 1.1: MLE of the covariance matrix of latent variables for Brand Awareness data

```
lavInspect(fit4, "cov.lv")
        L_BrAw L_AdAw L_Intr L_PI   L_PBeh
L_BrAw  19.135
L_AdAw  12.297 15.914
L_Inter 13.502 11.306 14.980
L_PI    16.248 15.070 15.564 21.128
L_PBeh   7.883  6.144  6.533  9.619 17.155
```

these two matrices is in the estimated variance for `L_AdAw`, latent advertising awareness. The value in `fit5` is 9.691, while the value in `fit4` is 15.914. The `fit4` value is the real MLE of the variance of this latent exogenous variable, and has a lot more credibility.

In fact, the low variance in question causes the estimated variance-covariance matrix of just the exogenous latent variables to not be positive definite[22]. Again, we see a problem with estimation in the same part of the latent variable model. It's in the first stage, the latent regression linking latent purchase intention to latent brand awareness, advertising awareness, and interest in the product.

In general, when a numerical search leaves the parameter space, it could be either because of the starting values, or because the model is wrong. Here, it seems very likely to be the starting values. The reason is that this is just a regression, and its parameters are one-to-one with a set of variances and covariances that have already been estimated successfully. This point will become clear as we work to obtain better starting values, based on the estimated variances and covariances in `fit4`. Again, `fit4` comes from the successful measurement model represented in Figure 1.11, the one with $\lambda_2, \lambda_4, \ldots, \lambda_{10}$.

It would be possible to accomplish our goal by translating the regression notation of (16), but it is more informative to derive the starting values using the current notation. Let $\mathbf{x}_i$ denote the vector of latent exogenous variables $(X_{i,1}, X_{i,2}, X_{i,3})^\top$. There was trouble estimating $\mathbf{\Phi}_x = cov(\mathbf{x}_i)$, but we already have a good estimate: the first three rows and columns of Table 1.1. So we'll use that.

[22] I played around with it.

Write the sub-model we're considering as $y_{i,1} = \boldsymbol{\gamma}^\top \mathbf{x}_i + \epsilon_{i,1}$, where $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)^\top$. We need estimates of $\boldsymbol{\gamma}$ and $\psi_1 = var(\epsilon_{i,1})$ to use as starting values. Basic variance and covariance calculations yield

$$\begin{aligned} cov(\mathbf{x}_i, y_{i,1}) &= \boldsymbol{\Phi}_x \boldsymbol{\gamma} \\ var(y_{i,1}) &= \boldsymbol{\gamma}^\top \boldsymbol{\Phi}_x \boldsymbol{\gamma} + \psi_1 \end{aligned}$$

Use $\boldsymbol{\Phi}_{x,y_1}$ to denote $cov(\mathbf{x}_i, y_{i,1})$, the vector of three covariances between the exogenous variables and purchase intention. Estimates are directly available from Table 1.1. Starting values for the estimate of $\boldsymbol{\gamma}$ will be the very respectable estimate $\widehat{\boldsymbol{\gamma}} = \widehat{\boldsymbol{\Phi}}_x^{-1} \widehat{\boldsymbol{\Phi}}_{x,y_1}$. Using the estimated variance of purchase intention from Table 1.1, we get $\widehat{\psi}_1 = \widehat{\phi}_{4,4} - \widehat{\boldsymbol{\gamma}}^\top \widehat{\boldsymbol{\Phi}}_x \widehat{\boldsymbol{\gamma}} = \widehat{\phi}_{4,4} - \widehat{\boldsymbol{\Phi}}_{x,y_1}^\top \widehat{\boldsymbol{\Phi}}_x^{-1} \widehat{\boldsymbol{\Phi}}_{x,y_1}$. These estimates are one-to-one functions of the MLE from a closely related model for these data, so they should be very good starting values for the parameters of the model in Figure 1.12. Calculating,

```
> # The names of all these quantities should include "hat."
> Phi = lavInspect(fit4, "cov.lv")
> Phix = Phi[1:3,1:3]; Phix
          L_BrAw   L_AdAw  L_Inter
L_BrAw  19.13510 12.29660 13.50213
L_AdAw  12.29660 15.91372 11.30579
L_Inter 13.50213 11.30579 14.98033
> Phixy = as.matrix(Phi[1:3,4]); Phixy
            [,1]
L_BrAw  16.24761
L_AdAw  15.07005
L_Inter 15.56443
> gamma = t(Phixy) %*% solve(Phix); gamma
        L_BrAw    L_AdAw   L_Inter
[1,] 0.1996458 0.3932861 0.5622287
> psi1 = Phi[4,4] - as.numeric(gamma %*% Phix %*% t(gamma)); psi1
[1] 3.206661
```

These numbers are much more reasonable than the ones from `fit5`. Let's see if we can get away with specifying just 10 starting values. We'll drop the inequality constraints too, since `lavaan` will issue a warning if any variance estimate is negative.

```
> torus6 =
+ '
+ # Latent variable model
+     L_PI ~ gamma1*L_BrAw + start(0.1996458)*L_BrAw +
+     gamma2*L_AdAw        + start(0.3932861)*L_AdAw +
+     gamma3*L_Inter       + start(0.5622287)*L_Inter
+     L_PBeh    ~ gamma4*L_Inter + beta*L_PI
+ # Measurement model
+     L_BrAw  =~ 1*w1 + lambda2*w2
+     L_AdAw  =~ 1*w3 + lambda4*w4
```

```
+     L_Inter =~ 1*w5 + lambda6*w6
+     L_PI    =~ 1*v1 + lambda8*v2
+     L_PBeh  =~ 1*v3 + lambda10*v4
+ # Variances  and covariances
+     # Exogenous latent variables
+         L_BrAw ~~ phi11*L_BrAw   + start(19.13510)*L_BrAw  # Var(L_BrAw)         = phi11
+         L_BrAw ~~ phi12*L_AdAw   + start(12.29660)*L_AdAw  # Cov(L_BrAw,L_AdAw)  = phi12
+         L_BrAw ~~ phi13*L_Inter  + start(13.50213)*L_Inter # Cov(L_BrAw,L_Inter) = phi13
+         L_AdAw ~~ phi22*L_AdAw   + start(15.91372)*L_AdAw  # Var(L_AdAw)         = phi22
+         L_AdAw ~~ phi23*L_Inter  + start(11.30579)*L_Inter # Cov(L_AdAw,L_Inter) = phi23
+         L_Inter ~~ phi33*L_Inter + start(14.98033)*L_Inter # Var(L_Inter)        = phi33
+     # Errors in the latent model (epsilons)
+         L_PI ~~ psi1*L_PI + start(3.206661)*L_PI   # Var(epsilon1) = psi1
+         L_PBeh ~~ psi2*L_PBeh                      # Var(epsilon2) = psi2
+     # Measurement errors
+         w1 ~~ omega1*w1          # Var(e1)  = omega1
+         w2 ~~ omega2*w2          # Var(e2)  = omega2
+         w3 ~~ omega3*w3          # Var(e3)  = omega3
+         w4 ~~ omega4*w4          # Var(e4)  = omega4
+         w5 ~~ omega5*w5          # Var(e5)  = omega5
+         w6 ~~ omega6*w6          # Var(e6)  = omega6
+         v1 ~~ omega7*v1          # Var(e7)  = omega7
+         v2 ~~ omega8*v2          # Var(e8)  = omega8
+         v3 ~~ omega9*v3          # Var(e9)  = omega9
+         v4 ~~ omega10*v4         # Var(e10) = omega10
+ ' # End of model torus6
> fit6 = lavaan(torus6, data=coffee)
>
```

`lavaan` returns the R prompt with minimal time lag and no warning messages, which is a good sign.

```
> fit6
lavaan 0.6-7 ended normally after 108 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         28
  Number of inequality constraints                  15

  Number of observations                           200

Model Test User Model:

  Test statistic                                18.962
  Degrees of freedom                                27
  P-value (Chi-square)                           0.871
```

Finally, the model fits! `summary` gives numerical estimates of all the parameters, along with standard errors (square roots of the diagonal elements of the inverse of the observed Fisher information matrix), and large-sample $z$-tests of the null hypothesis that the parameter equals zero.

```
> summary(fit6)
lavaan 0.6-7 ended normally after 108 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         28

  Number of observations                           200

Model Test User Model:

  Test statistic                                18.962
  Degrees of freedom                                27
  P-value (Chi-square)                           0.871

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  L_BrAw =~
    w1                1.000
    w2      (lmb2)    0.528    0.077    6.861    0.000
  L_AdAw =~
    w3                1.000
    w4      (lmb4)    0.543    0.090    6.013    0.000
  L_Inter =~
    w5                1.000
    w6      (lmb6)    1.092    0.081   13.528    0.000
  L_PI =~
    v1                1.000
    v2      (lmb8)    0.707    0.066   10.745    0.000
  L_PBeh =~
    v3                1.000
    v4      (lm10)    1.040    0.110    9.457    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  L_PI ~
```

```
    L_BrAw  (gmm1)    0.229    0.145    1.581    0.114
    L_AdAw  (gmm2)    0.369    0.161    2.285    0.022
    L_Inter (gmm3)    0.553    0.170    3.253    0.001
  L_PBeh ~
    L_Inter (gmm4)   -0.129    0.257   -0.502    0.615
    L_PI    (beta)    0.546    0.224    2.438    0.015

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  L_BrAw ~~
    L_AdAw  (ph12)   12.301    1.864    6.598    0.000
    L_Inter (ph13)   13.480    1.831    7.360    0.000
  L_AdAw ~~
    L_Inter (ph23)   11.312    1.694    6.679    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    L_BrAw  (ph11)   19.200    3.110    6.174    0.000
    L_AdAw  (ph22)   15.910    3.033    5.246    0.000
    L_Inter (ph33)   14.961    2.153    6.949    0.000
   .L_PI    (psi1)    3.301    1.340    2.463    0.014
   .L_PBeh  (psi2)   12.620    2.097    6.019    0.000
   .w1      (omg1)    5.041    2.075    2.430    0.015
   .w2      (omg2)   12.974    1.413    9.179    0.000
   .w3      (omg3)    7.038    2.218    3.172    0.002
   .w4      (omg4)   13.400    1.477    9.074    0.000
   .w5      (omg5)    6.224    0.960    6.484    0.000
   .w6      (omg6)    6.098    1.063    5.735    0.000
   .v1      (omg7)    8.280    1.479    5.598    0.000
   .v2      (omg8)   10.299    1.215    8.477    0.000
   .v3      (omg9)    4.612    1.682    2.742    0.006
   .v4      (om10)    3.809    1.789    2.129    0.033
```

The estimates of $\lambda_2, \ldots, \lambda_{10}$ are essentially the same as the estimates from `fit4`, which is good. Comparing other estimates to the starting values we supplied,

```
> parTable(fit6)
   id     lhs op     rhs user block group free ustart exo    label plabel  start    est    se
1   1    L_PI  ~  L_BrAw    1     1     1    1  0.200   0   gamma1   .p1.  0.200  0.229 0.145
2   2    L_PI  ~  L_AdAw    1     1     1    2  0.393   0   gamma2   .p2.  0.393  0.369 0.161
3   3    L_PI  ~ L_Inter    1     1     1    3  0.562   0   gamma3   .p3.  0.562  0.553 0.170
4   4  L_PBeh  ~ L_Inter    1     1     1    4     NA   0   gamma4   .p4.  0.000 -0.129 0.257
5   5  L_PBeh  ~    L_PI    1     1     1    5     NA   0     beta   .p5.  0.000  0.546 0.224
6   6  L_BrAw =~      w1    1     1     1    0  1.000   0             .p6.  1.000  1.000 0.000
7   7  L_BrAw =~      w2    1     1     1    6     NA   0  lambda2   .p7.  0.476  0.528 0.077
8   8  L_AdAw =~      w3    1     1     1    0  1.000   0             .p8.  1.000  1.000 0.000
9   9  L_AdAw =~      w4    1     1     1    7     NA   0  lambda4   .p9.  0.421  0.543 0.090
10 10 L_Inter =~      w5    1     1     1    0  1.000   0           .p10.  1.000  1.000 0.000
11 11 L_Inter =~      w6    1     1     1    8     NA   0  lambda6  .p11.  0.724  1.092 0.081
```

```
12 12    L_PI =~      v1    1     1     1    0  1.000   0          .p12.  1.000  1.000 0.000
13 13    L_PI =~      v2    1     1     1    9     NA   0  lambda8 .p13.  0.594  0.707 0.066
14 14  L_PBeh =~      v3    1     1     1    0  1.000   0          .p14.  1.000  1.000 0.000
15 15  L_PBeh =~      v4    1     1     1   10     NA   0 lambda10 .p15.  0.807  1.040 0.110
16 16  L_BrAw ~~  L_BrAw    1     1     1   11 19.135   0    phi11 .p16. 19.135 19.200 3.110
17 17  L_BrAw ~~  L_AdAw    1     1     1   12 12.297   0    phi12 .p17. 12.297 12.301 1.864
18 18  L_BrAw ~~ L_Inter    1     1     1   13 13.502   0    phi13 .p18. 13.502 13.480 1.831
19 19  L_AdAw ~~  L_AdAw    1     1     1   14 15.914   0    phi22 .p19. 15.914 15.910 3.033
20 20  L_AdAw ~~ L_Inter    1     1     1   15 11.306   0    phi23 .p20. 11.306 11.312 1.694
21 21 L_Inter ~~ L_Inter    1     1     1   16 14.980   0    phi33 .p21. 14.980 14.961 2.153
22 22    L_PI ~~    L_PI    1     1     1   17  3.207   0     psi1 .p22.  3.207  3.301 1.340
23 23  L_PBeh ~~  L_PBeh    1     1     1   18     NA   0     psi2 .p23.  0.050 12.620 2.097
24 24      w1 ~~      w1    1     1     1   19     NA   0   omega1 .p24. 12.120  5.041 2.075
25 25      w2 ~~      w2    1     1     1   20     NA   0   omega2 .p25.  9.162 12.974 1.413
26 26      w3 ~~      w3    1     1     1   21     NA   0   omega3 .p26. 11.474  7.038 2.218
27 27      w4 ~~      w4    1     1     1   22     NA   0   omega4 .p27.  9.046 13.400 1.477
28 28      w5 ~~      w5    1     1     1   23     NA   0   omega5 .p28. 10.593  6.224 0.960
29 29      w6 ~~      w6    1     1     1   24     NA   0   omega6 .p29. 11.965  6.098 1.063
30 30      v1 ~~      v1    1     1     1   25     NA   0   omega7 .p30. 14.725  8.280 1.479
31 31      v2 ~~      v2    1     1     1   26     NA   0   omega8 .p31. 10.439 10.299 1.215
32 32      v3 ~~      v3    1     1     1   27     NA   0   omega9 .p32. 10.797  4.612 1.682
33 33      v4 ~~      v4    1     1     1   28     NA   0  omega10 .p33. 11.085  3.809 1.789
```

The column `ustart` shows the user-supplied starting values, `start` shows all the starting values, and `est` contains the parameter estimates (MLEs). It is clear that where starting values were supplied, the search moved from them just a little bit, at most. They were very good.

The output of `summary`, shows that that the coefficients linking the Set Two measurements to the latent variables are all significantly different from zero; they'd better be! But are they all significantly different from one? Starting with a likelihood ratio test of the null hypothesis that all five coefficients equal one,

```
> # Likelihood ratio test of
> # H0: lambda2 = lambda4 = lambda6 = lambda8 = lambda10 = 1
> anova(fit1,fit6)
Chi-Squared Difference Test

     Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit6 27 10947 11039 18.962
fit1 32 10996 11071 77.752     58.789       5 0.00000000002162 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

For the corresponding Wald test, it is convenient to use the publicly available function `Wtest`.

```
# For Wald tests: Wtest = function(L,Tn,Vn,h=0) # H0: L theta = h
source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/Wtest.txt")
```

As the comment indicates, `Wtest` allows testing of the linear null hypothesis $H_0 : \mathbf{L}\boldsymbol{\theta} = \mathbf{h}$, based on maximum likelihood. The argument `Tn` is the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}_n$,

and `Vn` is its asymptotic covariance matrix. It is helpful to display $\widehat{\boldsymbol{\theta}}_n$, just to verify the order of the parameters.

```
> thetahat = coef(fit6); thetahat
  gamma1   gamma2   gamma3   gamma4     beta  lambda2  lambda4  lambda6  lambda8
   0.229    0.369    0.553   -0.129    0.546    0.528    0.543    1.092    0.707
lambda10     phi11    phi12    phi13    phi22    phi23    phi33     psi1     psi2
   1.040   19.200   12.301   13.480   15.910   11.312   14.961    3.301   12.620
  omega1   omega2   omega3   omega4   omega5   omega6   omega7   omega8   omega9
   5.041   12.974    7.038   13.400    6.224    6.098    8.280   10.299    4.612
 omega10
   3.809
> LL = rbind(c(0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
+            c(0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
+            c(0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
+            c(0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
+            c(0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0))
> hh = c(1,1,1,1,1)
> Wtest(LL,thetahat,vcov(fit6),hh)
                          W                           df                      p-value
84.5066737182521876547980  5.0000000000000000000000  0.0000000000000001110223
```

Both the likelihood ratio test and the Wald test confirm overwhelmingly that the coefficients in question are not all one. To test the individual coefficients, it's convenient to use the MLEs and standard errors from `parTable`. The next-to-last column is the parameter estimate, and the last column is the standard error. The following code computes the $z$ statistics for $H_0 : \theta_j = 1$ for *all* the parameters but then displays only the relevant ones.

```
> pt6 = parTable(fit6); dim(pt6)
[1] 33 15
> z = as.numeric( (pt6[,14]-1)/pt6[,15] )
> # Extract only meaningful z statistics (lambda_j)
> z = z[c(7,9,11,13,15)]
> names(z) = c('lambda2', 'lambda4', 'lambda6', 'lambda8', 'lambda10')
> z
   lambda2    lambda4    lambda6    lambda8   lambda10
-6.1368432 -5.0581710  1.1367154 -4.4540676  0.3614714
> pt6[c(7,9,11,13,15),14] # Corresponding theta-hats
[1] 0.5278696 0.5431214 1.0917385 0.7069418 1.0397428
```

And we see that the 1.09 and the 1.04 are not significantly different from one.

## 1.7 Criticisms of structural equation modeling

Not everybody likes structural equation modeling. One objection is subjectivity. It's true that quite a lot of theoretical input is required to use this tool on a data set. One cannot compose a path diagram (or equivalently, a system of model equations) without making

some very definite assertions about the way the process works. Statisticians might object that they are not subject matter experts, perhaps with the sub-text that they don't want to think too hard about it, and especially they don't want to read books and articles in a foreign discipline. The solution to this problem is either find a collaborator, or go do something more theoretical.

Scientists, too, may feel uncomfortable. It's not the math; they are already resigned to the fact that they need to use statistical methods they do not understand all the way down to the bedrock. The problem is that they see themselves as empiricists. They have gone to a lot of trouble to collect the data, and now they want to hear what the data have to say. They do not want to impose their conjectures on the data[23]; it strikes them as unscientific.

One such scientist once said to a friend of mine (Lennon Li) something like "All these variables are connected to each other. Why not just run arrows from everything to everything else, and then test whether the coefficients are zero?" Lennon was faced with the task of explaining parameter identifiability to a busy, impatient, sleep-deprived physician who was already running late. In the end, Lennon wound up doing almost all the modeling himself. He did the best he could, but it was not an optimal outcome.

Actually, I have a lot of sympathy for the empirically-oriented user who is reluctant to engage in modeling. Frequently, the objection is not to modeling or theorizing per se, but to mixing this enterprise with the statistical analysis. It's a reasonable position, but I do have a few questions. First of all, is the data set strictly observational, or have some variables been manipulated by random assignment to treatment conditions? In the latter case, causal inference is the objective, and surely arrows should be going from the manipulated variables to others that could be deemed outcomes. Structural equation methods may have some advantages over a traditional statistical analysis. See Chapter **??**. If it's a purely observational study, here is another question for the skeptical user. Have you ever used ordinary linear regression on data like these? If so, you've had to decide which were the explanatory variables, and which were the response variables. How did you decide? It seems that you may have already been doing structural equation modeling of a basic sort. Do you agree that in regression, most explanatory variables are measured with error? If so, see Chapter 0. It's a slippery slope.

Sometimes, the objection is not so much to constructing models that will be incorporated into the statistical method, but to the interpretation of those models as causal. To be explicit about this, the objection is to drawing causal conclusions from observational data. We are back to the correlation-causation issue. One response is that while of course one cannot firmly establish cause and effect without random assignment, at least one can propose a causal model, and reject it if it does not fit the data. That being said, frequently

[23]If this sounds like an objection to Bayesian statistics, I agree. There is no doubt that even strictly frequentist structural equation modeling makes heavy use of prior information. Without some opinion based on past data or experience, how can you draw a path diagram? As I see it, both Bayesians and frequentists incorporate prior information into the statistical model, while Bayesians also have a prior distribution on the parameters. In fact, one could say that for the Bayesian, the model is part of the prior, though in simple applications that part of the prior distribution is degenerate. This statement applies to statistical models in general, not just to structural equation models.

(but not always), a model with causality flowing in one direction fits exactly as well as another model with causality flowing in the opposite direction. Some theoretical input is required. When one variable is collected at an earlier time period, it's easy. Other cases can be more challenging. As will be seen in Chapter 4, successful models of mutual influence are also possible under some circumstances.

Unfortunately, it is not so easy to dispose of the correlation-causation issue. Consider two variables that are both impacted by variables for which no observable measures are available. These unmeasured variables are aptly named "confounding" variables, because they really do confuse matters. Are $x$ and $y$ correlated because $x$ influences $y$, or is it because they are both influenced by the unmeasured variables? Or, are $d_1$ and $d_2$ correlated because $d_1$ and $d_2$ are both influenced by a latent variable $F$ (that's what the model says), or is it because they are both influenced by the unmeasured variables?

Recalling that error terms represent "all other influences," a path diagram that acknowledges the unmeasured influencers would have an extra curved, double-headed arrow — between an exogenous variable and an error term, as in Figure 6, or between two error terms as in Figure 1.7. In such cases, parameter identifiability is likely to be lost[24].

It's sometimes possible to model one's way out of the problem, and come up with another model that is both believable, and whose parameters are identifiable. If this is not possible, the analyst is in an uncomfortable position. The choice may be between proceeding ot fit a model that no thoughtful person could believe (hoping that it's not "too wrong"), and simply giving up. Even if one chooses to hold one's nose and proceed, it does not always work. As shown in Example 1.5.1, correlated error terms can lead to an MLE that is firmly, reliably and significantly outside the parameter space. In such a situation, one should not trust any of the estimates or tests associated with the fitted model. To proceed is basically fraudulent. I was in this situation once, and I had to back out of a project with a valued collaborator. I'm still sorry about that, Ana.

This is just one aspect of a larger problem that makes it difficult for some researchers to embrace structural equation modeling. The problem is that sometimes, a superficially reasonable model with identifiable parameters, simply do not fit. Then on further reflection, the analyst comes up with a model that is more believable. Unfortunately, the parameters of this more believable model are not identifiable. The analyst may suspect the problem with identifiability, without being able to confirm it mathematically. In any case, he or she tries to fit the model, and it blows up. Maybe it's the starting values. As we saw in Section 0.10.2 lack of identifiability can produce numerical problems that are hard to distinguish from the ones caused by bad starting values. So the analyst tries different starting values, but it blows up every time. A few experiences like this with different data sets are enough to turn anyone off.

I can see two possible remedies. The first is to know, not just guess, whether param-

[24]A notable exception is the double measurement design of Section 0.10.3 in Chapter 0; also see the calculations leading to (1.23) on page 177. There, the measurement error terms for each set of measurements are allowed to be correlated, though they are not allowed to be correlated between sets. The virtue of this is that it's quite natural for the measurements in one set to be contaminated by common influences. Minimizing such contamination between sets is something that can be accomplished by good study design.

eters are identifiable. I hope this book helps. The second remedy is better data – that is, data from a study that was designed with a particular structural equation model in mind. Identifiability issues are taken care of at the planning stage. Potential confounding variables are included in the data set, with adequate measurements. Correlations between measurement errors are minimized by carrying out some of the measurements in varying ways. For example, ask farmers how may cows they have, but also count them from aerial photographs.

This is an ideal state of affairs. Mostly, structural equation models are applied to data that were collected with other considerations in mind. In such cases, we do the best we can.

## 1.8 The rest of the book

In structural equation modeling, it is imperative to check parameter identifiability before proceeding to model fitting. The most direct way to check is to solve the covariance structure equations for the unknown parameters, but that can be a big job. Fortunately, there is a set of rules that often allow one to verify identifiability simply by examining the path diagram, without explicitly solving any equations. The next task is to derive these rules.

We will follow the logic of proving identifiability in two steps, as in the Brand Awareness example of Section 1.6. In the general two-stage model of Section 1.2, the parameters of the measurement model ($\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$) are first recovered from $\boldsymbol{\Sigma}$, the variance-covariance matrix of a vector of an observable data vector. Then, the parameters of the latent variable model ($\boldsymbol{\Phi}, \boldsymbol{\Gamma}, \boldsymbol{\beta}$ and $\boldsymbol{\Psi}$) are recovered from $\boldsymbol{\Phi}$. Since $\boldsymbol{\Phi}$ has already been shown to be a function of $\boldsymbol{\Sigma}$, this shows that all the parameters are a function of $\boldsymbol{\Sigma}$, and hence are identifiable.

Chapters 2 and 3 treat the measurement model. This is also a major topic in its own right, and goes by the name *factor analysis*. Chapter 4 is entitled *path analysis*. It treats models in which a set of endogenous variables may be influenced by a set of exogenous variables, and the endogenous variables may in turn influence other endogenous variables. This is an accurate description of the latent variable model, and the principles developed in Chapter 4 apply directly to the latent variable model. In Chapter 4, however, as in traditional path analysis, the models are described as if all the variables were observable. This makes the exposition easier, and in spite of the dangers of ignoring measurement error (see Chapter 0), surface path models can occasionally be useful.

Though there is other discussion and a number of examples, the main task of chapters 3 and 4 is develop a set of simple rules for parameter identifiability. These rules are assembled and stated verbally at the beginning of Chapter 6. Illustrations are given. Chapter 6 goes on to document a set of additional methods for dealing with identifiability issues when the standard rules do not apply. The burden of computation is considerably eased by the use of computer algebra.

When I apply structural equation models, I tend to decide whether a model fits by simply applying the likelihood ratio test for goodness of fit. This is not a particularly

popular choice, and Chapter 7 presents a wider range of options. The reader will not be surprised to learn that in the end, I conclude that I am right.

At this point, the reader has the classical structural equation modeling toolkit, perhaps with a deeper understanding of identifiability than usual. The remainder of the book will cover topics including the following. This will be more complete once I have finished writing it.

- True experimental studies (MIMIC)
- Groebner basis
- Categorical data
- Multiple groups
-

# Chapter 2

# Exploratory Factor Analysis

In experimental design, the term "factor" refers to a categorical explanatory variable. In structural equation modeling and in the sub-field of factor analysis, a *factor* is a latent variable, period. Factor analysis may be said to originate with a 92-page article [60] by Charles Spearman in the 1901 *American Journal of Psychology*, entitled "General intelligence, objectively determined and measured." If you believe that some people are generally smarter than others, the basic idea is quite natural. True intelligence cannot be directly observed, so it's a latent variable. However, we can observe performance on various tests and puzzles. Spearman proposed that the correlations among observable variables arise from their connection to a common "g" factor — general intelligence.

The early history of factor analysis is described masterfully in Harman's (1960, 1967, 1976) classic *Modern factor analysis* [28]. Though Harman brings relative clarity to this murky literature, his book is almost guaranteed to be frustrating for a statistician to read. Lawley and Maxwell's (1971) *Factor analysis as a statistical method* is a welcome antidote. Bastlevsky's (1994) *Statistical factor analysis and related methods* [2] is a strong and more recent treatment of the topic.

Factor analysis may be divided into two types, commonly called *exploratory* factor analysis and *confirmatory* factor analysis. The books cited above are about exploratory factor analysis, which came first historically. While both types of factor analysis are special cases of structural equation models, it is confirmatory factor analysis that provides a useful measurement model. Exploratory factor analysis is helpful for understanding confirmatory factor analysis. Another good reason to learn about exploratory factor analysis is that some people still do it, or may ask you to do it.

## 2.1 Principal Components Analysis

Before describing what factor analysis is, it will be helpful to describe what it is not. Principal components analysis is not factor analysis. Factors are unobservable latent variables. Principal components are linear combinations of the sample data. The very existence of factors depends on one's acceptance of a fairly elaborate statistical model, while the statistical model underlying principal components is quite minimal, if there is

one at all. Still, principal components analysis and factor analysis have a similar flavour, and some of the ideas from principal components are used in factor analysis.

The main application of principal components analysis is data reduction. Suppose you have a large number of variables that are correlated with one another. Principal components analysis allows you to find a smaller set of linear combinations of the variables, linear combinations that contain most of the variation in the original set. It may be that little is lost by using the linear combinations in place of the original variables, and there can be substantial advantages in terms of storage and processing.

In the most relevant version of principal components, there are $k$ observable variables that are standardized[1], by subtracting off their means and dividing by their standard deviations. Collect the variables into a $k$-dimensional random vector $\mathbf{z} = [z_j]$, with $E(\mathbf{z}) = \mathbf{0}$ and $cov(\mathbf{z}) = \mathbf{\Sigma}$. Because of standardization, $\mathbf{\Sigma}$ is a correlation matrix.

Recall the spectral decomposition $\mathbf{\Sigma} = \mathbf{CDC}^\top$ (see Section A.2 in Appendix A), where $\mathbf{D}$ is a diagonal matrix containing the $k$ eigenvalues of $\mathbf{\Sigma}$ in descending order, and the columns of the $k \times k$ matrix $\mathbf{C} = [c_{ij}]$ contain the corresponding eigenvectors. The eigenvectors are orthonormal, so $\mathbf{CC}^\top = \mathbf{C}^\top\mathbf{C} = \mathbf{I}$.

Let $\mathbf{y} = \mathbf{C}^\top\mathbf{z} = [y_j]$. The transformed variables in $\mathbf{y}$ will be called the *principal components* of $\mathbf{z}$. Immediately, we have $E(\mathbf{y}) = \mathbf{0}$ and

$$\begin{aligned} cov(\mathbf{y}) &= cov(\mathbf{C}^\top\mathbf{z}) \\ &= \mathbf{C}^\top cov(\mathbf{z})\mathbf{C} \\ &= \mathbf{C}^\top\mathbf{\Sigma}\mathbf{C} \\ &= \mathbf{C}^\top\,\mathbf{CDC}^\top\,\mathbf{C} \\ &= \mathbf{D}, \end{aligned} \tag{2.1}$$

so that the elements of $\mathbf{y}$ are uncorrelated, and their variances are the eigenvalues of $\mathbf{\Sigma}$, sorted from largest to smallest.

Since $\mathbf{y} = \mathbf{C}^\top\mathbf{z}$, we can also write the original variables in terms of the principal components as $\mathbf{z} = \mathbf{Cy}$. In scalar form,

$$\begin{aligned} z_1 &= c_{11}y_1 + c_{12}y_2 + \cdots + c_{1k}y_k \\ z_2 &= c_{21}y_1 + c_{22}y_2 + \cdots + c_{2k}y_k \\ \vdots & \qquad\qquad \vdots \\ z_k &= c_{k1}y_1 + c_{k2}y_2 + \cdots + c_{kk}y_k. \end{aligned}$$

Because the elements of $\mathbf{y}$ are uncorrelated, the variance of variable $j$ is

$$\begin{aligned} Var(z_j) &= Var(c_{j1}y_1 + c_{j2}y_2 + \cdots + c_{jk}y_k) \\ &= c_{j1}^2 Var(y_1) + c_{j2}^2 Var(y_2) + \cdots + c_{jk}^2 Var(y_k) \\ &= c_{j1}^2\lambda_1 + c_{j2}^2\lambda_2 + \cdots + c_{jk}^2\lambda_k = 1. \end{aligned} \tag{2.2}$$

[1] In the other main version of principal components, the variables are not standardized. The development is very similar.

Thus, the variance of $z_j$ is decomposed into the part explained by $y_1$, the part explained by $y_2$, and so on. Specifically, $y_1$ explains $c_{j1}^2\lambda_1$ of the variance, $y_2$ explains $c_{j2}^2\lambda_2$ of the variance, etc.. Because $z_j$ is standardized, these are *proportions* of variance.

They are also squared correlations. Correlation is covariance divided by the product of standard deviations. Using the fact that $cov(y_i, y_j) = 0$ for $i \neq j$,

$$\begin{aligned} Cov(z_i, y_j) &= Cov(c_{i1}y_1 + c_{i2}y_2 + \cdots + c_{ij}y_j + \cdots + c_{jk}y_k, y_j) \\ &= c_{ij}Cov(y_j, y_j) \\ &= c_{ij}\lambda_j. \end{aligned}$$

Then,

$$\begin{aligned} Corr(z_i, y_j) &= \frac{Cov(z_i, y_j)}{SD(z_i)SD(y_j)} \\ &= \frac{c_{ij}\lambda_j}{1\ \sqrt{\lambda_j}} = c_{ij}\sqrt{\lambda_j}, \end{aligned} \tag{2.3}$$

and the *squared* correlation between $z_i$ and $y_j$ is $c_{ij}^2\lambda_j$.

Looking at the variances of all the original variables,

$$\begin{aligned} Var(z_1) &= c_{11}^2\lambda_1 + c_{12}^2\lambda_2 + \cdots + c_{1k}^2\lambda_k \\ Var(z_2) &= c_{21}^2\lambda_1 + c_{22}^2\lambda_2 + \cdots + c_{2k}^2\lambda_k \\ \vdots & \qquad\qquad \vdots \\ Var(z_k) &= c_{k1}^2\lambda_1 + c_{k2}^2\lambda_2 + \cdots + c_{kk}^2\lambda_k. \end{aligned} \tag{2.4}$$

The pieces of variance being added up are the squared correlations between the original variables and the principal components.

Imagine a $k \times k$ matrix of these squared correlations, with the original variables corresponding to rows, and the principal components corresponding to columns. The layout is the same as the equations (2.4). If you add the entries in any row, you get one. If you add the entries in a column, you get the total amount of variance in the original variables that is explained by that principal component. The sum of entries in column $j$ is

$$\begin{aligned} \sum_{i=1}^{k} c_{ij}^2\lambda_j &= \lambda_j \sum_{i=1}^{k} c_{ij}^2 \\ &= \lambda_j \cdot 1 = \lambda_j, \end{aligned} \tag{2.5}$$

where the squared weights add to one because the eigenvectors are of unit length. This means that the eigenvalues are both the variances of the principal components and the amounts of variance in the original variables that are explained by the respective principal components. The total variance in the original variables is the trace of $\mathbf{\Sigma}$, which equals $k$. The trace of a symmetric matrix is the sum of its eigenvalues, and everything adds up.

It's actually even better than that. There is a well-known theorem saying that $y_1$ has the greatest possible variance of any linear combination whose squared weights add up to

one. In addition, $y_2$ is the linear combination that has the greatest variance subject to the constraints that it's orthogonal to $y_1$ and its squared weights add to one. Continuing, $y_3$ is the linear combination that has the greatest variance subject to the constraints that it's orthogonal to $y_1$ and $y_2$, and its squared weights add to one — and so on. This means that the principal components are optimal in the sense that the first one explains the greatest possible amount of variance, and all the succeeding components explain the greatest possible amounts of the variance that remains unexplained by the earlier ones.

If the correlations among the original variables are substantial, the first few eigenvalues will be relatively large. The data reduction idea is to retain only the first several principal components, the ones that contain most of the variation in the original variables. The expectation is that they will capture most of the *meaningful* variation.

To apply this method to actual data, suppose you have $n$ observations on $k$ variables. First standardize all the variables, by subtracting off sample means and dividing by sample standard deviations. Assemble the standardized data into an $n \times k$ matrix $\mathbf{Z} = [z_{ij}]$. The true correlation matrix $\boldsymbol{\Sigma}$ is unknown, so use the sample correlation matrix $\widehat{\boldsymbol{\Sigma}}$. Based upon the spectral decomposition $\widehat{\boldsymbol{\Sigma}} = \widehat{\mathbf{C}}\widehat{\mathbf{D}}\widehat{\mathbf{C}}^\top$, calculate $\widehat{\mathbf{Y}} = \mathbf{Z}\widehat{\mathbf{C}}$. The rows of $\mathbf{Z}$ contain standardized data vectors, and the rows of $\widehat{\mathbf{Y}}$ contain the corresponding vectors of principal component values. $\widehat{\mathbf{Y}}$ has a hat because it is a matrix of the *sample* principal components. It can be informative to look at a matrix of squared sample correlations between the original variables and the components, because the entries are estimated proportions of variance in each variable that are explained by each component.

A nice feature of principal components is that the formulas given earlier in this section are exactly correct for sample principal components. This is because most of the rules for variances and covariances are also true for the sample versions[2]. As a result, it is possible to present principal components analysis as a purely descriptive procedure, without assuming any sampling model at all. Some textbooks do it this way; it's a matter of taste.

In any case, the main application of principal components is data reduction. The data reduction strategy is to retain just a few columns of $\widehat{\mathbf{Y}}$, because those principal components account for most of the variance in the original variables. But where do you draw the line? How many principal components should you preserve? A standard answer is to keep the components with eigenvalues greater than one, because one is the amount of variance in a single original variable. After that point, the principal components explain no more variance than the original variables.

**Example 2.1.1** *The Body-Mind Data*

[2]This statement is true and it's good enough, but here is an another way of thinking about it. The formulas developed for principal components are true for any distribution of the observed data. In particular, they are true for the rather peculiar discrete multivariate distribution that puts probability $\frac{1}{n}$ on each observed data vector. Think of the observed data vectors as strings of beads in an urn. We are sampling from this urn with replacement. It's the re-sampling model that is used in the bootstrap! For this distribution, the population mean, variance, covariance and so on may be calculated using usual formulas for the corresponding sample moments – provided that one uses the variance and covariance formulas with $n$ in the denominator rather than $n-1$. Consequently, all the formulas derived here apply directly to sample principal components.

The Body-Mind data are a set of educational test scores and physical measurements for a sample of high school students[3]. The variables are

- `sex`: F or M.
- `progmat`: Progressive matrices (puzzle) score.
- `reason`: Reasoning score.
- `verbal`: Verbal (reading and vocabulary) score.
- `headlng`: Head Length in mm.
- `headbrd`: Head Breadth in mm.
- `headcir`: Head Circumference in mm.
- `bizyg`: Bizygomatic breadth in mm, basically how far apart the eyes are.
- `weight`: In pounds.
- `height`: In cm.

These data will be used to illustrate true factor analysis as well as principal components. We begin by reading the data, and looking at basic descriptive statistics and the correlation matrix.

```
> rm(list=ls())
> bodymind = read.table('http://www.utstat.toronto.edu/~brunner/openSEM/data/bodymind.data.txt')
> head(bodymind)
  sex progmat reason verbal headlng headbrd headcir bizyg weight height
1   M     108    128    136     182     162     553   140    144   1769
2   F      81    110     94     192     156     571   143    144   1633
3   F     110    134    132     186     145     549   131    135   1672
4   F      95     88     83     189     139     536   124    109   1700
5   M      83     94    100     180     163     549   141    124   1679
6   M     105     77     92     195     148     560   134    126   1651
> dim(bodymind) # Number of rows,columns
[1] 80 10
dat = as.matrix(bodymind[,2:10]) # Omit sex, make dat a matrix rather than a data frame.
> # summary(dat)
> Sigma_hat = cor(dat); round(Sigma_hat,3)
        progmat reason verbal headlng headbrd headcir bizyg weight height
progmat   1.000  0.514  0.539   0.323   0.099   0.315 0.200  0.132  0.197
reason    0.514  1.000  0.728   0.203   0.053   0.322 0.291  0.171  0.207
verbal    0.539  0.728  1.000   0.260   0.139   0.354 0.337  0.236  0.199
headlng   0.323  0.203  0.260   1.000   0.255   0.821 0.475  0.506  0.554
```

[3]This is a modified subset of data reported in the journal *Human Biology* [17]. The data are used here without permission, but I believe they have been sufficiently hacked so that the original copyright no longer applies, and they can be protected under a Creative Commons license. Good luck trying to recover the original data values.

```
headbrd   0.099  0.053  0.139   0.255   1.000   0.604 0.692  0.368  0.362
headcir   0.315  0.322  0.354   0.821   0.604   1.000 0.713  0.641  0.591
bizyg     0.200  0.291  0.337   0.475   0.692   0.713 1.000  0.589  0.614
weight    0.132  0.171  0.236   0.506   0.368   0.641 0.589  1.000  0.599
height    0.197  0.207  0.199   0.554   0.362   0.591 0.614  0.599  1.000
```

The R functions `princomp` and `prcomp` will do principal components analysis, but we'll use spectral decomposition directly at first for illustrative purposes. The `eigen` function returns a list with two elements. The first element is a vector of eigenvalues, and the second element is the matrix $\mathbf{C}$ in $\mathbf{A} = \mathbf{CDC}^\top$. Column $j$ of the matrix $\mathbf{C}$ is the eigenvector corresponding to $\lambda_j$.

```
> eigenSigma = eigen(Sigma_hat); eigenSigma
eigen() decomposition
$values
[1] 4.28768216 1.77444482 0.87126975 0.64039055 0.47989427 0.40504511 0.26315906
[8] 0.21010253 0.06801175

$vectors
            [,1]        [,2]        [,3]       [,4]        [,5]        [,6]        [,7]
 [1,] -0.2274301 -0.47286949  0.10386693 -0.5037581  0.59758999 -0.29003259 -0.08152051
 [2,] -0.2405745 -0.54632083 -0.12052776  0.2965707 -0.17975676  0.26454653 -0.63108107
 [3,] -0.2665589 -0.51874379 -0.16430737  0.1996142 -0.24123089 -0.07990715  0.71935768
 [4,] -0.3622340  0.08683821  0.53544154 -0.3586357 -0.34767275  0.16737858  0.08869681
 [5,] -0.2933333  0.27697281 -0.66373737 -0.3094155  0.04112189 -0.03633303 -0.01019606
 [6,] -0.4377198  0.12657178  0.08577647 -0.2525368 -0.33524350  0.01081660 -0.14979213
 [7,] -0.4007471  0.17219323 -0.34669164  0.1054639  0.05971130  0.13277579  0.01297777
 [8,] -0.3513037  0.20963075  0.18723810  0.4548388  0.02650610 -0.74034211 -0.13982797
 [9,] -0.3556358  0.18698153  0.24048299  0.3351036  0.55960504  0.49428994  0.16579073
             [,8]         [,9]
 [1,]  0.10288021  0.040581025
 [2,] -0.11345554 -0.166563741
 [3,] -0.09839457  0.035693597
 [4,]  0.07347486 -0.532701450
 [5,] -0.44524651 -0.315589722
 [6,] -0.14639593  0.751580920
 [7,]  0.81059576 -0.002188321
 [8,] -0.07609981 -0.128655279
 [9,] -0.28094584  0.067358086
```

Since only the first two eigenvalues are greater than one, the conventional choice for data reduction would be to retain only the first two sample principal components. Dividing the eigenvalues by the number of variables yields the proportions of the total variance explained by each component.

```
> lambda_hat = eigenSigma$values
> lambda_hat/9         # Proportions of explained variance
[1] 0.476409129 0.197160535 0.096807750 0.071154506 0.053321586 0.045005012 0.029239896
[8] 0.023344726 0.007556861
> cumsum(lambda_hat/9) # Cumulative sum
[1] 0.4764091 0.6735697 0.7703774 0.8415319 0.8948535 0.9398585 0.9690984 0.9924431
[9] 1.0000000
```

It seems that the first two components account for around 67% of the total variance in the observed variables, and five components would account for about 90%.

Calculating $\mathbf{Z}$ and then $\widehat{\mathbf{Y}} = \mathbf{Z}\widehat{\mathbf{C}}$, we verify (2.1), which says $cov(\mathbf{y}) = \mathbf{D}$.

```
> > Z = scale(dat) # Standardize
> C_hat = eigenSigma$vectors #$
> Y_hat = Z %*% C_hat # Sample principal components
> # Looking at the variance-covariance matrix of the principal components,
> round(var(Y_hat), 4) # Should equal D
       [,1]   [,2]   [,3]   [,4]   [,5]  [,6]   [,7]   [,8]  [,9]
 [1,] 4.2877 0.0000 0.0000 0.0000 0.0000 0.000 0.0000 0.0000 0.000
 [2,] 0.0000 1.7744 0.0000 0.0000 0.0000 0.000 0.0000 0.0000 0.000
 [3,] 0.0000 0.0000 0.8713 0.0000 0.0000 0.000 0.0000 0.0000 0.000
 [4,] 0.0000 0.0000 0.0000 0.6404 0.0000 0.000 0.0000 0.0000 0.000
 [5,] 0.0000 0.0000 0.0000 0.0000 0.4799 0.000 0.0000 0.0000 0.000
 [6,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.405 0.0000 0.0000 0.000
 [7,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.000 0.2632 0.0000 0.000
 [8,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.000 0.0000 0.2101 0.000
 [9,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.000 0.0000 0.0000 0.068
```

There it is: a diagonal matrix with the eigenvalues on the diagonal.

Based on the eigenvalues, let's retain just the first two components and estimate how much variance they explain. First, look at the correlations.

```
> y = Y_hat[,1:2] # Just the first two components
> zy = cor(Z,y); zy
              [,1]       [,2]
progmat -0.4709330 -0.6299014
reason  -0.4981509 -0.7277446
verbal  -0.5519561 -0.6910097
headlng -0.7500678  0.1156757
headbrd -0.6073970  0.3689507
headcir -0.9063741  0.1686041
bizyg   -0.8298157  0.2293757
weight  -0.7274347  0.2792455
height  -0.7364050  0.2490749
```

All of the large correlations are negative, so they are a bit harder to look at. If this is a problem, the signs of a principal component can be flipped, reversing the signs of the correlation between that component and any variable. To see why this is true, recall the definition of an eigenvalue and associated eigenvector: $\mathbf{Ax} = \lambda\mathbf{x}$. Clearly if $\mathbf{x}$ is an eigenvector corresponding to $\lambda$, so is $-\mathbf{x}$. Since a principal component is a linear combination of variables whose weights are the elements of an eigenvector, the sign is arbitrary.

Now we will check Equation (2.3), which says $Corr(z_i, y_j) = c_{ij}\sqrt{\lambda_j}$. We should be able to reproduce the matrix of correlations between $\mathbf{Z}$ and the first two components by multiplying the first two columns of $\widehat{\mathbf{C}}$ by the matrix $\begin{pmatrix} \sqrt{\widehat{\lambda}_1} & 0 \\ 0 & \sqrt{\widehat{\lambda}_2} \end{pmatrix}$.

```
> A = rbind(c( sqrt(lambda_hat[1]), 0 ),
```

```
+           c(0, sqrt(lambda_hat[2]) )  )
> C_hat[,1:2] %*% A
            [,1]       [,2]
 [1,] -0.4709330 -0.6299014
 [2,] -0.4981509 -0.7277446
 [3,] -0.5519561 -0.6910097
 [4,] -0.7500678  0.1156757
 [5,] -0.6073970  0.3689507
 [6,] -0.9063741  0.1686041
 [7,] -0.8298157  0.2293757
 [8,] -0.7274347  0.2792455
 [9,] -0.7364050  0.2490749
```

Okay, it worked: Estimated $Corr(z_i, y_j)$ is $\widehat{c}_{ij}\sqrt{\widehat{\lambda}_j}$.

The squared correlations are components of variance. The `addmargins` function is used below to add row and column sums. It's easier to look at the output rounded to three decimal places.

```
> zy2 = zy^2
> round( addmargins(zy2, margin = c(1,2), FUN = sum) , 3)

Margins computed over dimensions
in the following order:
1:
2:
                      sum
progmat 0.222 0.397 0.619
reason  0.248 0.530 0.778
verbal  0.305 0.477 0.782
headlng 0.563 0.013 0.576
headbrd 0.369 0.136 0.505
headcir 0.822 0.028 0.850
bizyg   0.689 0.053 0.741
weight  0.529 0.078 0.607
height  0.542 0.062 0.604
sum     4.288 1.774 6.062
```

This shows, for example, that the first principal component explains 54.2% of the variance in height, and the second principal component explains an additional 6.2%. The first two principal components explain around 85% of the variance in head circumference, but only about 50.5% of the variance in head breadth. Also, the column totals are the eigenvalues, as in (2.5). These are all *estimated* values, of course.

**Principal components the easy way** It's a bit easier to use a specialized R function for principal components analysis, rather than relying on `eigen`. I prefer `prcomp` over `princomp`, because `princomp` has some unfortunate features that have been retained for compatibility with the defunct commercial software S-plus.

In the `prcomp` function, the `scale = T` option divides variables by their sample standard deviations. The option `center` is true by default, so the data are converted to $z$-scores. This is what we want.

```
> # Principal components the easy way
> # help(prcomp)
> pc = prcomp(dat, scale = T)
```

The object `pc` is a list. The `ls` function shows its elements.

```
> ls(pc)
[1] "center"   "rotation" "scale"    "sdev"     "x"
```

The element `pc$center` contains the sample means of the variables before standardization; `pc$scale` contains the standard deviations. `sdev` has the standard deviations of the components. Squaring the `sdev` vector yields the eigenvalues of the sample correlation matrix.

```
> pc$sdev^2 # Eigenvalues
[1] 4.28768216 1.77444482 0.87126975 0.64039055 0.47989427 0.40504511 0.26315906
[8] 0.21010253 0.06801175
> lambda_hat # For comparison
[1] 4.28768216 1.77444482 0.87126975 0.64039055 0.47989427 0.40504511 0.26315906
[8] 0.21010253 0.06801175
```

The list element `pc$rotation` corresponds to the $\widehat{\mathbf{C}}$ matrix produced by the spectral decomposition. Since $\widehat{\mathbf{C}}$ is an orthogonal matrix, it is indeed a rotation.

```
> pc$rotation
                PC1         PC2         PC3         PC4         PC5         PC6         PC7
progmat -0.2274301 -0.47286949  0.10386693 -0.5037581  0.59758999 -0.29003259  0.08152051
reason  -0.2405745 -0.54632083 -0.12052776  0.2965707 -0.17975676  0.26454653  0.63108107
verbal  -0.2665589 -0.51874379 -0.16430737  0.1996142 -0.24123089 -0.07990715 -0.71935768
headlng -0.3622340  0.08683821  0.53544154 -0.3586357 -0.34767275  0.16737858 -0.08869681
headbrd -0.2933333  0.27697281 -0.66373737 -0.3094155  0.04112189 -0.03633303  0.01019606
headcir -0.4377198  0.12657178  0.08577647 -0.2525368 -0.33524350  0.01081660  0.14979213
bizyg   -0.4007471  0.17219323 -0.34669164  0.1054639  0.05971130  0.13277579 -0.01297777
weight  -0.3513037  0.20963075  0.18723810  0.4548388  0.02650610 -0.74034211  0.13982797
height  -0.3556358  0.18698153  0.24048299  0.3351036  0.55960504  0.49428994 -0.16579073
                PC8          PC9
progmat -0.10288021  0.040581025
reason   0.11345554 -0.166563741
verbal   0.09839457  0.035693597
headlng -0.07347486 -0.532701450
headbrd  0.44524651 -0.315589722
headcir  0.14639593  0.751580920
bizyg   -0.81059576 -0.002188321
weight   0.07609981 -0.128655279
height   0.28094584  0.067358086
```

```
> C_hat # For comparison
            [,1]        [,2]        [,3]       [,4]        [,5]        [,6]        [,7]
 [1,] -0.2274301 -0.47286949  0.10386693 -0.5037581  0.59758999 -0.29003259 -0.08152051
 [2,] -0.2405745 -0.54632083 -0.12052776  0.2965707 -0.17975676  0.26454653 -0.63108107
 [3,] -0.2665589 -0.51874379 -0.16430737  0.1996142 -0.24123089 -0.07990715  0.71935768
 [4,] -0.3622340  0.08683821  0.53544154 -0.3586357 -0.34767275  0.16737858  0.08869681
 [5,] -0.2933333  0.27697281 -0.66373737 -0.3094155  0.04112189 -0.03633303 -0.01019606
```

```
 [6,] -0.4377198  0.12657178  0.08577647 -0.2525368 -0.33524350  0.01081660 -0.14979213
 [7,] -0.4007471  0.17219323 -0.34669164  0.1054639  0.05971130  0.13277579  0.01297777
 [8,] -0.3513037  0.20963075  0.18723810  0.4548388  0.02650610 -0.74034211 -0.13982797
 [9,] -0.3556358  0.18698153  0.24048299  0.3351036  0.55960504  0.49428994  0.16579073
             [,8]         [,9]
 [1,]  0.10288021  0.040581025
 [2,] -0.11345554 -0.166563741
 [3,] -0.09839457  0.035693597
 [4,]  0.07347486 -0.532701450
 [5,] -0.44524651 -0.315589722
 [6,] -0.14639593  0.751580920
 [7,]  0.81059576 -0.002188321
 [8,] -0.07609981 -0.128655279
 [9,] -0.28094584  0.067358086
```

Finally, `pc$x` has the principal components themselves.

```
> dim(pc$x) # x is a matrix of the principal components Y_hat = Z %*% C_hat
[1] 80  9
> head(pc$x) # Just the first 6 rows
         PC1        PC2         PC3         PC4        PC5         PC6         PC7
1 -2.9056790 -0.8163483 -2.05648959  0.89345100  1.0826163  0.09581676  0.09097201
2 -1.8420248  1.6868136 -1.11946332  0.61460425 -1.7388326  0.20651893  0.68366132
3 -1.1270571 -2.3088592  0.08809617  0.75714079  0.1575711 -0.19017485  0.51153249
4  1.6221315  0.2340440  1.62777485  0.07639917  0.3896938  0.65365783 -0.33948607
5 -0.6431189  1.8507668 -2.60883792  0.58933649 -0.1899104  0.47035165 -0.33536456
6 -0.5757390  0.9010777  0.79544134 -1.28687495  0.1150836 -0.39632242 -0.59876963
          PC8         PC9
1  0.66523233 -0.13093412
2 -0.60878367  0.09346307
3  0.34061367  0.13503816
4  0.44387949  0.16416604
5  0.06955952  0.02486900
6 -0.48127952  0.34279834
> head(Y_hat) # For comparison
        [,1]       [,2]        [,3]        [,4]       [,5]        [,6]        [,7]
1 -2.9056790 -0.8163483 -2.05648959  0.89345100  1.0826163  0.09581676 -0.09097201
2 -1.8420248  1.6868136 -1.11946332  0.61460425 -1.7388326  0.20651893 -0.68366132
3 -1.1270571 -2.3088592  0.08809617  0.75714079  0.1575711 -0.19017485 -0.51153249
4  1.6221315  0.2340440  1.62777485  0.07639917  0.3896938  0.65365783  0.33948607
5 -0.6431189  1.8507668 -2.60883792  0.58933649 -0.1899104  0.47035165  0.33536456
6 -0.5757390  0.9010777  0.79544134 -1.28687495  0.1150836 -0.39632242  0.59876963
         [,8]        [,9]
1 -0.66523233 -0.13093412
2  0.60878367  0.09346307
3 -0.34061367  0.13503816
4 -0.44387949  0.16416604
5 -0.06955952  0.02486900
6  0.48127952  0.34279834
```

A useful feature of `prcomp` is that it's easy to specify the number of components you want to extract. This is accomplished by specifying `rank` in the call to `prcomp`.

```
> pc2 = prcomp(dat, scale = T, rank = 2) # Retain two principal components
```

```
> pc2$rotation
               PC1         PC2
progmat -0.2274301 -0.47286949
reason  -0.2405745 -0.54632083
verbal  -0.2665589 -0.51874379
headlng -0.3622340  0.08683821
headbrd -0.2933333  0.27697281
headcir -0.4377198  0.12657178
bizyg   -0.4007471  0.17219323
weight  -0.3513037  0.20963075
height  -0.3556358  0.18698153
```

Only the first two columns of $\widehat{\mathbf{C}}$ are returned. Post-multiplying this matrix by the matrix of standardized data in $\mathbf{Z}$ yields an $80 \times 2$ matrix of just the first two principal components.

```
> head(pc2$x) # There should be 2 columns
         PC1        PC2
1 -2.9056790 -0.8163483
2 -1.8420248  1.6868136
3 -1.1270571 -2.3088592
4  1.6221315  0.2340440
5 -0.6431189  1.8507668
6 -0.5757390  0.9010777
```

This is all very nice, but it's not factor analysis. Principal components analysis and factor analysis are frequently confused, especially by social scientists. In a consulting situation, suppose your client claims to have done a factor analysis. You should ask "What kind of factor analysis?" If the client doesn't know, ask "What software did you use?" If it's SAS or SPSS, ask "Did you use the default options?" If the answer is yes, it was a principal components analysis. We now turn to true factor analysis.

## 2.2 True Factor Analysis

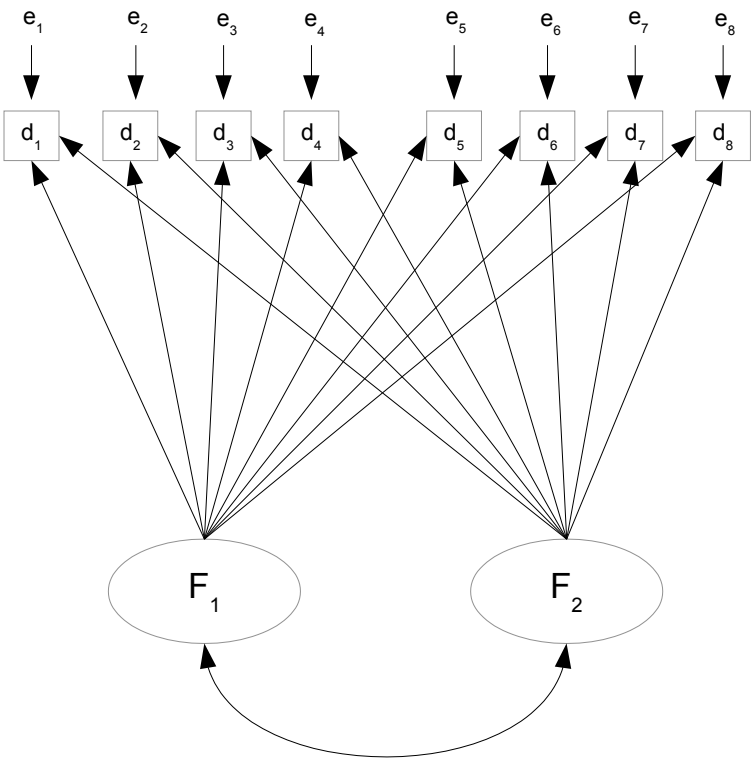
In exploratory factor analysis, the goal is to describe and summarize a data set by explaining a set of observed variables in terms of a smaller number of latent variables (factors). The factors are the reason the observable variables have the correlations they do. Figure 2.1 shows the path diagram of a model with two factors and eight observable variables. A common rule is at least three observable variables for each factor. In general, the more variables for each factor, the better.

The general factor analysis model may be written as follows. Independently for $i = 1, \ldots, n$, let

$$\mathbf{d}_i = \mathbf{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \tag{2.6}$$

where $\mathbf{d}_i$ is a $k \times 1$ observable random vector, $\mathbf{\Lambda}$ is a $k \times p$ matrix of constants, and $\mathbf{F}_i$ ($F$ for factor) is a $p \times 1$ latent random vector with covariance matrix $\mathbf{\Phi}$. The $k \times 1$ vector of error terms $\mathbf{e}_i$ is independent of $\mathbf{F}_i$; it has expected value zero and covariance matrix $\mathbf{\Omega}$,

Figure 2.1: A Two-factor Model




which is almost always assumed to be diagonal[4]. There are no intercepts, and $E(\mathbf{F}_i) = \mathbf{0}$. This is a centered surrogate model (see Section A.6.1). The notation here is consistent with the general two-stage model of Section 1.2, except that there, the dimension of $\mathbf{F}_i$ would be $(p+q) \times 1$. A multivariate normal assumption for $\mathbf{F}_i$ and $\mathbf{e}_i$ is common.

[4]The assumption that $\mathbf{\Omega}$ is diagonal helps with identifiability, and may be traced to what Spearman [60] (1904, p. 273) calls the "Law of the Universal Unity of the Intellective Function," to wit: *Whenever branches of intellectual activity are at all dis-similar, then their correlations with one another appear wholly due to their being all variously saturated with some common fundamental Function (or group of Functions* as well as positive definite. Note that in Figure 2.1, $\mathbf{\Omega}$ being diagonal corresponds to a lack of any curved, double-headed arrows connecting $e_1, \ldots, e_8$. This means that any correlations between observable variables must come from the factors.

To clarify the notation, the model equations for Figure 2.1 are

$$\begin{array}{ccccccc} \mathbf{d}_i & = & \mathbf{\Lambda} & & \mathbf{F}_i & + & \mathbf{e}_i \end{array}$$

$$\begin{pmatrix} d_{i,1} \\ d_{i,2} \\ d_{i,3} \\ d_{i,4} \\ d_{i,5} \\ d_{i,6} \\ d_{i,7} \\ d_{i,8} \end{pmatrix} = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \\ \lambda_{41} & \lambda_{42} \\ \lambda_{51} & \lambda_{52} \\ \lambda_{61} & \lambda_{62} \\ \lambda_{71} & \lambda_{27} \\ \lambda_{81} & \lambda_{82} \end{pmatrix} \begin{pmatrix} F_{i,1} \\ F_{i,2} \end{pmatrix} + \begin{pmatrix} e_{i,1} \\ e_{i,2} \\ e_{i,3} \\ e_{i,4} \\ e_{i,5} \\ e_{i,6} \\ e_{i,7} \\ e_{i,8} \end{pmatrix}. \tag{2.7}$$

The $\lambda_{ij}$ values will be called *factor loadings*. They are essentially regression coefficients linking the factors to the observed variables[5]. The factors $F_{i,1}$ and $F_{i,2}$ are sometimes called *common factors*, because they influence all the observed variables; all the observed variables have them in common. The error terms $e_{i,1}, \ldots, e_{i,8}$ are sometimes called *unique factors*, because each one influences only a single observed variable.

The defining feature of exploratory factor analysis is that it tries to be as unconstrained as possible. The method really wants the data to speak. In Figure 2.1 and in general, there are arrows from all factors to all observed variables.

**Number of factors** The number of factors (symbolized here by $p$) is a fundamental property of a factor analysis model. For example, it determines the number of parameters. It's typically very important to subject matter experts, too. You can always get their attention by asking if something they are talking about is uni-dimensional. For example, is creativity uni-dimensional? Are political attitudes uni-dimensional (primarily just left-right)? In market research, how about attitudes toward a particular product category? Is it just positive-negative? Their eyes will light up.

Of course, there can be lots of factors. For example, Cattell's Sixteen Personality Factor Questionnaire [16] (documented in a 1970 paper by Cattell, Eber and Tatsuoka) is based on factor analyses of a large number of personality test items. They came up with 16 factors.

In a classical factor analysis, the number of common factors is generally not known in advance; it is determined in an exploratory manner. The first guiding principle is a piece of wisdom [39] from Kaiser (1960), who pointed out that for the typical problem involving human behavior or any other complex system, there are probably hundreds of common factors. Including them all in the model is out of the question. The objective should be to come up with a model that includes the most important factors for the variables in the study, and captures the essence of what is going on. Simplicity is important. Other things being more or less equal, the fewer factors the better. I have already mentioned a

[5] In some books, the term "factor loading" is reserved for the correlations between factors and observed variables. When the factors are uncorrelated, the $\lambda_{ij}$ in (2.7) are indeed correlations, and the two common uses of the term coincide.

widely accepted rule of thumb[6] that says there should be at least three observed variables per factor [25]. This sets practical soft upper bound for the number of factors.

To narrow the search for the number of factors, quite a few methods are available. If the parameters are estimated by maximum likelihood, perhaps the most natural approach is to test goodness of fit using the likelihood ratio test (1.18) on page 171, increasing the number of factors until the model fits. This idea has quite a pedigree. It was essentially proposed by Lawley [41] in 1940[7], though he derived a slightly different large-sample chi-squared test. The reasoning is that if we really insist that the error terms are independent of the factors and have a diagonal covariance matrix, the only way that the model can be incorrect is that it does not have enough factors. Thus, any test for goodness of fit is also a test for number of factors.

Hypothesis testing may be attractive, but one thing to bear in mind is Kaiser's observation that in reality, there are probably hundreds of factors. Suppose the true number of factors is very large. Because the power of the likelihood ratio test increases with the sample size, significant lack of fit may be expected for any model with a modest number of factors, even if that model explains most of the non-error variance in an elegant and useful way. Statistically, rejecting the null hypothesis is a correct decision, because the model is wrong. Scientifically, it would be unfortunate. This suggests that while formal tests for lack of fit may be useful, one should not rely on them exclusively.

Another common method [39], and one that continues to be the default in some popular statistical software, is due to Kaiser (1960). Kaiser proposed estimating number of factors by the number of eigenvalues of the correlation matrix that are greater than one. The idea is that even though factor analysis and principal components analysis are different, still, if the correlations among the observed variables arise from $p$ common factors, then the optimality of principal components in explaining variance suggests that $p$ principal components will explain at least as much variance. And then, as in principal components, adding an additional factor that explains less variance than a single variable will not improve the model as a summary of the data.

A variation, called *parallel* analysis [31] is to test whether each eigenvalue is significantly larger than one would expect by chance. The meaning of "chance" is the probability distribution of an (ordered) eigenvalue under the null hypothesis that the variables are uncorrelated. These distributions are approximated by randomly independently permuting the observed data values a large number of times, and calculating the eigenvalues of the correlation matrix for each permutation. A factor is retained if the corresponding ordered eigenvalue is larger than the 95th percentile of the random values.

A graphical alternative called the *scree plot* [15] was proposed by Cattell (1966). Scree is a term from geology. It refers to the pile of rock and debris often found at the foot of a

[6] A rule of thumb is a rule that comes from experience and expert opinion, but is not backed up by hard evidence. The term apparently comes from brewing beer. In the early days before thermometers, the master brewer would stick a thumb in the vat of fermenting hops and stuff, and if the temperature felt right then it was on to the next stage.

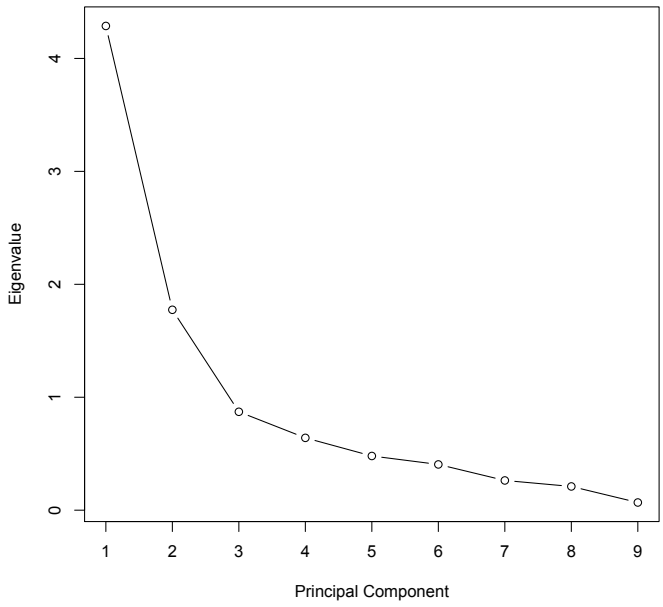
[7] This is the same article where Lawley proposed estimating factor loadings by maximum likelihood. Like many of the procedures that are now standard in multivariate analysis, maximum likelihood factor analysis became practical for most real data sets only after the invention of electronic computers.

mountain cliff or volcano. Scree slopes tend to be concave up, steepest near the cliff and then tailing off. In factor analysis, a scree plot shows the eigenvalues of the correlation matrix, sorted in order of magnitude. It has the numbers $1, \ldots, k$ ($k$ = the number of principal components as well as variables) on the $x$ axis, and the eigenvalues on the $y$ axis. The largest eigenvalue goes with 1, the second largest with 2, and so on. It is very common for the graph to decrease rapidly at first, and then straighten out with a small negative slope for the rest of the way. The point at which the linear trend begins is the estimated number of factors.

Figure 2.2 show a scree plot for the Mind-body data, described in Example 2.1 on page 207. Reading the data and creating the object `pc` with `prcomp` has already been illustrated.

```
> Eigenvalue = pc$sdev^2
> plot(1:9,Eigenvalue,type='b',xlab='Principal Component',xaxp=c(1,9,8))
```

Figure 2.2: Scree Plot for the Mind-Body Data




The linear part of decreasing trend appears to begin with the third eigenvalue, suggesting three factors. There are only nine variables, so the rule of at least three variables per factor would limit us to three factors at most, anyway. Two of the eigenvalues are greater than one, suggesting two factors. There is no requirement that these any of these criteria coincide, and in fact it is reassuring that they are this close.

A final criterion for number of factors is interpretability. What do the factors seem to represent? Typically, the answer is more clear for models with fewer factors. With more and more factors, explanation tends to become increasingly difficult, and the wise factor

analyst will stop at a point where there is still a convincing story to tell. This process is subjective, but reasonable and widely accepted. In a professional paper, one might read something like "A maximum likelihood factor analysis extracted four interpretable factors, accounting for an estimated 72% of the variance in the attitude scales. Table 3 shows the factor loadings . . . "

**Identifiability** The parameters of the general factor analysis model are massively non-identified. This is true even when, as in the example of Figure 2.1, the model passes the test of the parameter count rule. To see this, first observe that the parameters are the unique contents of the matrices $\mathbf{\Phi}$, $\mathbf{\Lambda}$ and $\mathbf{\Omega}$. If two distinct triples $(\mathbf{\Phi}, \mathbf{\Lambda}, \mathbf{\Omega})$ yield the same covariance matrix $\mathbf{\Sigma} = cov(\mathbf{d}_i)$, then the parameters cannot be identified from $\mathbf{\Sigma}$. In practice, that means they can't be identified at all. Calculating,

$$\begin{aligned} cov(\mathbf{d}_i) = \mathbf{\Sigma} &= cov(\mathbf{\Lambda}\mathbf{F}_i + \mathbf{e}_i) \\ &= \mathbf{\Lambda}\mathbf{\Phi}\mathbf{\Lambda}^\top + \mathbf{\Omega}. \end{aligned}$$

The square root matrix of a symmetric matrix is also symmetric, so

$$\begin{aligned} \mathbf{\Lambda}\mathbf{\Phi}\mathbf{\Lambda}^\top + \mathbf{\Omega} &= \mathbf{\Lambda}\,\mathbf{\Phi}^{1/2}\mathbf{I}\mathbf{\Phi}^{1/2}\,\mathbf{\Lambda}^\top + \mathbf{\Omega} \\ &= (\mathbf{\Lambda}\mathbf{\Phi}^{1/2})\mathbf{I}(\mathbf{\Phi}^{1/2\top}\mathbf{\Lambda}^\top) + \mathbf{\Omega} \\ &= (\mathbf{\Lambda}\mathbf{\Phi}^{1/2})\mathbf{I}(\mathbf{\Lambda}\mathbf{\Phi}^{1/2})^\top + \mathbf{\Omega} \\ &= \mathbf{\Lambda}_2\mathbf{I}\mathbf{\Lambda}_2^\top + \mathbf{\Omega} \end{aligned}$$

Unless $\mathbf{\Phi} = cov(\mathbf{F}_i)$ was equal to the identity in the first place, the triple $(\mathbf{I}, \mathbf{\Lambda}_2, \mathbf{\Omega})$ is different from $(\mathbf{\Phi}, \mathbf{\Lambda}, \mathbf{\Omega})$, yet it yields the same $\mathbf{\Sigma}$. This shows that the parameters are not identifiable.

Actually, $\mathbf{\Sigma}$ is produced by infinitely many parameter sets. Let $\mathbf{Q}$ be an arbitrary positive definite covariance matrix for $\mathbf{F}_i$. Then

$$\begin{aligned} \mathbf{\Sigma} &= \mathbf{\Lambda}_2\mathbf{I}\mathbf{\Lambda}_2^\top + \mathbf{\Omega} \\ &= \mathbf{\Lambda}_2\mathbf{Q}^{-\frac{1}{2}}\mathbf{Q}\mathbf{Q}^{-\frac{1}{2}}\mathbf{\Lambda}_2^\top + \mathbf{\Omega} \\ &= (\mathbf{\Lambda}_2\mathbf{Q}^{-\frac{1}{2}})\mathbf{Q}(\mathbf{Q}^{-\frac{1}{2}\top}\mathbf{\Lambda}_2^\top) + \mathbf{\Omega} \\ &= (\mathbf{\Lambda}_2\mathbf{Q}^{-\frac{1}{2}})\mathbf{Q}(\mathbf{\Lambda}_2\mathbf{Q}^{-\frac{1}{2}})^\top + \mathbf{\Omega} \\ &= \mathbf{\Lambda}_3\mathbf{Q}\mathbf{\Lambda}_3^\top + \mathbf{\Omega} \end{aligned} \tag{2.8}$$

No matter what the truth might be, one can make the covariance matrix of the factors absolutely anything, and then adjust the factor loadings to yield exactly the same $\mathbf{\Sigma}$ that is produced by the true parameter values. Note that for multivariate normal data with expected value zero (the usual assumption), all one can ever get from increasing amounts of data is a closer and closer approximation of $\mathbf{\Sigma}$. This means that empirical data cannot help us learn the model parameters. It's not a good situation.

The classical way out of this dilemma is to regard the covariance matrix of the factors as essentially arbitrary, and fix $\mathbf{\Phi} = \mathbf{I}$. The factors are said to be "orthogonal" (at right angles, uncorrelated). They are also standardized, meaning that the (scalar) expected value of each factor is zero, and its variance equals one. This is justified on the grounds of simplicity and ease of interpretation.

Of course, the assumption of uncorrelated factors may be difficult to justify. Furthermore, it is untestable given model (2.6), since all possible covariance matrices for the factors are equally compatible with any set of data. In exploratory factor analysis, the possibility of correlated factors is addressed by transforming the estimates from a model with orthogonal factors into estimates for a model in which the factors are oblique – that is, not at right angles. Accordingly, we will proceed with the orthogonal factor model for the present.

Again, setting $\mathbf{\Phi} = \mathbf{I}$ standardizes the factors as well as making them uncorrelated. The observed variables are standardized as well. For $j = 1, \ldots, k$ and (almost) independently for $i = 1, \ldots, n$ the data we work with are $z_{ij} = \frac{d_{ij} - \bar{d}j}{s_j}$. Thus, each observed variable has variance one as well as mean zero.

In the revised exploratory factor analysis model below, the subscripts $i$ on $\mathbf{z}_i$, $\mathbf{F}_i$ and $\mathbf{e}_i$ have been dropped to reduce notational clutter. Implicitly, everything applies independently for $i = 1, \ldots, n$. The model is

$$\mathbf{z} = \mathbf{\Lambda F} + \mathbf{e}, \text{ where} \tag{2.9}$$

- $\mathbf{z}$ is a $k \times 1$ observable random vector. Each element of $\mathbf{z}$ has expected value zero and variance one.
- $\mathbf{\Lambda}$ is a $k \times p$ matrix of constants.
- $\mathbf{F}$ ($F$ for factor) is a $p \times 1$ latent random vector with expected value zero and covariance matrix $\mathbf{I}_p$.
- The $k \times 1$ vector of error terms $\mathbf{e}$ has expected value zero and covariance matrix $\mathbf{\Omega}$, which is diagonal.

For this model, everything emerges in terms in terms of correlations rather than covariances. This is a virtue, because correlations are easier to interpret. First, $cov(\mathbf{z}_i) = \mathbf{\Sigma} = \mathbf{\Lambda\Lambda}^\top + \mathbf{\Omega}$ is a correlation matrix; correspondingly, estimation and inference will be based on the sample correlation matrix.

**Factor Loadings** Next, consider the matrix of correlations between the factors and the observed variables. Because all the variables are standardized,

$$\begin{aligned} corr(\mathbf{z}, \mathbf{F}) &= cov(\mathbf{z}, \mathbf{F}) \\ &= cov(\mathbf{\Lambda F} + \mathbf{e}, \mathbf{F}) \\ &= \mathbf{\Lambda} cov(\mathbf{F}, \mathbf{F}) + cov(\mathbf{e}, \mathbf{F}) \\ &= \mathbf{\Lambda} cov(\mathbf{F}) + \mathbf{0} \\ &= \mathbf{\Lambda I} \\ &= \mathbf{\Lambda} \end{aligned} \tag{2.10}$$

Thus, the factor loadings are correlations between the observable variables and the factors. In particular, the correlation between observed variable $i$ and factor $j$ is $\lambda_{ij}$. The square of $\lambda_{ij}$ is the reliability[8] of observed variable $i$ as a measure of factor $j$.

**Communality and Uniqueness** Observed variable $i$ (an element of $\mathbf{z}$; the index $i$ goes from $1, \ldots, k$) may be written in scalar form as

$$\begin{aligned} z_i &= \lambda_{i1} F_1 + \cdots + \lambda_{ip} F_p + e_i \\ &= \sum_{j=1}^{p} \lambda_{ij} F_j + e_i, \end{aligned}$$

so that

$$\begin{aligned} Var(z_i) &= Var\left( \sum_{j=1}^{p} \lambda_{ij} F_j + e_i \right) \\ &= \sum_{j=1}^{p} \lambda_{ij}^2 Var(F_j) + Var(e_i) \\ &= \sum_{j=1}^{p} \lambda_{ij}^2 + \omega_i, \end{aligned} \tag{2.11}$$

where $\omega_i = Var(e_i)$ is the $i$th diagonal element of $\mathbf{\Omega}$. Since the observed variables are standardized, we have $1 = \sum_{j=1}^{p} \lambda_{ij}^2 + \omega_i$.

The variance of the observed variable has been split into two components. $\sum_{j=1}^{p} \lambda_{ij}^2$ is the proportion of variance in observed variable $i$ that comes from the common factors. It is called the *communality*. To get the communality of a variable, add up the squares of the factor loadings in the corresponding row of $\mathbf{\Lambda}$. The other component is $\omega_i = 1 - \sum_{j=1}^{p} \lambda_{ij}^2$. It is is what's left over, the part that comes from error. It is called the *uniqueness* of the variable.

It may seem a bit peculiar for the variance of the error term to "know" about the factor loadings, but that's what you get when you standardize the observed variables.

[8] Psychometric reliability. See page 41.

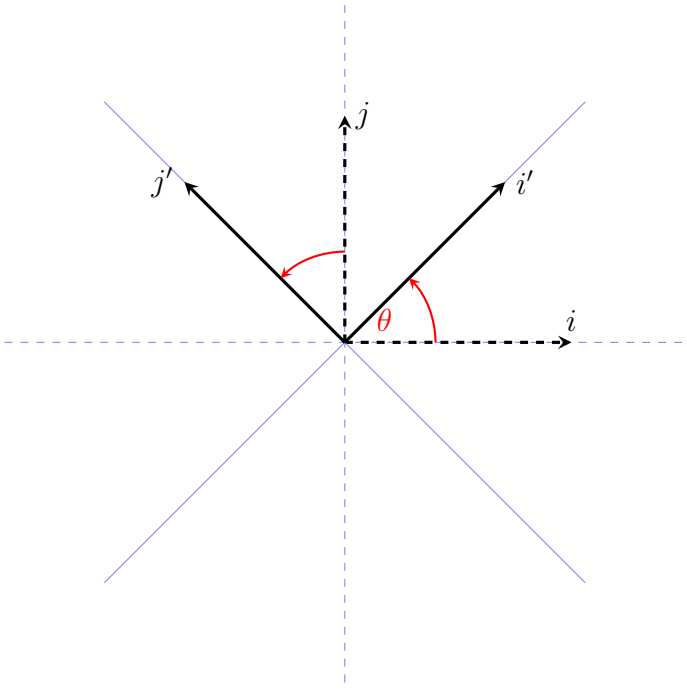
More important is that since the matrix $\mathbf{\Omega}$ is diagonal and its diagonal elements are functions of the $\lambda_{ij}$, the only parameters it contains are factor loadings that are already in $\mathbf{\Lambda}$. The role of $\mathbf{\Omega}$ is to make the diagonal elements of $\mathbf{\Sigma}$ equal one — that is, to make $\mathbf{\Sigma}$ a proper correlation matrix. In the standardized factor analysis model, the only unknown parameters are the factor loadings.

This really is quite nice. Since factor loadings are the correlations between the observable variables and the factors, they could be very informative about the processes driving the data. Squared factor loadings are reliabilities, another important feature of the measurement model. One could also use estimated factor loadings to estimate how much of the variance in each observable variable comes from each factor. All this could reveal what the underlying factors are, and what they mean.

## 2.3 Orthogonal Rotations

Unfortunately, the factor loadings are still not identifiable, so meaningful estimation is still out of the question. This part of the story depends on the idea of a rotation matrix. In Figure 2.3, a basis for $\mathbb{R}^2$ is provided by the unit vectors $\vec{\imath}$ and $\vec{\jmath}$, which are at right angles. These basis vectors are rotated through an angle $\theta$, yielding $\vec{\imath}\,'$ and $\vec{\jmath}\,'$. If a point

Figure 2.3: Rotation




on the plane is denoted in terms of $\vec{\imath}$ and $\vec{\jmath}$ by $(x, y)$, its position in terms of the rotated

basis vectors is

$$\begin{aligned} x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta. \end{aligned}$$

These are the well-known *equations of rotation*. They may be written in matrix form as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{R} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{2.12}$$

Using the identities $\cos(-\theta) = \cos\theta$ and $\sin(-\theta) = -\sin\theta$, one obtains a matrix that rotates the axes back to their original position.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{R}^\top \begin{pmatrix} x' \\ y' \end{pmatrix}. \tag{2.13}$$

As the notation indicates, the matrix that reverses the rotation is the transpose of the original rotation matrix. Verifying that it's also the inverse,

$$\begin{aligned} \mathbf{R}\mathbf{R}^\top &= \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \\ &= \begin{pmatrix} \cos^2\theta + \sin^2\theta & -\cos\theta\sin\theta + \sin\theta\cos\theta \\ -\sin\theta\cos\theta + \cos\theta\sin\theta & \sin^2\theta + \cos^2\theta \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbf{I}. \end{aligned}$$

So in two dimensions, the transpose of a rotation matrix is also its inverse. This fact holds in higher dimension as well. A $p \times p$ matrix $\mathbf{R}$ satisfying $\mathbf{R}^{-1} = \mathbf{R}^\top$ is called an *orthogonal matrix*, because the columns and rows are orthonormal vectors. Geometrically, pre-multiplication by an orthogonal matrix corresponds to a rotation or possibly a reflection in $p$-dimensional space. If you think of a set of factors $\mathbf{F}$ as a set of axes or underlying dimensions, then $\mathbf{RF}$ is a rotation (or reflection) of the factors. Call it an *orthogonal* rotation, because the factors remain uncorrelated — at right angles.

Rotation matrices are another source of non-identifiability. Returning to the standardized factor model, the covariance matrix of the observed data vector $\mathbf{z}$ is

$$\begin{aligned} \boldsymbol{\Sigma} &= \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Omega} \\ &= \boldsymbol{\Lambda}\mathbf{R}^\top\mathbf{R}\boldsymbol{\Lambda}^\top + \boldsymbol{\Omega} \\ &= (\boldsymbol{\Lambda}\mathbf{R}^\top)(\boldsymbol{\Lambda}\mathbf{R}^\top)^\top + \boldsymbol{\Omega} \\ &= \boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_2^\top + \boldsymbol{\Omega} \end{aligned}$$

That is, infinitely many rotation matrices produce the same $\boldsymbol{\Sigma}$, even though the factor loadings in $\boldsymbol{\Lambda}_2 = \boldsymbol{\Lambda}\mathbf{R}^\top$ can be very different for different $\mathbf{R}$ matrices.

Post-multiplication of $\mathbf{\Lambda}$ by $\mathbf{R}^\top$ is often called "rotation of the factors," for the following reason.

$$\begin{aligned} \mathbf{z} &= \mathbf{\Lambda F} + \mathbf{e} \\ &= (\mathbf{\Lambda R}^\top)(\mathbf{RF}) + \mathbf{e} \\ &= \mathbf{\Lambda}_2 \mathbf{F}' + \mathbf{e}. \end{aligned} \tag{2.14}$$

$\mathbf{F}' = \mathbf{RF}$ is a set of *rotated* factors. All rotations of the factors produce the same covariance matrix of the observable data.

In addition, all sets of rotated factors account for the same proportion of variance. To see this, recall that $\sum_{j=1}^{p} \lambda_{ij}^2$, the formula for the communality of observed variable $i$, instructs us to add up the squares of the factor loadings in row $i$ of $\mathbf{\Lambda}$. This equals the $ith$ diagonal of element of $\mathbf{\Lambda\Lambda}^\top$. Applying a rotation,

$$\begin{aligned} \mathbf{\Lambda}_2\mathbf{\Lambda}_2^\top &= (\mathbf{\Lambda R}^\top)(\mathbf{\Lambda R}^\top)^\top \\ &= \mathbf{\Lambda R}^\top \mathbf{R\Lambda}^\top \\ &= \mathbf{\Lambda\Lambda}^\top, \end{aligned} \tag{2.15}$$

so that rotation does not affect the proportions of variance explained by the common factors.

Confronted with this unpleasant situation, the exploratory factor analyst asks a question. Since all rotations of the factors explain the data equally well, why not just pick a good one? Here's an outline of the strategy.

1. Place some restrictions on the factor loadings, so that the only rotation matrix that preserves the restrictions is the identity matrix[9]. For example, $\lambda_{ij} = 0$ for $j > i$. There are other sets of restrictions that work — for example, forcing $\mathbf{\Lambda}^\top \mathbf{\Omega}^{-1} \mathbf{\Lambda}$ to be diagonal.

2. Generally, the restricted factor loadings may not make sense in terms of the data. Don't worry about it.

3. Estimate the loadings, perhaps by maximum likelihood. Other methods are available, but less commonly used than in the past.

4. Now apply a rotation, without any restriction on the resulting factor loadings. All (orthogonal) rotations result in the same maximum value of the likelihood function. That is, the maximum is not unique. Again, don't worry about it.

5. Pick a rotation that results in a simple pattern in the factor loadings, one that is easy to interpret.

The first and last steps require further discussion. The first step is to place restrictions on the factor loadings. Consider the restriction $\lambda_{ij} = 0$ for $j > i$. This means that observed

[9] This statement will require a bit of qualification, but it's the right idea.

variable one comes only from factor one, observed variable two comes only from factors one and two, observed variable three comes only from factors one, two and three – and so on. This pattern might be plausible for some sets of variables, but not in general. Carry on.

As an illustration, consider the case of two factors. In the path diagram of Figure 2.1, the straight arrow from $F_2$ to $d_1$ is missing. Also, the curved, double-headed arrow between $F_1$ and $F_2$ is missing, because the factors are orthogonal. In the model equations (2.7), the only restriction is $\lambda_{12} = 0$. Maintaining that restriction under rotation means

$$\begin{pmatrix} \lambda_{11} & 0 \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \\ \lambda_{41} & \lambda_{42} \\ \lambda_{51} & \lambda_{52} \\ \lambda_{61} & \lambda_{62} \\ \lambda_{71} & \lambda_{27} \\ \lambda_{81} & \lambda_{82} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} \lambda'_{11} & 0 \\ \lambda'_{21} & \lambda'_{22} \\ \lambda'_{31} & \lambda'_{32} \\ \lambda'_{41} & \lambda'_{42} \\ \lambda'_{51} & \lambda'_{52} \\ \lambda'_{61} & \lambda'_{62} \\ \lambda'_{71} & \lambda'_{27} \\ \lambda'_{81} & \lambda'_{82} \end{pmatrix}$$

Focusing on the zero in the right-hand side, we have

$$\begin{aligned} & \lambda_{11}\sin\theta + 0\cos\theta = 0 \\ \Rightarrow \quad & \lambda_{11}\sin\theta = 0 \\ \Rightarrow \quad & \sin\theta = 0 \text{ (provided } \lambda_{11} \neq 0). \end{aligned}$$

Therefore, the angle of rotation $\theta$ equals 0, or $\pi$, or $2\pi$, or $3\pi$, or .... For $\theta = 0$ or any even multiple of $\pi$, $\cos\theta = 1$, and the rotation matrix is the identity. For $\theta = \pi$ or any odd multiple of $\pi$, $\cos\theta = -1$, and the rotation matrix is minus the identity. This reverses the signs of all the factor loadings.

There are two more orthogonal matrices that preserve the constraint $\lambda_{12} = 0$. They are $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. The first matrix reverses the signs of the first column of $\mathbf{\Lambda}$, but leaves the second column alone. The second matrix reverses the signs of the second column of $\mathbf{\Lambda}$ while leaving the first column alone. These represent reflections. The set of orthogonal matrices corresponds to the set of all possible reflections and rotations about the origin.

This shows that the restriction $\lambda_{12} = 0$ does not quite make the remaining factor loadings identifiable from the correlation matrix. We have located four distinct sets of parameter values that yield exactly the same correlation matrix for the observed data vector. On the other hand, these multiple solutions will not produce trouble in the numerical search for the MLE, because they are separated in the parameter space. The search will find just one of them, or it will wander off into nowhere, depending on the starting value and the topography of the likelihood function. It does not really matter which one we find. The plan is to apply a rotation later to find a more interpretable set of factor loadings, so the meaning of the parameter estimates is not an issue at this point.

To see what happens in higher dimension, it is enough to examine the case of $p = 3$. Denoting the orthogonal matrix by $\mathbf{R} = [r_{ij}]$ and insisting that it preserve the constraints $\lambda_{ij} = 0$ for $j > i$, we require

$$\begin{pmatrix} \lambda_{11} & 0 & 0 \\ \lambda_{21} & \lambda_{22} & 0 \\ \lambda_{31} & \lambda_{32} & \lambda_{33} \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} \lambda'_{11} & 0 & 0 \\ \lambda'_{21} & \lambda'_{22} & 0 \\ \lambda'_{31} & \lambda'_{32} & \lambda'_{33} \\ \vdots & \vdots & \vdots \end{pmatrix} \tag{2.16}$$

Carrying out the row by column multiplications that yield the three zeros, conclude $r_{12} = r_{13} = r_{23} = 0$. Then use the fact that $\mathbf{R}\mathbf{R}^\top = \mathbf{0}$. Conclude that $r_{21} = r_{31} = r_{32} = 0$, and that

$$\begin{pmatrix} r_{11}^2 & 0 & 0 \\ 0 & r_{22}^2 & 0 \\ 0 & 0 & r_{33}^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

So, the off-diagonal elements of $\mathbf{R}$ are zero, and the diagonal elements are either plus or minus one, with entries of minus one representing reflections. This is how it goes in general, with $2^p$ different orthogonal matrices preserving the restriction $\lambda_{ij} = 0$ for $j > i$. The result is $2^p$ distinct minima of the minus log-likelihood function, all with the same value at the local minimum. Again, no numerical difficulties are created, because the multiple minima are separated in the parameter space, and the search for the MLE will only go down one of the holes.

The restriction $\lambda_{ij} = 0$ for $j > i$ is fairly easy to understand, but the restriction most used in practice is for $\mathbf{J} = \mathbf{\Lambda}^\top \mathbf{\Omega}^{-1} \mathbf{\Lambda}$ to be diagonal. In *Factor analysis as a statistical method* [42], Lawley and Maxwell (1971) show how this way of restricting $\mathbf{\Lambda}$ allows an efficient iterative solution of the equations obtained by differentiating the log likelihood and setting all the derivatives to zero.

Full details of Lawley's method will not be given here, but a few remarks are in order. First, since $\mathbf{\Lambda}$ is $k \times p$, the matrix $\mathbf{J}$ is $p \times p$. It is also symmetric, so insisting it be diagonal places $p(p-1)/2$ restrictions on $\mathbf{\Lambda}$. The restriction $\lambda_{ij} = 0$ for $j > i$ also induces a little triangle of zeros, as in (2.16); there are $p(p-1)/2$ of them, so the two methods impose the same number of restrictions. This is useful when it comes to counting degrees of freedom.

Second, let the $p \times p$ matrix $\mathbf{R}$ be a restricted kind of orthogonal matrix, a diagonal matrix, with values of plus or minus one on the diagonal. Any diagonal element of $\mathbf{R}$ equal to minus one reverses the signs of all the loadings in the corresponding column of $\mathbf{\Lambda}$. That's a reflection.

Replacing $\mathbf{\Lambda}$ with $\mathbf{\Lambda R}$,

$$\begin{aligned} (\mathbf{\Lambda R})^\top \mathbf{\Omega}^{-1} \mathbf{\Lambda R} &= \mathbf{R}^\top \mathbf{\Lambda}^\top \mathbf{\Omega}^{-1} \mathbf{\Lambda R} \\ &= \mathbf{R}^\top \mathbf{J R} \\ &= \mathbf{J}, \end{aligned}$$

since $\mathbf{J}$ is diagonal. Therefore, as in the simpler case of $\lambda_{ij} = 0$ for $j > i$, there are $2^p$

different $\mathbf{\Lambda}$ matrices that satisfy the constraint, and also produce the same $\mathbf{\Sigma} = corr(\mathbf{z})$. Again, there are $2^p$ corresponding minima of the minus log likelihood function.

We need some notation. The initial (restricted) maximum likelihood estimates of the factors will be denoted by $\widetilde{\lambda}_{ij}$, while $\widehat{\lambda}_{ij}$ will be reserved for the final estimates after applying a rotation. In matrix form, $\widehat{\mathbf{\Lambda}} = \widetilde{\mathbf{\Lambda}}\mathbf{R}^\top$.

**The Heywood case** It is by no means guaranteed that the numerical search for the MLE will stop at a point that is in the parameter space. In fact, it is surprisingly common for the estimates to violate the inequality constraints of the model, as in the negative variance Example 1.5.1. Because the observed variables are standardized, an application of invariance to (2.11) yields $Var(z_i) = 1 = \sum_{j=1}^{p} \widetilde{\lambda}_{ij}^2 + \widetilde{\omega}_i$. A negative $\widetilde{\omega}_i$ would thus induce $\sum_{j=1}^{p} \widetilde{\lambda}_{ij}^2 > 1$, an estimated communality greater than one. Since the communality is the proportion of variance that comes from the common factors, this is a bit of a problem. It is sometimes called a *Heywood case*. Or sometimes, $\sum_{j=1}^{p} \widetilde{\lambda}_{ij}^2 = 1$ is called a Heywood case, and $\sum_{j=1}^{p} \widetilde{\lambda}_{ij}^2 > 1$ is called an *ultra-Heywood* case. You have to feel sorry for the user, and also for Mr. Heywood, since his name has been so often cursed[10]. Rotation will not solve this problem, because communality is unaffected by rotation (2.15).

Provided that an acceptable MLE has been located, the result is a set of estimated factor loadings that might be interpretable if the restrictions on $\mathbf{\Lambda}$ made sense in terms of the problem, but not otherwise. With respect to the original parameter space (without the restrictions), the set of estimated factor loadings we have found is only one of an uncountable infinity, all with the same value of the (minus log) likelihood function. There is one such set of factor loadings for every $p \times p$ orthogonal matrix. The last step in the 5-step recipe given earlier is to pick a good one, and go with that.

In the final step, the factors are rotated, so that $\widehat{\mathbf{\Lambda}} = \widetilde{\mathbf{\Lambda}}\mathbf{R}^\top$ has a "simple structure" that is easy to interpret. The concept of simple structure is not precisely defined, which in the past made factor analysis a bit subjective. There were many fruitless arguments in which researchers came to different conclusions because they used different rotations, even though they all claimed to have rotated to "simple structure."

It is helpful to lift the criteria for simple structure from Harman [28], 1976, p. 98; Harman takes them from Thurstone's highly influential (1947) book [64], which I cannot get my hands on right now[11]. Here are Thurstone's criteria for simple structure, using our notation.

1. Each row of $\widehat{\mathbf{\Lambda}}$ should have at least one zero.

[10] Heywood [29] gets the blame because of a 1931 paper in which he proves, among other things, that there can be legitimate correlation matrices that would imply a communality greater than one. It's one of the "cases" he considers, so I assume that's why they call it a Heywood case. From the perspective of this book, the factor analysis model implies inequality constraints that are not true of all positive definite correlation matrices. There is no mystery here.

[11] I am writing this in the Spring of 2021. The covid-19 pandemic is going strong, and the library is closed. One could not ask for a better excuse.

2. Each column of $\widehat{\mathbf{\Lambda}}$ should have at least $p$ zeros, where $p$ is the number of factors.

3. For every pair of columns of $\widehat{\mathbf{\Lambda}}$, there should be several variables with loadings that vanish in one column but not in the other.

4. For every pair of columns of $\widehat{\mathbf{\Lambda}}$, a large proportion of the variables should have loadings in both columns that are small in absolute value, when there are four or more factors.

5. For every pair of columns of $\widehat{\mathbf{\Lambda}}$, there should be only a small number of variables with non-vanishing loadings in both columns.

There are various ways of trying to approximate these goals in an objective manner. The methods are all iterative, taking a number of steps to approach some criterion. The most popular rotation method is *varimax* rotation. As described by Harman [28], the initial version of varimax was based on the following reasonable idea. To move the loadings in a particular column of $\widehat{\mathbf{\Lambda}}$ toward zero or $\pm 1$, maximize the sample variance of the squared factor loadings. That is, maximize

$$\frac{1}{k}\sum_{i=1}^{k}\left(\widehat{\lambda}_{ij}^2\right)^2 - \frac{1}{k^2}\left(\sum_{i=1}^{k}\widehat{\lambda}_{ij}^2\right)^2$$

for column $j$. Adding up the columns yields the criterion

$$\frac{1}{k}\sum_{j=1}^{p}\sum_{i=1}^{k}\widehat{\lambda}_{ij}^4 - \frac{1}{k^2}\sum_{j=1}^{p}\left(\sum_{i=1}^{k}\widehat{\lambda}_{ij}^2\right)^2 .$$

In empirical tests, maximizing this criterion often yielded results that were less pleasing than a subjective rotation. In particular, the loadings near plus and minus one tended to be concentrated in just a few columns, which is inconsistent with properties three through five of simple structure given above. Not bothering with the intuitive justification (see Harman [28], p. 291), the work-around was to give somewhat less weight to factor loadings from variables with higher communality. This is accomplished by dividing by the communalities. The whole expression is also multiplied by $k^2$, which does not affect the point where the maximum occurs. The resulting criterion is

$$V = k\sum_{j=1}^{p}\sum_{i=1}^{k}\left(\frac{\widehat{\lambda}_{ij}}{\widehat{h}_i}\right)^4 - \sum_{j=1}^{p}\left(\sum_{i=1}^{k}\frac{\widehat{\lambda}_{ij}^2}{\widehat{h}_i^2}\right)^2 , \tag{2.17}$$

where $\widehat{h}_i^2 = \sum_{j=1}^{p}\widehat{\lambda}_{ij}^2$. That's the communality of variable $i$, the proportion of variance explained by the common factors. Another way to express (2.17) is to say the squared (estimated) factor loadings are adjusted so that each row adds to one. This is sometimes called "Kaiser normalization" after the guy who came up with the idea of varimax.

Expression (2.17) is not directly maximized over the factor loadings. Rather, the process starts with an initial set of estimated loadings (say, from constrained maximum

likelihood), and then rotates the factors two at a time as in Figure 2.3, picking the angle of rotation $\theta$ that maximizes $V$ at each step. An iteration consists of going through $p-1$ steps, rotating factors 1 and 2, factors 2 and 3, and so on[12]. The process contines to iterate until $V$ does not increase any more, to some specified number of decimal places. You might see a message like "Varimax converged in 5 iterations."

Varimax solutions are not unique. Suppose the rotation matrix $\mathbf{R}$ yields a solution $\widehat{\mathbf{\Lambda}} = \widetilde{\mathbf{\Lambda}}\mathbf{R}^\top$ that minimizes the varimax criterion (2.17). Let $\mathbf{M}$ be a $p \times p$ diagonal matrix, with each diagonal element equal to plus or minus one. $\mathbf{M}$ is an orthogonal matrix, and so is $\mathbf{R}^\top\mathbf{M}$. Therefore, $\widehat{\mathbf{\Lambda}}\mathbf{M} = \widetilde{\mathbf{\Lambda}}\mathbf{R}^\top\mathbf{M}$ is another orthogonal rotation/reflection. In $\widehat{\mathbf{\Lambda}}\mathbf{M}$, the columns of $\widehat{\mathbf{\Lambda}}$ are multiplied by the corresponding diagonal elements of $\mathbf{M}$. Potentially, this reverses the signs of the coefficients in one or more columns of $\widehat{\mathbf{\Lambda}}$. There is no effect on the value of the varimax criterion (2.17), because the varimax criterion is based on *squared* factor loadings. With $p$ factors, the varimax criterion has $2^p$ minima, as each element of $\mathbf{M}$ switches between $\pm 1$. The solution obtained from software will depend on where the numerical search happens to start.

Perhaps surprisingly, this does not make interpretation of results more difficult.. Reflecting a factor (multiplying by minus one) reverses the signs of the correlations between that factor and all the observable variables. It also directly reverses the *meaning* of the factor. So for example (recalling that the factors are standardized), if a factor represents wealth, then minus the factor represents poverty. After a varimax rotation, factors may be reflected at will if that makes it easier to think about the results.

In practice, varimax rotation tends to maximize the squared loading of each observable variable with just one underlying factor. In the typical varimax solution, each variable has a big loading on (correlation with) just one of the factors, and small loadings on the rest. It's usually not hard to look at the loadings and decide what the factors mean. Naming the factors is a fun game that is easy to play. In fact, the whole exercise is so satisfying that many casual users of exploratory factor analysis do not go beyond an orthogonal solution with a varimax rotation. Even the most casual class of users, who carry out a principal components analysis thinking it's factor analysis, often apply a varimax rotation to the correlations between variables and components, and are very happy with the result. Later, it will be seen that applying a rotation to principal components is really not such a bad idea, since the rotated components explain the same total amount of variance as the original set, and are easier to talk about.

**Exploratory factor analysis of the Mind-body data** We will start by re-reading the Mind-body data for the described in Example 2.1.

```
> # Factor analysis with orthogonal rotation
> rm(list=ls())
> bodymind = read.table('http://www.utstat.toronto.edu/~brunner/openSEM/data/bodymind.data.txt')
> dat = as.matrix(bodymind[,2:10]) # Omit sex. dat is now a numeric matrix.
> help(factanal)
```

[12]The result would seem to depend on the order in which the factors are sorted. I don't know of any proof that all orderings of factors yield the same varimax solution, but I expect that they are all pretty similar.

The built-in `factanal` function does maximum likelihood factor analysis with orthogonal factors. The first argument is an input data matrix, covariance matrix or correlation matrix. The second argument is the number of factors. How many factors should we have? We know from the principal components analysis that two eigenvalues of the correlation matrix are greater than two. That's one reason to try fitting a two-factor model. Another reason is that some of the variables are educational measurements (mental), while the rest are physical measures. Since the input comes from two distinct domains, I would expect two factors[13]. We'll start with two factors. Because there are only nine variables, the guideline of at least three variables per factor implies a maximum of three factors. The scree plot in Figure 2.2 suggests three factors, so we'll definitely consider a three-factor model after this.

```
> # Maximum likelihood, varimax, 2 factors
> fit2 = factanal(dat,factors=2) # rotation='varimax' is the default
> fit2

Call:
factanal(x = dat, factors = 2)

Uniquenesses:
progmat  reason  verbal headlng headbrd headcir   bizyg  weight  height
  0.616   0.274   0.264   0.324   0.618   0.016   0.473   0.577   0.633

Loadings:
        Factor1 Factor2
progmat 0.181   0.592
reason  0.124   0.843
verbal  0.160   0.843
headlng 0.806   0.161
headbrd 0.618
headcir 0.963   0.238
bizyg   0.687   0.236
weight  0.638   0.129
height  0.588   0.144

               Factor1 Factor2
SS loadings      3.257   1.948
Proportion Var   0.362   0.216
Cumulative Var   0.362   0.578

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 87.55 on 19 degrees of freedom.
The p-value is 8.97e-11
```

First, look at the (estimated) factor loadings. We'll go over other details later. Notice that the loading for head breadth on Factor 2 appears to be missing. This happens because the matrix of factor loadings is a special kind of R object with its own elaborate print

[13]This kind of reasoning often works. To steal a joke from Tom Lehrer, factor analysis is like a sewer. What you get out of it depends on what you put into it.

method. By default, loadings below 0.1 in absolute value are not displayed. The objective is to make the loadings easier to understand by hiding trivial ones. As an SPSS jock in a past life, I am more used to loadings under 0.3 being blanked out, which works better in the present case. The cutoff is controlled by the `cutoff` option on `print`, as shown below.

```
> L2 = fit2$loadings
> print(L2,cutoff=0.3)

Loadings:
        Factor1 Factor2
progmat         0.592
reason          0.843
verbal          0.843
headlng 0.806
headbrd 0.618
headcir 0.963
bizyg   0.687
weight  0.638
height  0.588

               Factor1 Factor2
SS loadings      3.257   1.948
Proportion Var   0.362   0.216
Cumulative Var   0.362   0.578
```

Looking at this, it's a little difficult to believe that `L2` is just a matrix.

```
> is.matrix(L2)
[1] TRUE
> dim(L2)
[1] 9 2
```

So `L2` really just a $9 \times 2$ matrix. The little table under the loadings is produced automatically by the print method. It will be discussed presently.

With the small loadings hidden, it is easy to see that the mental measurements (`progmat`, `reason` and `verbal`) load primarily on the second factor, while the other variables (all physical) load on the first factor. One could name Factor One "Physical" and Factor Two "Mental." Or perhaps they could me named "Size" and "Smarts." This is a typical case. Often, the meaning of the factors jumps out at you, and they are easy to name. This is because of the varimax rotation. Unrotated factor loadings are often very difficult to interpret.

At the bottom of the output displayed for the `fit2` object, there is a chi-squared test for goodness of fit. The $p$-value is very small, indicating that the model does not fit well at all. For this reason and also for other reasons mentioned earlier, we need to look at a three-factor model. First, however, let's back up and look at some details, to clarify what the software is doing.

We will begin with an unrotated two-factor model, displaying all the factor loadings[14]. Note how the `cutoff=0` option on `print(fit2a)` is passed down to the printing of the factor loadings.

```
> fit2a = factanal(dat,factors=2,rotation='none')
> print(fit2a,cutoff=0)

Call:
factanal(x = dat, factors = 2, rotation = "none")

Uniquenesses:
progmat  reason  verbal headlng headbrd headcir   bizyg  weight  height
  0.616   0.274   0.264   0.324   0.618   0.016   0.473   0.577   0.633

Loadings:
        Factor1 Factor2
progmat  0.335   0.521
reason   0.348   0.778
verbal   0.383   0.768
headlng  0.820  -0.064
headbrd  0.600  -0.149
headcir  0.992  -0.033
bizyg    0.725   0.040
weight   0.649  -0.049
height   0.605  -0.021

               Factor1 Factor2
SS loadings      3.708   1.497
Proportion Var   0.412   0.166
Cumulative Var   0.412   0.578

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 87.55 on 19 degrees of freedom.
The p-value is 8.97e-11
```

For an orthogonal factor model, squared factor loadings are components of explained variance. If you square the factor loadings and add, the row totals are commonalities, or proportions of variance explained by the common factors. The column totals are amounts of variance explained by each factor. The `addmargins` function is a convenient way to add row and column totals to a matrix.

```
> L2a = fit2a$loadings
> CompVar = addmargins(L2a^2) # Squared factor loadings are components of variance
> round(CompVar,3)
        Factor1 Factor2   Sum
progmat   0.112   0.271 0.384
reason    0.121   0.605 0.726
verbal    0.147   0.589 0.736
headlng   0.672   0.004 0.676
```

[14] They are estimated factor loadings, of course. Everything here is an estimate.

```
headbrd   0.360   0.022 0.382
headcir   0.983   0.001 0.984
bizyg     0.526   0.002 0.527
weight    0.421   0.002 0.423
height    0.366   0.000 0.367
Sum       3.708   1.497 5.205
```

Factor One explains a whopping 98.3% of the variance in head circumference, and 52.6% of the variance head length. Maybe the unrotated version it could be called "Head size" rather than just "Size." Anyway, the last column of numbers contains the commonalities. Checking that communality plus uniqueness equals one,

```
> fit2a$uniquenesses + CompVar[1:9,3] # Should equal ones
  progmat    reason    verbal   headlng   headbrd   headcir     bizyg    weight    height
0.9999884 0.9999994 1.0000003 0.9999984 0.9999912 1.0000000 1.0000001 1.0000041 1.0000124
```

Close enough. The column totals of `CompVar` are the amounts of variance explained by each factor, and indeed they match `SS loadings` in the display of `fit2a`. To convert these amounts of explained variance to proportions, divide by the number of variables (since the variables are standardized, the total amount of variance to explain is $k$, the number of variables). This yields the `Proportion Var` line. `Cumulative Var` is self-explanatory.

Notice that the `Proportion Var` lines are different for `fit2` (the rotated solution) and `fit2a` (unrotated). Rotation affects the amounts of variance explained by the factors. However, rotation does not affect the commonalities. So, it does not affect the uniquenesses or the total amount of variance explained.

To obtain the unrotated solution by maximum likelihood, `factanal` uses Lawley's [41] constraint that $\widetilde{\mathbf{\Lambda}}^\top \widetilde{\mathbf{\Omega}}^{-1} \widetilde{\mathbf{\Lambda}}$ must be diagonal[15]. Checking that the unrotated solution obeys this restriction,

```
> Omegahat = diag(fit2a$uniquenesses) # Diagonal matrix of uniquenesses little-omega-hat
> J = t(L2a) %*% solve(Omegahat) %*% L2a
> round(J,10)
         Factor1  Factor2
Factor1 69.10492 0.000000
Factor2  0.00000 5.002347
```

It's diagonal, as advertised. There is no reason to expect the rotated loadings to obey this constraint. Using the fact that $\widehat{\mathbf{\Omega}}$ is unaffected by rotation,

```
> J = t(L2) %*% solve(Omegahat) %*% L2; round(J,10)
         Factor1   Factor2
Factor1 64.36786 16.769564
Factor2 16.76956  9.739412
```

[15] Remember that $\widetilde{\mathbf{\Lambda}}$ and $\widetilde{\mathbf{\Omega}}$ are the initial estimates before rotation, obtained by constrained maximum likelihood. Of course, $\widetilde{\mathbf{\Omega}} = \widehat{\mathbf{\Omega}}$, because rotation does not affect the uniquenesses.

It is standard to specify the rotation when fitting the model, as in `fit2`. However, one may also fit a model without rotation as we have done here, and then rotate the factors as a separate step. R has a built-in `varimax` function (and also `promax`, which will not be discussed).

```
> varimax(L2a)
$loadings

Loadings:
        Factor1 Factor2
progmat 0.181   0.592
reason  0.124   0.843
verbal  0.160   0.843
headlng 0.806   0.161
headbrd 0.618
headcir 0.963   0.238
bizyg   0.687   0.236
weight  0.638   0.129
height  0.588   0.144

               Factor1 Factor2
SS loadings      3.257   1.948
Proportion Var   0.362   0.216
Cumulative Var   0.362   0.578

$rotmat
           [,1]      [,2]
[1,]  0.9623418 0.2718422
[2,] -0.2718422 0.9623418
```

The loadings are identical to the rotated factor matrix from `fit2` on page 229. The `varimax` function returns a list with two items, the factor loadings and the rotation matrix that maximizes the varimax criterion (2.17). The same matrix is also available as `fit2$rotmat`. Note that in our notation, `rotmat` is $\mathbf{R}^\top$, not $\mathbf{R}$.

**More factors** Next, we will try a model with three factors, as suggested by the scree plot and the highly significant chi-squared test for the the two-factor model. The `sort=TRUE` option re-orders the variables in the table of factor loadings, in an attempt to make the output easier to read.

```
> # Try a 3-factor model
> fit3 = factanal(dat,factors=3)
> print(fit3,cutoff=0.30, sort=TRUE)

Call:
factanal(x = dat, factors = 3)

Uniquenesses:
progmat  reason  verbal headlng headbrd headcir   bizyg  weight  height
  0.606   0.215   0.309   0.005   0.268   0.094   0.256   0.560   0.565
```

```
Loadings:
        Factor1 Factor2 Factor3
headbrd  0.852
bizyg    0.787
weight   0.523           0.387
progmat          0.583
reason           0.879
verbal           0.811
headlng                  0.959
headcir  0.631           0.669
height   0.465           0.445

               Factor1 Factor2 Factor3
SS loadings      2.318   1.945   1.859
Proportion Var   0.258   0.216   0.207
Cumulative Var   0.258   0.474   0.680

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 30.89 on 12 degrees of freedom.
The p-value is 0.00205
```

This is more challenging. Factor 2 still definitely represents the mental measurements, while Factors 1 and 3 seem to reflect different aspects of head size. Factor 1 loads most highly on head breadth, followed closely by bizygomatic breadth, which is basically how far apart the eyes are. One could call Factor 1 "Face width." Factor 3 loads primarily on head length, and that's what it appears to be. Head circumference, which includes both face width and led length, loads about equally on the two factors. This makes pretty good sense. Height and weight, aspects of overall body size, also load on both of the head factors, though not as highly. We can live with this.

The chi-squared test for lack of fit is still significant, though the $p$-value of 0.00205 is a lot closer to 0.05 than 8.97e-11 is. Strictly speaking, the model still does not fit. Let's check the degrees of freedom. There are nine observed variables, so the correlation matrix $\mathbf{\Sigma}$ has 9(9-1)/2 = 36 unique elements. There would be 36 covariance structure equations in $9 \times 3 = 27$ unknown parameters, except that some of the unknown factor loadings are functions of the others, because of the constraint that $\mathbf{\Lambda}^\top \mathbf{\Omega}^{-1} \mathbf{\Lambda}$ is diagonal. There are $p(p-1)/2 = 3$ such functional connections among the factor loadings. Thus, the degrees of freedom for the test of fit should be $36 - 27 + 3 = 12$. That's what the printout says; okay.

Which model is better, the two-factor or the three-factor? The two-factor model explains an estimated 58% of the total variance, while the three-factor model explains an estimated 68%. Since there are nine observed variables, that 10% gain is worth about one variable. It's borderline. The two-factor model is a bit easier to talk about, but the three-factor model makes sense too. The three-factor model fits better, but it still does not fit in an absolute sense. How about a four-factor model? We would be violating the reasonable rule of at least three variables per factor, and we are almost running out of degrees of freedom, but it's worth a try.

```
> # A four-factor model?!
> print( factanal(dat,factors=4), cutoff=0.30, sort=TRUE)

Call:
factanal(x = dat, factors = 4)

Uniquenesses:
progmat  reason  verbal headlng headbrd headcir   bizyg  weight  height
  0.580   0.216   0.305   0.005   0.005   0.109   0.248   0.356   0.437

Loadings:
        Factor1 Factor2 Factor3 Factor4
bizyg    0.633           0.527
weight   0.761
height   0.672
progmat          0.599
reason           0.872
verbal           0.813
headbrd                  0.957
headlng  0.423                   0.886
headcir  0.555           0.418   0.582

               Factor1 Factor2 Factor3 Factor4
SS loadings      2.037   1.946   1.433   1.321
Proportion Var   0.226   0.216   0.159   0.147
Cumulative Var   0.226   0.443   0.602   0.749

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 8.98 on 6 degrees of freedom.
The p-value is 0.175
```

Now it seems that Factor 1 is overall body size, Factor 2 is educational test performance (or "intelligence," if you want to walk down that dark path), Factor 3 is face width, and Factor 4 is head length. Furthermore, the model technically fits. As for choice among the models, it's really a judgement call. As I see it, the clearest part of the picture is that the mental measurements form one cluster, and the physical measurements form another cluster, but one that may be more differentiated. I'm really torn between the two-factor model (appealing because of its simplicty), and the four-factor model, which may reveal the most detail. But is that detail real, or is it the result of over-fitting? If I had to choose, I suppose I would choose the two-factor model. It does not fully fit the data, but it tells a simple story that makes sense.

If you disagree, it does not mean that you are wrong. In the end, the choice of a model is quite subjective, though the way these analyses are written up, the semi-arbitrary final choice will probably seem like the only possibility. This is especially true because only one set of factor loadings will be presented. If you were looking for the TRUTH here, I'm sorry to disappoint you. This is in the nature of the beast called exploratory factor analysis.

In spite of all the uncertainty, this enterprise has been blessed with apparent success. There are many hundreds of published factor analytic studies in the social sciences, especially in psychology. For example, in their book *The measurement of meaning* [50],

Osgood Suci and Tannenbaum (1957) describe a series of investigation into how people describe objects, using 7-point scales ranging from Ugly to Beautiful, Strong to Weak, Fast to Slow and so on. Exploratory factor analysis revealed the same three factors across many different domains. One of the factors had high factor loadings for Good-Bad, Beautiful-Ugly, and similar adjective pairs. The investigators named the factor *evaluative*. Similar considerations led them to identify the other two main factors as *potency* and *activity*. Osgood et al. proposed that these are the main dimensions of connotative (as opposed to denotative) meaning in the English language.

In another famous application [24], Hans Eysenck[16] (1947) factor analyzed questions from a large number of personality scales, arriving at two factors, *neuroticism* and *extraversion*. It's a bit interesting that in order to get a high score on neuroticism, you have to be willing to say bad things about yourself, while if you say mostly good things you will get a low neuroticism score. Perhaps it's just Osgood et al.'s evaluative factor, reversed. In any case, there are hordes of other examples, including Cattell's Sixteen Personality Factor Questionnaire [16] mentioned earlier. The earlier work, including the examples cited here, tended to use estimation methods that are less computationally demanding than maximum likelihood. Varimax rotation also caught on gradually, as computing equipment became more available. Rotation to a "simple structure" used to be graphical and more than a little subjective.

## 2.4 Oblique Rotations

**Correlated Factors** Naturally, not everybody is comfortable with uncorrelated factors. The question of whether factors are correlated seems like something that should be decided based on the data, and not simply assumed. The problem is that by the calculation (2.8), any correlation matrix of the factors is equally compatible with any data set. This means that estimating $\mathbf{\Phi} = cov(\mathbf{F})$ is futile. However, there is almost no limit to human ingenuity.

An early subjective method (as usual, see Harman [28]) for the history) is well adapted to a setting in which there are several clusters of variables, highly correlated within sets, and much less so between sets. Compare the formula for the sample correlation coefficient to the formula for the cosine of the angle between two vectors.

$$\cos\theta = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}|\,|\vec{y}|} \qquad\qquad r = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2}} \tag{2.18}$$

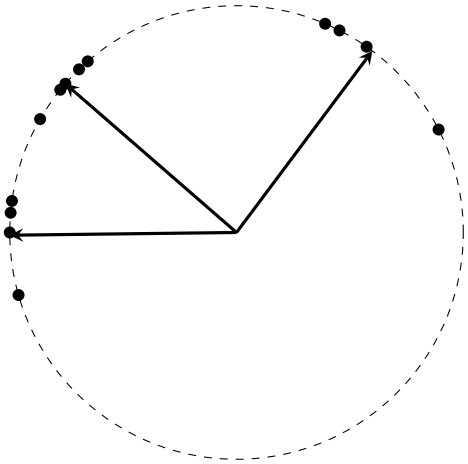
Now consider the vector of $n$ values for a variable as a point in $\mathbb{R}^n$. Suppose that the data are centered by subtracting off sample means, as they are in the standardized case we are considering. Then the correlation between two variables equals the cosine of the angle between the two data vectors. This means that considered as points in $\mathbb{R}^n$, a set of highly correlated variables are physically clustered together. To estimate the factor that

[16] Eminent research psychologist, racist scum, running dog of the tobacco companies, fabricator of data and student of Sir Cyril Burt, who was also racist scum and a fabricator of data. See the Wikipedia article.

gives rise to them, run a vector through the center of the cluster. The natural choice is to have the estimated factor pass through the centroid — that is, through the multivariate sample mean of the data vectors belonging to that particular cluster. Then the estimated factor is normalized, giving it variance one.

Figure 2.4 shows a hypothetical example in two dimensions. Since the variables are standardized, they all have length one. This means that in $\mathbb{R}^n$, the data points lie on the surface of a hyper-sphere of radius one, centered at the origin. Since Figure 2.4 is in two dimensions, all the points are on the unit circle.

Figure 2.4: Correlated factors estimated by centroids



The estimated correlations between factors are the cosines of the angles between the arrows, and the correlations of variables with factors are the cosines of the angles between data points and the arrows. It all makes sense, and looking at this example, it is hard to see why the parameters cannot be estimated successfully by this method. The trick is that by calculating the arrows based only on the points in a single cluster, we are implicitly assuming that the points in that cluster arise from only one common factor (plus random error). Under this assumption, lots of the $\lambda_{ij}$ values are zero, and in fact the remaining factor loadings and the correlations between factors are uniquely identifiable — provided there are at least three variables in each cluster. Chapter 3 treats confirmatory factor analysis models in which the parameters are identifiable, including the one just indicated.

The informal centroid method just described does work under some circumstances, but the big problem is cluster membership. When the variables form distinct, highly correlated clusters then everything is fine. More often, it will not be really clear how many clusters there are, and some variables will be difficult to classify. This uncertainty makes the method subjective, and led the developers of factor analysis to look for something more objective.

**Oblique Rotations** An *oblique* rotation is one in which the axes[17] need not remain at right angles. Starting with an initial orthogonal solution, the axes are rotated separately so as to achieve a simple structure in the factor loadings. There are various criteria for what "simple" means, leading to various flavours of the method.

The following account leads to the classical results, by a route that statisticians should be able to follow. The original explanations are much more complicated. Everything here is based on a model with equations $\mathbf{z} = \mathbf{\Lambda F} + \mathbf{e}$. The factors are standardized, and they are potentially correlated. Because the variance of each factor equals one, $cov(\mathbf{F}) = \mathbf{\Phi}$ is a correlation matrix. All other model specifications are the same as in Model (2.9) on page 219.

In an orthogonal factor model, the factor loadings in $\mathbf{\Lambda}$ are also the correlations between the observed variables and the factors. This is no longer true when the factors are correlated. With correlated factors, the calculations in (2.10) lead to

$$corr(\mathbf{z}, \mathbf{F}) = cov(\mathbf{z}, \mathbf{F}) = \mathbf{\Lambda\Phi}.$$

It is common to call the matrix of coefficients $\mathbf{\Lambda}$ the *factor pattern matrix*, while the matrix of correlations between variables and factors in $\mathbf{\Lambda\Phi}$ is called the *factor structure matrix*. In the factor analysis literature, these terms are applied to both the true parameter matrices and to their estimates.

When factors are correlated, some of the pleasing simplicity of the orthogonal model disappears. In particular, the explained variance of an observed variable no longer neatly splits itself into the variance explained by each factor. In scalar terms,

$$\begin{aligned} Var(z_i) &= var\left(\lambda_{i1}F_1 + \cdots + \lambda_{ip}F_p + e_i\right) \\ &= \sum_{j=1}^{p} \lambda_{ij}^2 Var(F_j) + \sum_{\ell \neq j} \lambda_{ij}\lambda_{i\ell} cov(F_\ell, F_j) + Var(e_i) \\ &= \sum_{j=1}^{p} \lambda_{ij}^2 + \sum_{\ell \neq j} \lambda_{ij}\lambda_{i\ell}\phi_{\ell j} + \omega_i. \end{aligned}$$

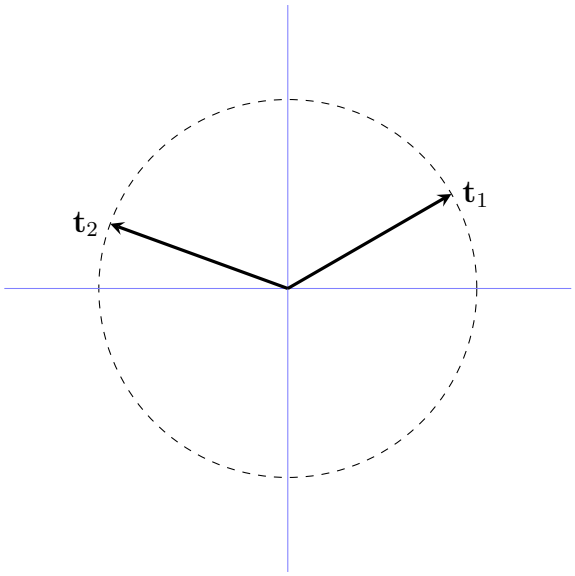
So, while the variance of $z_i$ is still decomposed into an explained part and an unexplained part, the explained variance includes terms that come from each pair of factors, with the contribution governed by the correlation between factors as well as the factor loadings. Notice that while the factor loadings and correlations between factors may be mutually adjusted as in the re-parameterizations (2.8), the amount of unexplained variance $\omega_i$ is not affected. The choice of an oblique rotation is one such re-parameterization, and we will presently see that oblique rotations do not affect estimates of the uniqueness (explained variance) for any variable.

Oblique rotations are carried out using a $p \times p$ transformation matrix $\mathbf{T} = [t_{ij}]$ satisfying $\mathbf{T}^\top\mathbf{T} = \mathbf{\Phi}$. Denote column $j$ of $\mathbf{T}$ by $\mathbf{t}_j$, so that $\mathbf{T} = (\mathbf{t}_1|\mathbf{t}_2|\cdots|\mathbf{t}_p)$. Because $\mathbf{\Phi}$ is a correlation matrix, $\mathbf{t}_j^\top\mathbf{t}_j = 1$. Thinking of $\mathbf{t}_1, \ldots, \mathbf{t}_p$ as vectors in $\mathbb{R}^p$ and using the formula in (2.18), the cosine of the angle between $\mathbf{t}_i$ and $\mathbf{t}_j$ is $\mathbf{t}_i^\top\mathbf{t}_j = Corr(F_i, F_j)$.

The matrix $\mathbf{T}$ is not unique. For $p = 2$, we have the picture in Figure 2.5. Spin the

[17] Think of the factors as dimensions, or axes of a co-ordinate system.

Figure 2.5: Columns of the $\mathbf{T}$ matrix




vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ around the unit circle[18] while keeping the angle between them constant. The cosine of the angle remains constant too, so there are infinitely many transformation matrices $\mathbf{T}$ that yield the same $\mathbf{\Phi}$. The square root matrix $\mathbf{\Phi}^{1/2}$ is just one of them. By the way, based on the similarity of Figure 2.5 to Figure 2.4, it would be easy to mistake the arrows in Figure 2.5 for factors. They are not. They are columns of the $\mathbf{T}$ matrix.

For a general number of factors $p$, the same spinning idea applies. Let $\mathbf{R}$ be a $p \times p$ orthogonal matrix. Then $(\mathbf{RT})^\top \mathbf{RT} = \mathbf{T}^\top \mathbf{R}^\top \mathbf{RT} = \mathbf{T}^\top \mathbf{T} = \mathbf{\Phi}$, and $\mathbf{RT}$ is another transformation matrix that produces $\mathbf{\Phi}$.

The next theorem says that, as Figure 2.5 suggests, *all* the transformation matrices for a given $\mathbf{\Phi}$ arise from spinning or reflecting a set of column vectors.

**Theorem 2.1** *Let* $\mathbf{T}_1$ *and* $\mathbf{T}_2$ *be square matrices satisfying* $\mathbf{T}_1^\top \mathbf{T}_1 = \mathbf{\Phi} = \mathbf{T}_2^\top \mathbf{T}_2$*, where* $\mathbf{\Phi}$ *is symmetric and positive definite. Then* $\mathbf{T}_2 = \mathbf{R}\,\mathbf{T}_1$*, where* $\mathbf{R}$ *is an orthogonal matrix.*

**Proof.** Because $\mathbf{\Phi}$ is positive definite, $\mathbf{T}_1$ and $\mathbf{T}_2$ are both full rank, and have inverses.

$$\begin{aligned} & \mathbf{T}_2^\top \mathbf{T}_2 = \mathbf{\Phi} \mathbf{T}_1^{-1} \mathbf{T}_1 \\ \Longrightarrow \quad & \mathbf{T}_2 = \left( (\mathbf{T}_2^\top)^{-1} \mathbf{\Phi} \mathbf{T}_1^{-1} \right) \mathbf{T}_1 = \mathbf{R}\,\mathbf{T}_1 \end{aligned}$$

[18] If the axes were being rotated, the rotation matrix $\mathbf{R}$ in (2.12) would be employed. Here, the axes are remaining in position, while the points are being rotated through an angle $\theta$. From the perspective of one of the points, it looks like the axes are being rotated through an angle of $-\theta$. So, to rotate the points, one would use the matrix $\mathbf{R}^\top$ in (2.13). Actually, in this case it does not matter which direction you spin the points.

Showing that $\mathbf{R}$ is an orthogonal matrix,

$$\begin{aligned}
\mathbf{R}^\top\mathbf{R} &= \left((\mathbf{T}_2^\top)^{-1}\mathbf{\Phi}\mathbf{T}_1^{-1}\right)^\top\left((\mathbf{T}_2^\top)^{-1}\mathbf{\Phi}\mathbf{T}_1^{-1}\right) \\
&= \mathbf{T}_1^{-1\top}\mathbf{\Phi}^\top\mathbf{T}_2^{\top\,-1\top}(\mathbf{T}_2^\top)^{-1}\mathbf{\Phi}\mathbf{T}_1^{-1} \\
&= \mathbf{T}_1^{\top\,-1}\mathbf{\Phi}\mathbf{T}_2^{-1}(\mathbf{T}_2^\top)^{-1}\mathbf{\Phi}\mathbf{T}_1^{-1} \\
&= \mathbf{T}_1^{\top\,-1}\mathbf{\Phi}\left(\mathbf{T}_2^\top\mathbf{T}_2\right)^{-1}\mathbf{\Phi}\mathbf{T}_1^{-1} \\
&= \mathbf{T}_1^{\top\,-1}\mathbf{\Phi}\mathbf{\Phi}^{-1}\mathbf{\Phi}\mathbf{T}_1^{-1} \\
&= \mathbf{T}_1^{\top\,-1}\mathbf{\Phi}\mathbf{T}_1^{-1} \\
&= \mathbf{T}_1^{\top\,-1}\left(\mathbf{T}_1^\top\mathbf{T}_1\right)\mathbf{T}_1^{-1} \\
&= \mathbf{I}\cdot\mathbf{I} = \mathbf{I} \quad \blacksquare
\end{aligned}$$

You might be thinking that representing a set of unknown parameters in a way that is not unique will just make estimation more difficult. In fact, estimation of $\mathbf{\Phi}$ cannot be successful by conventional standards anyway, because $\mathbf{\Phi}$ is not identifiable. As you will see, the matrix $\mathbf{T}$ will be chosen to yield a nice simple factor structure. The fact that $\mathbf{T}$ is not unique just provides a wider range of options.

In the meantime, consider a standard orthonormal basis for $\mathbb{R}^p$, with basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_p$, where $\mathbf{b}_i$ has a one in position $i$, and zeros elsewhere. Noting that

$$\mathbf{t}_j = \begin{pmatrix} t_{1j} \\ t_{2j} \\ \vdots \\ t_{pj} \end{pmatrix},$$

the cosine of the angle between $\mathbf{b}_i$ and $\mathbf{t}_j$ is $\mathbf{b}_i^\top\mathbf{t}_j = t_{ij}$. Now suppose we were to adopt $\mathbf{t}_1, \ldots, \mathbf{t}_p$ as an alternative basis for $\mathbb{R}^p$. Column $j$ of the transformation matrix $\mathbf{T}$ contains the cosines of the angles between $\mathbf{t}_j$ and the original basis vectors.

Geometrically, changing to the basis $\mathbf{t}_1, \ldots, \mathbf{t}_p$ corresponds to rotating each of the original basis vectors through a set of angles satisfying the cosines in $\mathbf{T}$. It is an *oblique* rotation rather than an orthogonal rotation, because the new basis vectors need not be at right angles. The operation can be represented as a matrix multiplication:

$$\mathbf{T}^\top\mathbf{b}_j = \mathbf{t}_j.$$

This rotation can be applied to $\mathbf{a} = [a_j]$, a general point in $\mathbb{R}^p$. We have

$$\mathbf{a} = a_1\mathbf{b}_1 + \cdots + a_p\mathbf{b}_p,$$

so that

$$\begin{aligned}
\mathbf{T}^\top\mathbf{a} &= \mathbf{T}^\top(a_1\mathbf{b}_1 + \cdots + a_p\mathbf{b}_p) \\
&= a_1\mathbf{T}^\top\mathbf{b}_1 + \cdots + a_p\mathbf{T}^\top\mathbf{b}_p \\
&= a_1\mathbf{t}_1 + \cdots + a_p\mathbf{t}_p,
\end{aligned}$$

representing the point $\mathbf{a}$ in terms of the new co-ordinate system. The main point here is that it makes sense to describe pre-multiplication by $\mathbf{T}^\top$ as a rotation, one that is not necessarily orthogonal.

Here is how oblique rotation may be used[19] to estimate the unknown parameters $\mathbf{\Lambda}$ and $\mathbf{\Phi}$. Returning to the model equations, we start by applying a change of variables to the factors.

$$\begin{aligned}\mathbf{z} &= \mathbf{\Lambda F} + \mathbf{e} \\ &= \mathbf{\Lambda T}^\top(\mathbf{T}^\top)^{-1}\mathbf{F} + \mathbf{e} \\ &= \mathbf{AF}' + \mathbf{e},\end{aligned}$$

where $\mathbf{A} = \mathbf{\Lambda T}^\top$ and $\mathbf{F}' = (\mathbf{T}^\top)^{-1}\mathbf{F}$. We have

$$\begin{aligned}cov(\mathbf{F}') &= cov\left((\mathbf{T}^\top)^{-1}\mathbf{F}\right) \\ &= (\mathbf{T}^\top)^{-1} cov(\mathbf{F}) \left((\mathbf{T}^\top)^{-1}\right)^\top \\ &= (\mathbf{T}^\top)^{-1}\mathbf{\Phi}\mathbf{T}^{-1} \\ &= (\mathbf{T}^\top)^{-1}\mathbf{T}^\top\mathbf{T}\mathbf{T}^{-1} \\ &= \mathbf{I},\end{aligned}$$

so the change of variables and the accompanying re-parameterization results in an orthogonal factor model. The new parameter matrix $\mathbf{A} = [a_{ij}]$ is not identifiable, but it can be estimated up to an orthogonal rotation, perhaps by constrained maximum likelihood. This yields $\widehat{\mathbf{A}}$. (In Section 2.3, the symbol $\widetilde{\mathbf{A}}$ was employed for the constrained MLE. Here, we return to a more standard notation.)

Now perform another change of variables, to return to a version of the original model with correlated factors.

$$\begin{aligned}\mathbf{z} &= \mathbf{AF}' + \mathbf{e} \\ &= \mathbf{A}\,(\mathbf{T}^\top)^{-1}\mathbf{T}^\top\,\mathbf{F}' + \mathbf{e} \\ &= \mathbf{A}(\mathbf{T}^\top)^{-1}\,\mathbf{T}^\top(\mathbf{T}^\top)^{-1}\mathbf{F} + \mathbf{e} \\ &= \mathbf{A}(\mathbf{T}^\top)^{-1}\,\mathbf{F} + \mathbf{e}\end{aligned}$$

Instead of expanding $\mathbf{A}$ and simplifying back to the original model, we will use our earlier estimate of $\mathbf{A}$, which is an estimate of $\mathbf{\Lambda T}^\top$. Symbolically, $\widehat{\mathbf{A}} = \widehat{\mathbf{\Lambda T}^\top}$. The matrix of original factor loadings $\mathbf{\Lambda}$ (the factor pattern matrix) is estimated by

$$\widehat{\mathbf{\Lambda}} = \widehat{\mathbf{\Lambda T}^\top}(\mathbf{T}^\top)^{-1} = \widehat{\mathbf{A}}(\mathbf{T}^\top)^{-1}. \tag{2.19}$$

The factor structure matrix $corr(\mathbf{z}, \mathbf{F}) = \mathbf{\Lambda\Phi}$ is estimated by

$$\begin{aligned}\widehat{\mathbf{\Lambda\Phi}} &= \widehat{\mathbf{A}}(\mathbf{T}^\top)^{-1}\mathbf{\Phi} \\ &= \widehat{\mathbf{A}}(\mathbf{T}^\top)^{-1}\mathbf{T}^\top\mathbf{T} \\ &= \widehat{\mathbf{A}}\mathbf{T}\end{aligned} \tag{2.20}$$

[19] I say "may be" used, because this is not the typical way of describing the process. However, it is clear to me and it leads to the usual estimates.

The problem is that the estimates (2.19) and (2.20) both depend on the transformation matrix $\mathbf{T}$, which unknown and un-knowable[20]. The solution, as in the case of orthogonal rotation, is to choose a $\mathbf{T}$ matrix that results in a nice simple structure, – either in the factor pattern (2.19) or the factor structure (2.20). As a by-product, the choice of $\mathbf{T}$ yields an estimate of $\mathbf{\Phi}$. Using $\widehat{\mathbf{T}}$ to denote the chosen $\mathbf{T}$ matrix, $\widehat{\mathbf{\Phi}} = \widehat{\mathbf{T}}^\top \widehat{\mathbf{T}}$.

The way in which $\mathbf{T}$ is chosen does not affect the estimated uniqueness, the portion of the variance in an observed variable that comes from the factors. From $cov(\mathbf{z}) = \mathbf{\Lambda\Phi\Lambda}^\top + \mathbf{\Omega}$, the estimated explained variances of the observed variables are the diagonal elements of

$$\begin{aligned}
\widehat{\mathbf{\Lambda}}\widehat{\mathbf{\Phi}}\widehat{\mathbf{\Lambda}}^\top &= \widehat{\mathbf{A}}\left(\widehat{\mathbf{T}}^\top\right)^{-1}\widehat{\mathbf{T}}^\top\widehat{\mathbf{T}}\left(\widehat{\mathbf{A}}\left(\widehat{\mathbf{T}}^\top\right)^{-1}\right)^\top \\
&= \widehat{\mathbf{A}}\widehat{\mathbf{T}}\left(\left(\widehat{\mathbf{T}}^\top\right)^{-1}\right)^\top\widehat{\mathbf{A}}^\top \\
&= \widehat{\mathbf{A}}\widehat{\mathbf{T}}\widehat{\mathbf{T}}^{-1}\widehat{\mathbf{A}}^\top \\
&= \widehat{\mathbf{A}}\widehat{\mathbf{A}}^\top,
\end{aligned}$$

which does not depend on the oblique rotation $\widehat{\mathbf{T}}$.

The choice of $\mathbf{T}$ depends on what criterion is optimized in search of "simple structure." A variety of criteria have been proposed, each with its own impressive name and cadre of enthusiastic supporters — most of whom, sad to say, are no longer with us. Harman [28] describes the oblimax, oblimin (including quartimin, covarimin, and biquartimin), direct oblimin, binormamin and orthoblique methods, and I may have missed some. Confronted with this wealth of alternatives, I have decided to present the oblimin family, mostly because of its connection to varimax.

**Oblimin rotation** Initially, oblimin rotation sought to simplify the factor structure matrix, while later work focused on simplifying the factor pattern. Logically but not chronologically, the story begins with the *covarimin* method. Consider any two columns of the estimated factor structure matrix in Expression 2.20, but square all the elements in the matrix. Suppose that all the squared correlations in the matrix are either close to one or close to zero, and that large squared correlations in one column are beside near-zero squared correlations in the other column. If this could be achieved for every pair of columns, it would be a nice simple structure in which each observed variable has a large correlation (positive or negative) with just one factor, and near zero correlations with the others. In other words, we want negative relationships between the squared correlations in all the columns.

Accordingly, square all the estimated correlations in Expression 2.20, and think of the resulting $k \times p$ matrix as a kind of data file, with $k$ observations on $p$ "variables."

[20]The matrix $\mathbf{T}$ is constrained by the fact the its columns are vectors of length one, and also by $\mathbf{T}^\top\mathbf{T} = \mathbf{\Phi}$. This does not help us get at $\mathbf{T}$, because the correlation matrix $\mathbf{\Phi}$ is not just unknown, it is not even identifiable. In addition, it has previously been shown that uncountably many $\mathbf{T}$ matrices produce a given $\mathbf{\Phi}$. Therefore, even if $\mathbf{\Phi}$ were known exactly, recovery of the "true" $\mathbf{T}$ would be impossible.

Calculate the $p \times p$ sample covariance matrix for these "data." The covarmin criterion is the sum of unique off-diagonal elements (multiplied by $k^2$):

$$\sum_{i=1}^{p} \sum_{j=i+1}^{p} \left( k \sum_{\ell=1}^{k} c_{\ell i}^2 c_{\ell j}^2 - \sum_{\ell=1}^{k} c_{\ell i}^2 \sum_{\ell=1}^{k} c_{\ell j}^2 \right), \tag{2.21}$$

where $c$ is an estimated correlation. Minimize (2.21) over the elements of the matrix $\mathbf{T}$. This can be done one column (axis) at a time, literally rotating the axes. As an option, it is possible to adjust for communalities as in (2.17). Again, one divides squared correlations by the communality, that is, by the total amount of variance in the variable that is explained by the common factors.

Covarimin is similar in approach to varimax, and in fact they are both described in the same 1958 paper by H. F. Kaiser [38]. Both methods treat a matrix of squared estimated correlations as data. Varimax maximizes the sum of sample variances of the columns, and covarimin minimizes the sum of sample covariances of the columns.

Covarimin was a nice idea, but based on application to real data sets, it did not yield satisfactory results. The problem was that it tended to produce solutions that were "too orthogonal." That is, the estimated correlation matrix of the factors $\widehat{\mathbf{\Phi}} = \mathbf{T}^\top \mathbf{T}$ tended to be quite close to the identity, regardless of the data. Perhaps as a way of reducing how negative the covariances were, a modification was to drop the negative part of (2.21). This yielded a criterion called *quartimin*, which had been proposed some years earlier:

$$\sum_{i=1}^{p} \sum_{j=i+1}^{p} \left( \sum_{\ell=1}^{k} c_{\ell i}^2 c_{\ell j}^2 \right). \tag{2.22}$$

The quartimin criterion tended to yield solutions that were "too oblique." As a compromise, putting back the $k$ that was omitted from (2.22) and then averaging the two criteria yielded the *biquartimin* criterion:

$$\sum_{i=1}^{p} \sum_{j=i+1}^{p} \left( k \sum_{\ell=1}^{k} c_{\ell i}^2 c_{\ell j}^2 - \frac{1}{2} \sum_{\ell=1}^{k} c_{\ell i}^2 \sum_{\ell=1}^{k} c_{\ell j}^2 \right), \tag{2.23}$$

effectively retaining half of the second term in the covarimin criterion (2.21). Some viewed the biquartimin compromise as "just right," but it is a matter of taste how much of the second term to retain. To accommodate all preferences, the general oblimin criterion replaces the fraction $\frac{1}{2}$ with a number between zero and one inclusive, symbolized by $\gamma$.

$$\sum_{i=1}^{p} \sum_{j=i+1}^{p} \left( k \sum_{\ell=1}^{k} c_{\ell i}^2 c_{\ell j}^2 - \gamma \sum_{\ell=1}^{k} c_{\ell i}^2 \sum_{\ell=1}^{k} c_{\ell j}^2 \right), \tag{2.24}$$

where $0 \leq \gamma \leq 1$. Setting $\gamma = 0$ yields quartimin, while $\gamma = \frac{1}{2}$ yields biquartimin, and $\gamma = 1$ yields covarimin.

**Direct oblimin** The oblimin method just described seeks to simplify the factor structure (the matrix of estimated correlations between variables and rotated factors). In contrast, *direct* oblimin seeks to simplify the factor pattern, the matrix of estimated factor loadings[21]. Both versions of oblimin find a transformation matrix $\mathbf{T}$ that minimizes a criterion of the form (2.24), subject to the restriction that the column vectors of $\mathbf{T}$ have length one. In the original oblimin, the $c_{\ell j}$ are elements of $\widehat{\mathbf{A}}\mathbf{T}$ (see Expression 2.20), while for direct oblimin, the $c_{\ell j}$ are elements of $\widehat{\mathbf{A}}(\mathbf{T}^\top)^{-1}$ (Expression 2.19). It can make a difference, because there is no reason to expect the $\mathbf{T}$ that optimizes $\widehat{\mathbf{A}}(\mathbf{T}^\top)^{-1}$ will also optimize $\widehat{\mathbf{A}}\mathbf{T}$, unless $\mathbf{T}$ is close to the identity.

The name "direct" oblimin seems to be something of a historical accident. The original oblimin algorithm really was very complicated and indirect. In the paper that introduced direct oblimin [33], Jennrich and Sampson (1966) provided a much more straightforward algorithm for minimizing the factor pattern version of (2.24). With more than a half century of hindsight, it seems that there was a failure to distinguish between directness in the criterion to be minimized and directness in the algorithm used to get the job done. At any rate, everyone seems to have bought it, and the original "indirect" version of oblimin has faded away.

The direct oblimin of Jennrich and Sampson (1966) came to full fruition almost 40 years later [7] in Bernaards and Jennrich (2005). Yes, it's the same Jennrich. Bernaards and Jennrich do the optimization directly over the columns of the $\mathbf{T}$ matrix, alternating between a gradient descent step and a projection onto the set of column vectors with length one. The mathematical expressions are remarkably simple and elegant when written in matrix form.

Bernaards and Jennrich have provided the R package `GPArotation`, which implements their method for a variety of orthogonal and oblique rotations. The options naturally include direct oblimin, but they do not include indirect oblimin, as far as I can tell. R's built-in `factanal` function has a `rotation=` option, and it can use all the methods in `GPArotation`, provided that the `GPArotation` package is loaded. Otherwise, `factanal` only knows about `varimax` and `promax`. The widely used `psych` package does factor analysis with oblique rotation using functions from `GPArotation`, so oblimin rotation in `psych` is direct oblimin. This has a lot of prominence because in `psych`'s workhorse `fa` function (`fa` for factor analysis), the default is to apply an oblimin rotation unless the user specifies otherwise. The `EFAtools` package [61] uses `GPArotation` and `psych`. I have been unable to find any R packages that do the original "indirect" oblimin.

In terms of commercial software, online documentation suggests that in SAS and SPSS, oblimin means direct oblimin. The once-great BMDP package had both direct and indirect oblimin options, but it is no longer available. In practical terms, direct oblimin rotation is your only choice unless you write your own function.

There is no obvious reason why the Bernaards and Jennrich algorithm could not be applied to the factor structure matrix instead of the factor pattern matrix. The result would be a very direct version of indirect oblimin. Should you bother to write the

[21] Again, the factor loadings are constants that are like regression coefficients, linking the rotated factors to the observed variables.

code? In my judgement, the answer is no, because simplicity in the factor loadings is probably more desirable than simplicity in the correlations between variables and factors anyway. Thinking of factor analysis as a causal model (that's the structural equation model perspective), the factors are literally producing the observed variables through the factor loadings in the factor pattern matrix. On the other hand, the factor structure matrix $\widehat{\mathbf{A}}\mathbf{T}$ is estimating $corr(\mathbf{z}, \mathbf{F})$. From the formula $corr(\mathbf{z}, \mathbf{F}) = \mathbf{\Lambda\Phi}$, the correlation between an observed variable and a factor depends on the correlations between factors as well as the direct connection of the factor to the variable.

Consider the two-factor example of Figure 2.1 and Equations (2.7), except with the observed variables $d_{i,j}$ standardized. We have, for example,

$$\begin{aligned} corr(z_{i,4}, F_{i,1}) &= cov(z_{i,4}, F_{i,1}) \\ &= cov(\lambda_{41}F_{i,1} + \lambda_{42}F_{i,2} + e_{i,4},\ F_{i,1}) \\ &= \lambda_{41}\, cov(F_{i,1}, F_{i,1}) + \lambda_{42}\, cov(F_{i,1}, F_{i,2}) + cov(e_{i,4}, F_{i,1}) \\ &= \lambda_{41}\, Var(F_{i,1}) + \lambda_{42}\, corr(F_{i,1}, F_{i,2}) + 0 \\ &= \lambda_{41} + \lambda_{42}\phi_{1,2}. \end{aligned}$$

If there were $p$ factors, the formula would be $corr(z_{i,4}, F_{i,1}) = \lambda_{41} + \sum_{j=2}^{p} \lambda_{4j}\phi_{1,j}$.

We see that the correlation between an observed variable and a factor includes the direct link between the variable and the factor, but mixed together with the links between the variable and all the other factors, in a way that depends on the correlations between factors. This means that the interpretation of such a correlation may not be straightforward at all. For example, a high correlation could come from a strong direct link between the variable and the factor, but it could also come from a weak or zero direct link, accompanied by strong indirect effects of the other factors. Conversely, evidence of a strong direct link could be suppressed by the operation of the other factors, resulting in a near zero correlation. It's very much like the correlation-causation picture in general. Though they did not suggest this argument, Jennrich and Sampson showed good taste when they decided to focus on the factor pattern matrix.

It is important to mention that the effect of the $\gamma$ parameter is vastly different for the two oblimin methods. Recall that $0 \leq \gamma \leq 1$ for indirect oblimin, with $\gamma = 0$ producing the most oblique solutions (largest estimated correlations between factors), and $\gamma = 1$ producing the most orthogonal solutions. For direct oblimin, the connection is reversed, with obliqueness *increasing* as a function of $\gamma$, rather than decreasing. For direct oblimin, very large negative $\gamma$ values yield near zero correlations between factors, while the estimated correlations between factors rapidly approach $\pm 1$ for fairly small positive values of $\gamma$. Then, still for very modest positive $\gamma$ values, the matrix $\mathbf{T}$ becomes numerically singular, and the algorithm fails to converge. The usual recommendation is that $\gamma$ should be zero or negative for direct oblimin.

To avoid confusion, Harman [28] uses the symbol $\delta$ instead of $\gamma$ for direct oblimin, reserving the symbol $\gamma$ for indirect oblimin. While SPSS follows Harman's notation, R does not. In the `oblimin` function of the `GPArotation` package, the `gam=` argument controls the value of $\gamma$ for direct oblimin. Similarly, `rotate=quartimin` means direct quartimin; that is, direct oblimin with $\gamma = 0$.

If you happen to know about indirect oblimin, the vocabulary in the `GPArotation` documentation can be a trap for the unwary. In `help(oblimin)`, the `gam=` argument is documented by "0=Quartimin, .5=Biquartimin, 1=Covarimin." These are all the direct oblimin versions. It's a bit strange because, while $\gamma = 0$ is reasonable and in fact is the default, $\gamma = 1/2$ does not correspond to anything interesting for direct oblimin, and the value $\gamma = 1$ frequently leads to convergence problems.

Here is an illustration of factor analysis with oblique rotation for the Mind-body data. To make the example complete, we begin by reading the data. Then, loading the `GPArotation` package makes `rotate=oblimin` available in `factanal`.

```
> rm(list=ls())
> bodymind = read.table('http://www.utstat.toronto.edu/~brunner/openSEM/data/bodymind.data.txt')
> dat = as.matrix(bodymind[,2:10]) # Omit sex. dat is now a numeric matrix.
> # install.packages("GPArotation", dependencies=TRUE) # Only need to do this once
> library(GPArotation)
> ob2 = factanal(dat, factors=2, rotation='oblimin'); print(ob2, cutoff=0)

Call:
factanal(x = dat, factors = 2, rotation = "oblimin")

Uniquenesses:
progmat  reason  verbal headlng headbrd headcir   bizyg  weight  height
  0.616   0.274   0.264   0.324   0.618   0.016   0.473   0.577   0.633

Loadings:
        Factor1 Factor2
progmat  0.081   0.584
reason  -0.027   0.862
verbal   0.012   0.853
headlng  0.829  -0.019
headbrd  0.655  -0.125
headcir  0.982   0.025
bizyg    0.688   0.088
weight   0.656  -0.014
height   0.600   0.014

               Factor1 Factor2
SS loadings      3.353   1.836
Proportion Var   0.373   0.204
Cumulative Var   0.373   0.577

Factor Correlations:
        Factor1 Factor2
Factor1   1.000   0.384
Factor2   0.384   1.000

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 87.55 on 19 degrees of freedom.
The p-value is 8.97e-11
```

Note the matrix $\widehat{\mathbf{\Phi}}$ under `Factor Correlations`, with an estimated correlation between factors of 0.348. For comparison, here is a repeat of the analysis with a varimax rotation.

```
> print( factanal(dat, factors=2, rotation='varimax'), cutoff=0) # For comparison

Call:
factanal(x = dat, factors = 2, rotation = "varimax")

Uniquenesses:
progmat  reason  verbal headlng headbrd headcir   bizyg  weight  height
  0.616   0.274   0.264   0.324   0.618   0.016   0.473   0.577   0.633

Loadings:
        Factor1 Factor2
progmat 0.181   0.592
reason  0.124   0.843
verbal  0.160   0.843
headlng 0.806   0.161
headbrd 0.618   0.019
headcir 0.963   0.238
bizyg   0.687   0.236
weight  0.638   0.129
height  0.588   0.144

               Factor1 Factor2
SS loadings      3.257   1.948
Proportion Var   0.362   0.216
Cumulative Var   0.362   0.578

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 87.55 on 19 degrees of freedom.
The p-value is 8.97e-11
```

The estimated uniquenesses are the same, as they should be. The factor loadings are quite similar; they are perhaps a bit sharper for the oblique rotation, so the oblique rotation allowed a closer approach to simple structure. The little sub-table under the factor loadings, starting with `SS loadings`, is also similar for the varimax and oblimin rotations. However, this is deceiving. That subtable, generated as part of the print method for an object of class loadings, is appropriate only when factors are orthogonal. In that case, squared factor loadings are separate components of variance, and `SS loadings` makes sense. With an oblique rotation there is no such interpretation, and the next example will make that table look as nonsensical as it really is.

First, just note that the test for number of factors (the chi-squared test for goodness of fit) is identical for the varimax and oblimin rotations. That is because displays for all rotations, orthogonal or oblique, simply report the test for the initial solution, which is orthogonal.

Now let us return to the `SS loadings` table under the factor loadings for oblimin rotation. With higher values of $\gamma$ in (2.24), estimated correlations between factors become larger, and the factor pattern matrix becomes more dissimilar to the factor structure matrix. Now, R's built-in `factanal` function will not accept a $\gamma$ argument (at least not in a natural way), but the `fa` function in the `psych` package will.

```
> # Try fa with gam: Check SS loadings
> # install.packages("psych", dependencies=TRUE) # Only need to do this once
> library(psych); library(psychTools)
> # fa(dat,nfactors=2, fm='ml', rotate = 'oblimin', gam=0) # Same results as ob2
> psych0 = fa(dat,nfactors=2, fm='ml', rotate = 'oblimin', gam=1)
> psych0$loadings

Loadings:
        ML1    ML2
progmat -0.503 -1.064
reason  -1.030 -1.733
verbal  -0.943 -1.672
headlng  1.640  0.952
headbrd  1.420  0.969
headcir  1.888  1.033
bizyg    1.243  0.585
weight   1.295  0.750
height   1.155  0.634

                  ML1    ML2
SS loadings    15.031 11.149
Proportion Var  1.670  1.239
Cumulative Var  1.670  2.909
```

I rest my case. The matrix of factor loadings is definitely a factor pattern and not a factor structure matrix, because its elements are not correlations. The `SS loadings` table still squares them, adds them up and divides by nine (the number of variables) in order to get proportions of explained variance. This is nonsense, because the resulting proportions are greater than one.

One does not need to use the `psych` package to be able to specify the $\gamma$ parameter. It's better to use the `oblimin` function in the `GPArotation` package[22]. To do this, first fit an initial, orthogonal model. Then use the `oblimin` function on the unrotated factor loadings $\widehat{\mathbf{A}}$.

```
> fit2a = factanal(dat,factors=2,rotation='none')
> Ahat = fit2a$loadings
> O2b = oblimin(Ahat); O2b # Matches ob2 (gamma=0)
Oblique rotation method Oblimin Quartimin converged.
Loadings:
        Factor1 Factor2
progmat  0.0810  0.5837
reason  -0.0271  0.8619
verbal   0.0121  0.8533
headlng  0.8293 -0.0194
headbrd  0.6554 -0.1249
headcir  0.9822  0.0251
bizyg    0.6880  0.0876
weight   0.6558 -0.0140
```

[22] Of course the people who wrote the `psych` package might not agree. `psych` can do a lot of things, and if you need or want to do them, you should use the `psych` package.

```
height   0.6001  0.0141

Rotating matrix:
       [,1]   [,2]
[1,]  0.975 0.0608
[2,] -0.471 1.0813

Phi:
      [,1]  [,2]
[1,] 1.000 0.384
[2,] 0.384 1.000
```

The factor loadings match `ob2`, with a default value of $\gamma = 0$. Note that the rotation is described as `Oblimin Quartimin`, which is accurate as long as it's understood to be direct oblimin. The so-called `Rotating matrix` is $\left(\widehat{\mathbf{T}}^{\top}\right)^{-1}$. Looking at the list of items produced by the `oblimin` function,

```
> ls(O2b)
[1] "convergence" "Gq"          "loadings"    "method"      "orthogonal"  "Phi"         "Table"
[8] "Th"
```

The `Th` item is $\widehat{\mathbf{T}}$, so the following matches the "`Rotating matrix`."

```
> solve(t(O2b$Th))
           [,1]       [,2]
[1,]  0.9750642 0.06083524
[2,] -0.4713192 1.08129141
```

The `loadings` item is the rotated factor pattern matrix. Fortunately, it is not an object of class "loadings," so it does not use the misleading print method.

```
> O2b$loadings
            Factor1     Factor2
progmat  0.08096306  0.58366083
reason  -0.02710434  0.86193259
verbal   0.01213684  0.85326892
headlng  0.82927554 -0.01942290
headbrd  0.65541656 -0.12490579
headcir  0.98223567  0.02510846
bizyg    0.68800789  0.08760370
weight   0.65576088 -0.01399468
height   0.60011168  0.01406469
```

It is instructive to look at the results for $\gamma = 1/2$, described as (direct) `Oblimin Biquartimin`.

```
> O2c = oblimin(Ahat, gam = 0.5); O2c
Oblique rotation method Oblimin Biquartimin converged.
Loadings:
        Factor1 Factor2
progmat -0.0399  0.6440
reason  -0.2201  0.9756
verbal  -0.1751  0.9593
```

```
headlng  0.9153 -0.1604
headbrd  0.7476 -0.2502
headcir  1.0734 -0.1358
bizyg    0.7364 -0.0162
weight   0.7234 -0.1253
height   0.6561 -0.0844

Rotating matrix:
       [,1]    [,2]
[1,]  1.058 -0.0944
[2,] -0.756  1.2969

Phi:
      [,1]  [,2]
[1,] 1.000 0.639
[2,] 0.639 1.000
```

The estimated correlation between factors is larger, and so are the estimated factor loadings. It still tells the same general story, with the first factor representing physical size, the the second factor reflecting performance on the mental tests.

To give an idea of how the correlation between factors varies as a function of $\gamma$, I fit a series of models with different $\gamma$ values, covering a wide range.

```
> # Correlation between factors as a function of gamma
> options(scipen=999) # To suppress scientific notation
> gammaval = c(-500, -100, -50, -10, 0, 0.25, 0.50, 0.75, 1)
> ngamma = length(gammaval); phi12 = numeric(ngamma)
> for(j in 1:ngamma) phi12[j] = oblimin(Ahat, gam = gammaval[j])$Phi[1,2]
> round(rbind(gammaval,phi12),3)
            [,1]     [,2]    [,3]    [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
gammaval -500.000 -100.000 -50.000 -10.000 0.000 0.250 0.500 0.750 1.000
phi12       0.002    0.011   0.022   0.105 0.384 0.481 0.639 0.802 0.935
```

Observe how the correlation between factors approaches zero very slowly as $\gamma \to -\infty$, and approaches one rapidly for positive values of $\gamma$ ; it could also approach -1 for increasing gamma, depending on the data and the starting values for the oblimin minimization.

One might well ask, what's the right $\gamma$ value? The answer is that there is no right answer. $\gamma$ is not an unknown parameter of the statistical model, and it is not something that can be estimated. It's a setting that determines the criterion to be minimized in order to seek a simple structure in the estimated factor loadings. Typically, users try different $\gamma$ values, and settle on one that produces results that seem reasonable for the data. Or, they just use the software default of $\gamma = 0$.

As a final example, consider a three-factor model for the Mind-body data. Before doing this, I will disclose that I expect one mental factor and two physical factors, and that the physical factors will be more correlated with one another than either of them is with the mental factor.

```
> fit3a = factanal(dat,factors=3,rotation='none')
> A3hat = fit3a$loadings
```

```
> oblimin(A3hat)
Oblique rotation method Oblimin Quartimin converged.
Loadings:
         Factor1 Factor2 Factor3
progmat  0.18819 -0.0948  0.5686
reason  -0.07027 -0.0170  0.9104
verbal  -0.00706  0.0312  0.8253
headlng  1.02610 -0.0549 -0.0135
headbrd -0.10556  0.9136 -0.0746
headcir  0.59540  0.4652  0.1061
bizyg    0.11682  0.7509  0.1408
weight   0.31027  0.4429  0.0422
height   0.38873  0.3609  0.0477

Rotating matrix:
        [,1]    [,2]    [,3]
[1,]  1.0070 -0.0171 0.00256
[2,] -0.5830  0.8516 0.60471
[3,]  0.0544 -0.7553 0.87791

Phi:
      [,1]  [,2]  [,3]
[1,] 1.000 0.465 0.327
[2,] 0.465 1.000 0.254
[3,] 0.327 0.254 1.000
```

The third factor is definitely mental, and could be called "academic ability" without raising much controversy. The first factor is dominated by head length and to a lesser extent by head circumference; it could be called "head size." The second factor has its highest loadings on head breadth and bizygomatic breadth. It could be called "face width." The picture is quite similar to what appeared with an orthogonal (varimax) rotation. The correlation between the two physical factors is higher than the others in the $\widehat{\mathbf{\Phi}}$ matrix, but not notably so.

## 2.5 Factor Scores

My man Harman [28] suggests that there are two potential reasons for doing factor analysis. One is to understand how certain unobservable factors give rise to a set of observable data. The other reason is data reduction. You have a lot of variables, and you'd like to work with a smaller set that contains essentially the same information. So you do a factor analysis, and then somehow "estimate" the values of the factors for all the members of your sample. The estimates are called *factor scores*. They may be more interpretable than the original data, in the sense that they might represent the underlying quantities that the data were intended to measure. Certainly, there will be fewer of them. If only for this reason, they may be easier to think about and to incorporate into subsequent data analyses.

**Principal components** Frequently,

## 2.6 A Dose of Reality

Let us take a step back from all these interesting details, and consider what we have. In Sections 2.2 and 2.3, it was shown that the parameters of the exploratory factor analysis model are not identifiable, even if they are constrained by making the factors uncorrelated. Infinitely many sets of parameter values are consistent with any data set, so that using the data alone to distinguish between them is hopeless. The solution in exploratory factor analysis is rotation. After locating a family of parameter sets that are all equally reasonable given the data (and arguably better than other values outside the family), one rotates the factors in such a way that the factor loadings achieve a simple structure, one that is scientifically meaningful.

The problem is that in statistics, there *is* such a thing as a true parameter value[23]. If the truth resembles simple structure, rotation will take you closer to the truth. If the truth does not resemble simple structure, rotation will take you farther away. The factor analysts have a deep philosophical answer to this, but before dealing with that I will give a few examples using simulated data. The advantage of simulated data is that we know exactly what the true parameter values are.

In the first example, the truth corresponds to simple structure. There are two uncorrelated factors and eight observed variables. The first four variables load only on factor one, and the last four load only on factor two. This is an extreme case of simple structure, and looks very much like varimax. All the distributions are normal, so that the model underlying maximum likelihood estimation is exactly correct. All the variables are centered, and the true variances of both factors and observed variables are exactly equal to one. In the code, the factor loadings (the only parameters) are denoted by $\mathtt{L}_{ij}$. The sample size is huge, so that sampling error does not make the pattern of results harder to see.

```
> rm(list=ls())
> n = 50000 # Huge sample size
> # True factor loadings have a simple structure like varimax (All communalities = 0.49)
> # Factor loadings
> L11 = 0.7; L12 = 0.0
> L21 = 0.7; L22 = 0.0
> L31 = 0.7; L32 = 0.0
> L41 = 0.7; L42 = 0.0
> L51 = 0.0; L52 = 0.7
> L61 = 0.0; L62 = 0.7
> L71 = 0.0; L72 = 0.7
> L81 = 0.0; L82 = 0.7
> # Error Variances
> v1 = 1 - L11**2 - L12**2
> v2 = 1 - L21**2 - L22**2
> v3 = 1 - L31**2 - L32**2
> v4 = 1 - L41**2 - L42**2
> v5 = 1 - L51**2 - L52**2
> v6 = 1 - L61**2 - L62**2
```

[23] Except maybe in the mind of the most radical subjective Bayesian.

```
> v7 = 1 - L71**2 - L72**2
> v8 = 1 - L81**2 - L82**2
> # Generate data
> set.seed(9999)
> F1 = rnorm(n,0,1); F2 = rnorm(n,0,1)
> d1 = L11*F1 + L12*F2 + rnorm(n,0,sqrt(v1))
> d2 = L21*F1 + L22*F2 + rnorm(n,0,sqrt(v2))
> d3 = L31*F1 + L32*F2 + rnorm(n,0,sqrt(v3))
> d4 = L41*F1 + L42*F2 + rnorm(n,0,sqrt(v4))
> d5 = L51*F1 + L52*F2 + rnorm(n,0,sqrt(v5))
> d6 = L61*F1 + L62*F2 + rnorm(n,0,sqrt(v6))
> d7 = L71*F1 + L72*F2 + rnorm(n,0,sqrt(v7))
> d8 = L81*F1 + L82*F2 + rnorm(n,0,sqrt(v8))
> dmat = cbind(d1,d2,d3,d4,d5,d6,d7,d8)
```

We fit a two-factor model by maximum likelihood, with a varimax rotation.

```
> factanal(dmat,factors=2,rotation='varimax')

Call:
factanal(x = dmat, factors = 2, rotation = "varimax")

Uniquenesses:
   d1    d2    d3    d4    d5    d6    d7    d8
0.506 0.510 0.519 0.511 0.507 0.505 0.508 0.510

Loadings:
   Factor1 Factor2
d1          0.698
d2          0.694
d3          0.688
d4          0.695
d5  0.697
d6  0.699
d7  0.696
d8  0.695

                Factor1 Factor2
SS loadings       1.971   1.953
Proportion Var    0.246   0.244
Cumulative Var    0.246   0.491

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 10.22 on 13 degrees of freedom.
The p-value is 0.676
```

It is arbitrary which factor is called Factor 1 and which is called Factor 2. Other than that, all the estimates are right on the money. The model is correct, and it fits. Everything is perfect.

In the second example, the true pattern of factor loadings is not at all like varimax. Everything else is very similar to the first example.

```
> # Truth is not like varimax (All communalities = 0.50)
> # Factor loadings
> L11 = 0.5; L12 = -0.5
> L21 = 0.5; L22 = -0.5
> L31 = 0.5; L32 = -0.5
> L41 = 0.5; L42 = -0.5
> L51 = 0.5; L52 =  0.5
> L61 = 0.5; L62 =  0.5
> L71 = 0.5; L72 =  0.5
> L81 = 0.5; L82 =  0.5
> # Error Variances
> v1 = 1 - L11**2 - L12**2
> v2 = 1 - L21**2 - L22**2
> v3 = 1 - L31**2 - L32**2
> v4 = 1 - L41**2 - L42**2
> v5 = 1 - L51**2 - L52**2
> v6 = 1 - L61**2 - L62**2
> v7 = 1 - L71**2 - L72**2
> v8 = 1 - L81**2 - L82**2
> # Generate data
> set.seed(8888)
> F1 = rnorm(n,0,1); F2 = rnorm(n,0,1)
> d1 = L11*F1 + L12*F2 + rnorm(n,0,sqrt(v1))
> d2 = L21*F1 + L22*F2 + rnorm(n,0,sqrt(v2))
> d3 = L31*F1 + L32*F2 + rnorm(n,0,sqrt(v3))
> d4 = L41*F1 + L42*F2 + rnorm(n,0,sqrt(v4))
> d5 = L51*F1 + L52*F2 + rnorm(n,0,sqrt(v5))
> d6 = L61*F1 + L62*F2 + rnorm(n,0,sqrt(v6))
> d7 = L71*F1 + L72*F2 + rnorm(n,0,sqrt(v7))
> d8 = L81*F1 + L82*F2 + rnorm(n,0,sqrt(v8))
> dmat = cbind(d1,d2,d3,d4,d5,d6,d7,d8)
```

Again we fit a two-factor model with a varimax rotation.

```
> notsimple = factanal(dmat,factors=2,rotation='varimax'); notsimple

Call:
factanal(x = dmat, factors = 2, rotation = "varimax")

Uniquenesses:
   d1    d2    d3    d4    d5    d6    d7    d8
0.496 0.496 0.504 0.504 0.497 0.495 0.503 0.499

Loadings:
   Factor1 Factor2
d1          0.708
d2          0.708
d3          0.702
d4          0.702
d5  0.708
d6  0.709
d7  0.703
d8  0.706
```

```
                Factor1 Factor2
SS loadings       2.007   2.000
Proportion Var    0.251   0.250
Cumulative Var    0.251   0.501

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 9.58 on 13 degrees of freedom.
The p-value is 0.728
```

This time, only the estimates of communality (which are identifiable) and the goodness of fit test perform well. Everything else is awful. In particular, the estimates of the loadings are very similar to the estimates in the first example, and very far from the truth.

While the factor analysis for this second example clearly failed to yield a good estimate, it did yield a set of numbers that are only an orthogonal rotation away from an estimate that is very good indeed. If you think of the likelihood function as a high-dimensional mountain range, the maximum elevation is attained on a sort of ridge, with all points on the ridge at the same altitude. As the sample size increases, the ridge gets higher and higher, and its location changes a little bit, but less and less with increasing $n$. Meanwhile, the rest of the landscape melts into a featureless plain. The initial constrained maximum likelihood estimation lands you at one point on the ridge, and then an orthogonal rotation walks you along the ridge (say there is a path along the ridge)[24]. In these simulated data, the path actually passes very close to the true parameter value — very close indeed, since the sample size in this simulated data set is so large.

To find the point on the path that is closest to where the treasure is hidden, we will rotate the factor solution in the second example using a criterion that has not been mentioned before now. We will carry out a *Procrustes* rotation[25]. In Procustes rotation, the rotation matrix is chosen to minimize the difference between the matrix of estimated loadings and a target matrix, using least squares. There are orthogonal and oblique versions of Procrustes rotation. For our present purposes we want an orthogonal version. The `MCMCpack` package has a good one.

```
> # Procrustes rotation
> # install.packages("MCMCpack", dependencies=TRUE) # Only need to do this once
> library(MCMCpack)
Loading required package: coda
Loading required package: MASS
##
## Markov Chain Monte Carlo Package (MCMCpack)
```

[24] In this picture of the likelihood function, what is simple structure? Parameter values are literally coordinates, like latitude and longitude. This means that choosing a simple structure is like choosing a "good" location on the path, based on pleasing numerical values for the co-ordinates. For example, both latitude and longitude are integers, or divisible by eight. It's a lucky spot; let's stop here.

[25] Procrustes is a character in classic Greek mythology. He was a very bad man who would invite travellers to a free dinner and bed at his castle. Everybody fit the bed in the guest room, one way or the other. If travellers were too short, Procrustes would hammer them and stretch them with ropes until they fit. If they were too tall, he would cut off their feet. The survival rate for his guests was essentially zero. Then one day Theseus came along and gave Procrustes a taste of his own medicine.

```
## Copyright (C) 2003-2021 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
##
## Support provided by the U.S. National Science Foundation
## (Grants SES-0350646 and SES-0350613)
##
> # help(procrustes)
> L = notsimple$loadings; print(L,cutoff=0) # Factor loadings for the second example
Loadings:
   Factor1 Factor2
d1  0.047   0.708
d2  0.056   0.708
d3  0.054   0.702
d4  0.052   0.702
d5  0.708  -0.050
d6  0.709  -0.054
d7  0.703  -0.052
d8  0.706  -0.052

               Factor1 Factor2
SS loadings      2.007   2.000
Proportion Var   0.251   0.250
Cumulative Var   0.251   0.501
```

The target matrix will be the matrix of true factor loadings. Of course we can only do this because it's a simulation, and we know what the true parameter values are.

```
> Lambda = rbind(c(L11,L12),               # True factor loadings
+                c(L21,L22),
+                c(L31,L32),
+                c(L41,L42),
+                c(L51,L52),
+                c(L61,L62),
+                c(L71,L72),
+                c(L81,L82)  )
> Lambda # True Lambda -- How close can we get to this?
     [,1] [,2]
[1,]  0.5 -0.5
[2,]  0.5 -0.5
[3,]  0.5 -0.5
[4,]  0.5 -0.5
[5,]  0.5  0.5
[6,]  0.5  0.5
[7,]  0.5  0.5
[8,]  0.5  0.5
```

Now carry out the Procrustes rotation.

```
> pro = procrustes(X = L, Xstar = Lambda) # Rotate X to approximate Xstar.
> pro$X.new
        [,1]       [,2]
d1 0.4981332 -0.5056613
d2 0.5049341 -0.4994469
d3 0.4994946 -0.4962512
```

```
d4 0.4978127 -0.4979441
d5 0.5033950  0.5000182
d6 0.5014220  0.5038790
d7 0.4986956  0.4981095
d8 0.5003778  0.5008127
```

That's *very* close to the target. To see how close, look at it rounded and compare the result to `Lambda` above.

```
> round(pro$X.new,2)
   [,1]  [,2]
d1  0.5 -0.51
d2  0.5 -0.50
d3  0.5 -0.50
d4  0.5 -0.50
d5  0.5  0.50
d6  0.5  0.50
d7  0.5  0.50
d8  0.5  0.50
```

To really see how impressive this is, note that a Procruste rotation cannot fit an arbitrary target very well. In the final part of this example, the matrix `M` contains factor loadings that produce a covariance matrix very different from the one produced by `Lambda`. Can we rotate to fit this one?

```
> M  = rbind(c(0.30,0.64),
+            c(0.30,0.64),
+            c(0.30,0.64),
+            c(0.30,0.64),
+            c(0.30,0.64),
+            c(0.30,0.64),
+            c(0.30,0.64)  ); M
     [,1] [,2]
[1,]  0.3 0.64
[2,]  0.3 0.64
[3,]  0.3 0.64
[4,]  0.3 0.64
[5,]  0.3 0.64
[6,]  0.3 0.64
[7,]  0.3 0.64
[8,]  0.3 0.64
> procrustes(X = L, Xstar = M)$X.new
         [,1]      [,2]
d1 -0.2470153 0.6654424
d2 -0.2385050 0.6689701
d3 -0.2379146 0.6626890
d4 -0.2401607 0.6618827
d5  0.6661897 0.2441638
d6  0.6688511 0.2407410
d7  0.6624699 0.2407156
d8  0.6656312 0.2410942
```

So the closest one can get to this particular target with an orthogonal rotation is ridiculously far away.

The main point here that for the second simulated data example, the one where varimax rotation failed, an excellent estimate is actually somewhere in the collection of factor matrices that can be reached by an orthogonal rotation. The problem is that we don't know which one. This is true for real data sets, too. I cannot think of any exceptions.

**Oblique rotations** If anything, the problem is a bit worse with oblique rotations, because they can miss the truth and find an inferior solution, even if the true factor loadings typify simple structure. For the next example, there will be three factors. The first factor is independent of the others, but factors two and three are highly correlated. There are nine observed variables. The first three variables load only on factor one, the second three load only on factor two, and the last three load only on factor three. That's a clear example of simple structure.

```
> Phi = rbind(c(1.0, 0.0, 0.0),
+             c(0.0, 1.0, 0.9),
+             c(0.0, 0.9, 1.0))
>
> Lambda = rbind(c(0.9, 0.0, 0.0),
+                c(0.9, 0.0, 0.0),
+                c(0.9, 0.0, 0.0),
+                c(0.0, 0.9, 0.0),
+                c(0.0, 0.9, 0.0),
+                c(0.0, 0.9, 0.0),
+                c(0.0, 0.0, 0.9),
+                c(0.0, 0.0, 0.9),
+                c(0.0, 0.0, 0.9)  )
```

The standardized model will hold exactly in the population. For this, it is necessary to calculate the matrix $\mathbf{\Omega}$ in $cov(\mathbf{z}) = cov(\mathbf{\Lambda F} + \mathbf{e}) = \mathbf{\Lambda \Phi \Lambda}^\top + \mathbf{\Omega}$.

```
> # Calculate Omega, the 9 x 9 covariance matrix of the error terms.
> Lambda %*% Phi %*% t(Lambda)
      [,1] [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
 [1,] 0.81 0.81 0.81 0.000 0.000 0.000 0.000 0.000 0.000
 [2,] 0.81 0.81 0.81 0.000 0.000 0.000 0.000 0.000 0.000
 [3,] 0.81 0.81 0.81 0.000 0.000 0.000 0.000 0.000 0.000
 [4,] 0.00 0.00 0.00 0.810 0.810 0.810 0.729 0.729 0.729
 [5,] 0.00 0.00 0.00 0.810 0.810 0.810 0.729 0.729 0.729
 [6,] 0.00 0.00 0.00 0.810 0.810 0.810 0.729 0.729 0.729
 [7,] 0.00 0.00 0.00 0.729 0.729 0.729 0.810 0.810 0.810
 [8,] 0.00 0.00 0.00 0.729 0.729 0.729 0.810 0.810 0.810
 [9,] 0.00 0.00 0.00 0.729 0.729 0.729 0.810 0.810 0.810
> diag(Lambda %*% Phi %*% t(Lambda))
[1] 0.81 0.81 0.81 0.81 0.81 0.81 0.81 0.81 0.81
> Omega = diag(1-0.81,nrow=9,ncol=9); Omega
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
```

```
 [1,] 0.19 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 [2,] 0.00 0.19 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 [3,] 0.00 0.00 0.19 0.00 0.00 0.00 0.00 0.00 0.00
 [4,] 0.00 0.00 0.00 0.19 0.00 0.00 0.00 0.00 0.00
 [5,] 0.00 0.00 0.00 0.00 0.19 0.00 0.00 0.00 0.00
 [6,] 0.00 0.00 0.00 0.00 0.00 0.19 0.00 0.00 0.00
 [7,] 0.00 0.00 0.00 0.00 0.00 0.00 0.19 0.00 0.00
 [8,] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.19 0.00
 [9,] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.19
```

Now we will generate the random data set. We need a function for simulating multivariate normal data. Such a function is available in several packages, but I prefer one that I wrote; it is available for download and free for public use under the usual GNU conditions. As you can see from the code, it uses spectral decomposition to transform a set of standard normals into a multivariate normal.

```
> rm(list=ls())
> # Need function for simulating multivariate normal data.
> source("http://www.utstat.toronto.edu/~brunner/Rfunctions/rmvn.txt")
> rmvn # Type the function name to see the code.
function(nn,mu,sigma)
# Returns an nn by kk matrix, rows are independent MVN(mu,sigma)

    kk <- length(mu)
    dsig <- dim(sigma)
    if(dsig[1] != dsig[2]) stop("Sigma must be square.")
    if(dsig[1] != kk) stop("Sizes of sigma and mu are inconsistent.")
    ev <- eigen(sigma)
    sqrl <- diag(sqrt(ev$values))
    PP <- ev$vectors
    ZZ <- rnorm(nn*kk) ; dim(ZZ) <- c(kk,nn)
    rmvn <- t(PP%*%sqrl%*%ZZ+mu)
    rmvn
```

In the simulation below, the large sample size of $n = 10,000$ means that the results will not be blurred much by sampling error.

```
> # Generate data
> set.seed(9999)
> n = 10000
> Fac = rmvn(n,mu=c(0,0,0),sigma=Phi)     # n x 3 matrix of factor values
> err = rmvn(n,mu=numeric(9),sigma=Omega) # n x 9 matrix of error terms
>
> #     n x 3       3 x 9     n x 9
> dat =  Fac  %*% t(Lambda) +  err
```

Compare the sample correlation matrix to the true correlation matrix. They are close, as one would expect with this sample size. Of course this is a way to check for mistakes in calculation or programming.

```
> round(cor(dat),3)
       [,1]   [,2]   [,3]   [,4]   [,5]  [,6]   [,7]   [,8]   [,9]
 [1,]  1.000  0.808  0.806 -0.004 -0.001 0.008  0.004  0.003 -0.002
 [2,]  0.808  1.000  0.809 -0.001 -0.006 0.000 -0.002 -0.003 -0.006
 [3,]  0.806  0.809  1.000 -0.002 -0.007 0.003 -0.002 -0.001 -0.004
 [4,] -0.004 -0.001 -0.002  1.000  0.811 0.809  0.731  0.732  0.730
 [5,] -0.001 -0.006 -0.007  0.811  1.000 0.812  0.726  0.728  0.725
 [6,]  0.008  0.000  0.003  0.809  0.812 1.000  0.729  0.730  0.726
 [7,]  0.004 -0.002 -0.002  0.731  0.726 0.729  1.000  0.810  0.809
 [8,]  0.003 -0.003 -0.001  0.732  0.728 0.730  0.810  1.000  0.808
 [9,] -0.002 -0.006 -0.004  0.730  0.725 0.726  0.809  0.808  1.000
> Lambda %*% Phi %*% t(Lambda) + Omega # Compare
      [,1] [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
 [1,] 1.00 0.81 0.81 0.000 0.000 0.000 0.000 0.000 0.000
 [2,] 0.81 1.00 0.81 0.000 0.000 0.000 0.000 0.000 0.000
 [3,] 0.81 0.81 1.00 0.000 0.000 0.000 0.000 0.000 0.000
 [4,] 0.00 0.00 0.00 1.000 0.810 0.810 0.729 0.729 0.729
 [5,] 0.00 0.00 0.00 0.810 1.000 0.810 0.729 0.729 0.729
 [6,] 0.00 0.00 0.00 0.810 0.810 1.000 0.729 0.729 0.729
 [7,] 0.00 0.00 0.00 0.729 0.729 0.729 1.000 0.810 0.810
 [8,] 0.00 0.00 0.00 0.729 0.729 0.729 0.810 1.000 0.810
 [9,] 0.00 0.00 0.00 0.729 0.729 0.729 0.810 0.810 1.000
```

When I decided on this example, I thought that common methods for determining the number of factors might fail, because it could be hard to tell the two highly correlated factors from a single factor. Indeed, there are two eigenvalues greater than one, and the others are not even close; this points to two factors. Other common tests for number of factors give varying results. However, testing for goodness of fit performed really well. Fitting a model with just two factors,

```
> factanal(dat,factors=2)

Call:
factanal(x = dat, factors = 2)

Uniquenesses:
[1] 0.195 0.189 0.192 0.235 0.239 0.238 0.240 0.239 0.243

Loadings:
      Factor1 Factor2
 [1,]          0.897
 [2,]          0.900
 [3,]          0.899
 [4,]  0.875
 [5,]  0.872
 [6,]  0.873
 [7,]  0.872
 [8,]  0.873
 [9,]  0.870

                Factor1 Factor2
SS loadings       4.566   2.424
```

```
Proportion Var   0.507   0.269
Cumulative Var   0.507   0.777

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 3179.9 on 19 degrees of freedom.
The p-value is 0
```

Varimax was fooled into thinking that the last six variables all came from the same factor, but the two-factor model did not come close to fitting. Trying a three-factor model,

```
> factanal(dat,factors=3)

Call:
factanal(x = dat, factors = 3)

Uniquenesses:
[1] 0.195 0.189 0.192 0.193 0.185 0.190 0.188 0.192 0.193

Loadings:
      Factor1 Factor2 Factor3
 [1,]          0.897
 [2,]          0.900
 [3,]          0.899
 [4,]  0.878          -0.191
 [5,]  0.878          -0.212
 [6,]  0.877          -0.201
 [7,]  0.877           0.205
 [8,]  0.877           0.197
 [9,]  0.875           0.204

               Factor1 Factor2 Factor3
SS loadings      4.615   2.424   0.244
Proportion Var   0.513   0.269   0.027
Cumulative Var   0.513   0.782   0.809

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 12.73 on 12 degrees of freedom.
The p-value is 0.389
```

This model fits nicely; the goodness of fit test located the true number of factors. This has been my experience with uncorrelated factors, too. The chi-squared test for goodness of fit is an excellent tool for determining the number of factors, and the larger the sample size, the better it gets — with simulated data. Of course that's the problem. We must bear in mind the suggestion that for any real data set, there could easily be hundreds of common factors. When this is true, no model will fit if the sample size is large enough.

In any case, suppose we know that there are three factors. The true matrix of factor loadings is an extreme example of simple structure. Can oblimin find it? If the factors were uncorrelated, one coud trust varimax to locate this easy truth. The first attempt will use the default setting of $\gamma = 0$.

```
> # install.packages("GPArotation", dependencies=TRUE) # Only need to do this once
> library(GPArotation)
>
> threefac = factanal(dat,factors=3,rotation='none'); Ahat = threefac$loadings
> options(scipen=999) # Suppress scientific notation for now
> oblimin(Ahat)
Oblique rotation method Oblimin Quartimin converged.
Loadings:
        Factor1  Factor2    Factor3
 [1,]  0.002820  0.89720  0.0024216
 [2,] -0.001803  0.90038 -0.0022990
 [3,] -0.000956  0.89891  0.0000235
 [4,]  0.878590 -0.00141 -0.1905515
 [5,]  0.878124 -0.00382 -0.2111222
 [6,]  0.877928  0.00538 -0.2005067
 [7,]  0.876644  0.00142  0.2058478
 [8,]  0.876572  0.00120  0.1972533
 [9,]  0.874316 -0.00314  0.2046256

Rotating matrix:
         [,1]      [,2]       [,3]
[1,]  1.00000 -0.001673 -0.0003306
[2,]  0.00307  1.000000 -0.0000706
[3,] -0.00201  0.000551  1.0000028

Phi:
         [,1]      [,2]      [,3]
[1,]  1.00000 -0.001395  0.002345
[2,] -0.00140  1.000000 -0.000484
[3,]  0.00234 -0.000484  1.000000
```

Both the $\widehat{\mathbf{\Lambda}}$ and $\widehat{\mathbf{\Phi}}$ matrices are way off. $\widehat{\mathbf{\Phi}}$ is nearly the identity, and $\widehat{\mathbf{\Lambda}}$ is essentially the varimax solution. Increasing the value of $\gamma$ to encourage more highly correlated factors,

```
> oblimin(Ahat, gam = 0.5) # For more highly correlated factors (truth).
Oblique rotation method Oblimin Biquartimin converged.
Loadings:
      Factor1 Factor2 Factor3
 [1,]   0.117  1.0222   0.220
 [2,]   0.114  1.0234   0.215
 [3,]   0.114  1.0227   0.217
 [4,]   0.978  0.0304  -0.291
 [5,]   0.983  0.0197  -0.317
 [6,]   0.981  0.0342  -0.301
 [7,]   0.864  0.1860   0.195
 [8,]   0.866  0.1825   0.184
 [9,]   0.861  0.1801   0.193

Rotating matrix:
       [,1]  [,2]    [,3]
[1,]  1.052 0.118 -0.0661
[2,]  0.131 1.138  0.2414
```

```
[3,] -0.286 0.385  1.2240

Phi:
       [,1]   [,2]   [,3]
[1,]  1.000 -0.313  0.396
[2,] -0.313  1.000 -0.551
[3,]  0.396 -0.551  1.000
```

Once again, the results are nowhere near the true parameter values. Increasing the value of $\gamma$ once again,

```
> oblimin(Ahat, gam = 0.75)
Oblique rotation method Oblimin g=0.75 NOT converged.
Loadings:
      Factor1 Factor2 Factor3
 [1,]    4.31   9.678    4.72
 [2,]    4.28   9.650    4.72
 [3,]    4.28   9.662    4.73
 [4,]    7.81  -0.429   -8.08
 [5,]    7.76  -0.685   -8.30
 [6,]    7.82  -0.468   -8.14
 [7,]    8.43   4.017   -4.01
 [8,]    8.42   3.919   -4.10
 [9,]    8.39   3.950   -4.03

Rotating matrix:
     [,1]  [,2]  [,3]
[1,] 9.23  1.93 -6.99
[2,] 4.80 10.76  5.24
[3,] 1.57 11.15 10.21

Phi:
       [,1]   [,2]   [,3]
[1,]  1.000 -0.995  0.994
[2,] -0.995  1.000 -0.997
[3,]  0.994 -0.997  1.000
Warning message:
In GPFoblq(L, Tmat = Tmat, normalize = normalize, eps = eps, maxit = maxit,  :
  convergence not obtained in GPFoblq. 1000 iterations used.
```

This time, the algorithm did not converge (this is common with "large" positive values of $\gamma$), and the estimates are to be ignored. They are just the current values when the job ran out of iterations. The value of the oblimin criterion was marching off to $-\infty$.

Lowering the value of $\gamma$ a bit,

```
> oblimin(Ahat, gam = 0.6)
Oblique rotation method Oblimin g=0.6 converged.
Loadings:
      Factor1 Factor2   Factor3
 [1,]   0.306  1.3650  0.427736
 [2,]   0.301  1.3641  0.423094
 [3,]   0.301  1.3642  0.425963
```

```
 [4,]   1.213  0.1018 -0.651437
 [5,]   1.216  0.0792 -0.686801
 [6,]   1.217  0.1028 -0.664617
 [7,]   1.138  0.4683  0.013499
 [8,]   1.140  0.4600 -0.000951
 [9,]   1.134  0.4595  0.010177

Rotating matrix:
       [,1]  [,2]   [,3]
[1,]  1.341 0.314 -0.379
[2,]  0.341 1.519  0.472
[3,] -0.188 0.915  1.673

Phi:
       [,1]   [,2]   [,3]
[1,]  1.000 -0.669  0.659
[2,] -0.669  1.000 -0.812
[3,]  0.659 -0.812  1.000
```

This time, the maximum absolute correlation between factors is in the right vicinity, but the values of the estimated correlations are way off, and the estimated factor loadings are nowhere near the truth.

It is clear that adjusting the value of $\gamma$ does not help at all. Possibly the numerical search is getting caught in a local minimum. By default, the search starts with the transformation matrix $\mathbf{T}$ equal to the identity. Using a combination of calculation and guesswork (the details are not important), I came up with a promising $\mathbf{T}$ matrix, denoted by `T_try`.

```
> T_try
                [,1]         [,2]        [,3]
Factor1 -0.001603061  0.975610583 0.973941825
Factor2  0.999998098  0.002998459 0.002938147
Factor3  0.001110803 -0.219488039 0.226778941
```

This transformation matrix reproduces the true correlations between factors and the true factor loadings quite well. Checking $\mathbf{T}^\top\mathbf{T} = \mathbf{\Phi}$,

```
# Test T_try
> M = t(T_try) %*% T_try; round(M,2)
     [,1] [,2] [,3]
[1,]    1  0.0  0.0
[2,]    0  1.0  0.9
[3,]    0  0.9  1.0
> Phi
     [,1] [,2] [,3]
[1,]    1  0.0  0.0
[2,]    0  1.0  0.9
[3,]    0  0.9  1.0
```

The match is perfect, to two decimal places. Now try $\mathbf{\Lambda} = \mathbf{A}\left(\mathbf{T}^\top\right)^{-1}$.

```
> round(Ahat %*% solve(t(T_try)), 2) # Compare Lambda
      [,1] [,2]  [,3]
 [1,]  0.9 0.00  0.00
 [2,]  0.9 0.01 -0.01
 [3,]  0.9 0.00  0.00
 [4,]  0.0 0.88  0.02
 [5,]  0.0 0.93 -0.03
 [6,]  0.0 0.91 -0.01
 [7,]  0.0 0.00  0.90
 [8,]  0.0 0.01  0.89
 [9,]  0.0 0.00  0.90
> Lambda
      [,1] [,2] [,3]
 [1,]  0.9  0.0  0.0
 [2,]  0.9  0.0  0.0
 [3,]  0.9  0.0  0.0
 [4,]  0.0  0.9  0.0
 [5,]  0.0  0.9  0.0
 [6,]  0.0  0.9  0.0
 [7,]  0.0  0.0  0.9
 [8,]  0.0  0.0  0.9
 [9,]  0.0  0.0  0.9
```

The reason it's possible to approximate $\mathbf{\Phi}$ and $\mathbf{\Lambda}$ so well is the large sample size. Of course, as in the orthogonal case, there are infinitely many other $\mathbf{T}$ matrices that fit the data equally well. When `T_try` is used as a starting value, the simple structure in $\widehat{\mathbf{\Lambda}}$ ensures that the numerical search stays very close to where it started.

```
> # Use T_try as a starting value
> oblimin(Ahat, Tmat=T_try, gam=0)
Oblique rotation method Oblimin Quartimin converged.
Loadings:
       Factor1   Factor2  Factor3
 [1,]  0.89721 -0.003859  0.00671
 [2,]  0.90038  0.004258 -0.00617
 [3,]  0.89891 -0.000471 -0.00056
 [4,] -0.00139  0.876491  0.02450
 [5,] -0.00381  0.922002 -0.02156
 [6,]  0.00540  0.898292  0.00198
 [7,]  0.00159 -0.006151  0.90648
 [8,]  0.00136  0.012927  0.88730
 [9,] -0.00297 -0.004632  0.90257

Rotating matrix:
             [,1]     [,2]    [,3]
Factor1 -0.001572  0.51597 0.51025
Factor2  1.000000  0.00182 0.00127
Factor3  0.000933 -2.22516 2.22648

Phi:
         [,1]     [,2]     [,3]
[1,]  1.00000 -0.00122 -0.00159
```

```
[2,] -0.00122  1.00000  0.89908
[3,] -0.00159  0.89908  1.00000
```

One could not ask for nicer results. Notice how a large value of $\gamma$ is not necessary to get a high estimated correlation between factors. Furthermore, the oblimin criterion is actually *lower* for this solution than for the one with the default starting value, so the earlier search found a local minimum that was higher than the global minimum. Here's how to tell.

An `oblimin` object is a list, and one of the items in the list is a table showing the iteration history. For some reason, the table is called `Table`. The second column of the table gives the value of the oblimin criterion. There is one row in the table for each iteration, so tables can be quite long. We will use R's `tail` function to look at just the last four lines of the tables. First comes the one with the default starting value for $\mathbf{T}$ (the identity), and then the one starting with `T_try`.

```
> tail(oblimin(Ahat, gam=0)$Table)
      [,1]       [,2]      [,3] [,4]
[91,]   90 0.09396019 -4.936630 0.50
[92,]   91 0.09396019 -4.947460 0.50
[93,]   92 0.09396019 -4.958262 0.50
[94,]   93 0.09396019 -4.969037 0.50
[95,]   94 0.09396019 -4.745909 1.00
[96,]   95 0.09396019 -5.054387 0.25
> tail(oblimin(Ahat, Tmat=T_try, gam=0)$Table)
      [,1]         [,2]      [,3]   [,4]
[38,]   37 0.0005909207 -4.741182 0.1250
[39,]   38 0.0005909207 -4.949552 0.0625
[40,]   39 0.0005909207 -4.985353 0.1250
[41,]   40 0.0005909207 -4.991386 0.1250
[42,]   41 0.0005909207 -4.950799 0.1250
[43,]   42 0.0005909207 -5.158412 0.0625
```

Starting with `T_try` was possible only because I knew the true $\mathbf{\Lambda}$ matrix. The key to finding such a hidden solution with real data (if one exists) is to try different starting values for $\mathbf{T}$. The `GPArotation` package has a useful function called `Random.Start`, which generates a random orthogonal matrix. The single argument of the function `Random.Start` is the number of rows and columns. While the transformation matrix $\mathbf{T}$ is not constrained to be orthogonal, the non-zero off-diagonal elements mix things up enough so that it works quite well. What I did was to execute the following code repeatedly until something interesting happened.

```
> oblimin(Ahat, Tmat=Random.Start(3))
```

After just three tries, I got the following.

```
Oblique rotation method Oblimin Quartimin converged.
Loadings:
       Factor1   Factor2   Factor3
 [1,] -0.89721  0.006711  0.003860
```

```
 [2,] -0.90038 -0.006167 -0.004258
 [3,] -0.89891 -0.000561  0.000472
 [4,]  0.00139  0.024497 -0.876491
 [5,]  0.00381 -0.021562 -0.922002
 [6,] -0.00540  0.001983 -0.898292
 [7,] -0.00159  0.906484  0.006151
 [8,] -0.00136  0.887303 -0.012927
 [9,]  0.00297  0.902572  0.004632

Rotating matrix:
          [,1]    [,2]     [,3]
[1,]  0.001572 0.51025 -0.51597
[2,] -1.000000 0.00127 -0.00182
[3,] -0.000933 2.22648  2.22516

Phi:
         [,1]     [,2]     [,3]
[1,]  1.00000  0.00159 -0.00122
[2,]  0.00159  1.00000 -0.89908
[3,] -0.00122 -0.89908  1.00000
```

This is the same solution obtained using `T_try` as a starting value, except that the the signs of all the factor loadings for factors one and three are reversed, and the correlation between factors two and three is negative instead of positive. This is perfectly good. Since the oblimin criterion is a function of the *squared* factor loadings, switching the signs of the loadings in any column produces the same value of the function being minimized, and the function has at least $2^p$ local minima. As in orthogonal factor analysis, one may reflect factors at will, and the only consequence is the word one uses to describe the factor. One may call it "anti-racism" instead of "racism," or "mental health" instead of "mental illness." It is entirely a matter of convenience. Naturally, when one does this one must also switch the signs of the correlations between the factor in question and all the other factors. That is what has happened here.

Continuing to execute the code, on the eleventh try I got a version of the correct solution with only factor three reflected, and on the thirteenth try I got a version with only factor one reflected.

The conclusion is that if the true factor pattern has a simple structure, oblimin rotation may miss it unless one tries numerous starting values[26]. In fact, even if the truth has a fairly simple structure, there may be another solution that fits the data just as well and which has a structure that is even simpler. In this case, multiple starting values will lead you to the answer that is prettier, but wrong. Of course, if the truth does not happen to be simple, there is no hope at all.

---

Frequently, simulation studies involve thousands, or even millions of random data sets. Here, you really only need two simulated data sets to see how unsuccessful exploratory

[26] I tried `Random.Start` a large number of times with the Mind-body data, and got the same results each time apart from reflections.

factor analysis can be. It all depends on how closely the true pattern of factor loadings approximates simple structure. If the truth looks like the result of a varimax rotation, then a varimax rotation will probably find it – or it will find something equivalent, with one or more factors reflected. If the truth does not resemble a varimax rotation, then a varimax rotation will settle on a simple structure that may be quite different from the truth. Should we expect the truth in any particular field to resemble simple structure? I really can't see why.

The factor analysts have an answer, and it goes back to the early days, from the time when the indeterminacy of factor solutions was first recognized. The argument is that a factor solution is essentially a scientific theory of the data. In the philosophy of science, it is widely accepted that there can be many different theories that fit a set of data equally well. In this situation, a principle known as Occam's razor says that all other things being equal, a simpler explanation is better. Thus, the simple structure located by a varimax or some other good rotation method is the preferred estimate.

My response is that while the factor analysis model itself is like a scientific theory, the unknown constants in the model are numerical quantities that are subject to estimation, like the speed of light in relativity theory. In the case of unconstrained exploratory factor analysis, lack of parameter identifiability means that there are infinitely many potential estimates that are equally reasonable given the data. Choosing a set of numerical values that tells a pleasing story is one option, but the truth may be much closer to a completely different set of values – one that is equally compatible with the data. Viewed as a method of statistical estimation, exploratory factor analysis is a failure, period. It is something a statistician should never do, except perhaps for money.

## 2.7 Rotating Principal Components

Something can be salvaged from all this. Rotation is what makes factor analysis results understandable. In R, a nice thing about the stand-alone `varimax` function is that it can also be used to rotate principal components. The result is a set of uncorrelated linear combinations of the variables that explain exactly the same amount of variance as the original components, but are easier to interpret. This section is a bit of a digression, but the end product is a useful data analysis trick.

From Section 2.1, we have the $k \times 1$ standardized data vector $\mathbf{z}$, the correlation matrix $cov(\mathbf{z}) = \boldsymbol{\Sigma}$, the spectral decomposition $\boldsymbol{\Sigma} = \mathbf{CDC}^\top$, and the vector of principal components $\mathbf{y} = \mathbf{C}^\top\mathbf{z}$. The ordered eigenvalues in the diagonal matrix $\mathbf{D}$ are both the variances of the principal components and the amounts of variance in $\mathbf{z}$ that they explain. It is

helpful to calculate the matrix of correlations

$$\begin{aligned} corr(\mathbf{z},\mathbf{y}) &= cov(\mathbf{z},\mathbf{D}^{-1/2}\mathbf{y}) \\ &= cov(\mathbf{z},\mathbf{D}^{-1/2}\mathbf{C}^\top\mathbf{z}) \\ &= cov(\mathbf{z})\left(\mathbf{D}^{-1/2}\mathbf{C}^\top\right)^\top \\ &= \mathbf{\Sigma}\mathbf{C}\mathbf{D}^{-1/2} \\ &= \mathbf{C}\mathbf{D}\underbrace{\mathbf{C}^\top\mathbf{C}}_{\mathbf{I}}\mathbf{D}^{-1/2} \\ &= \mathbf{C}\mathbf{D}\mathbf{D}^{-1/2} \\ &= \mathbf{C}\mathbf{D}^{1/2}, \end{aligned} \tag{2.25}$$

a formula equivalent to the scalar version (2.3).

We don't retain all the principal components. Instead, we summarize the variables with a smaller set of $p$ principal components that explain a good part of the total variance. Typically, components associated with eigenvalues greater than one are retained. This may be accomplished with a $p\times k$ *selection matrix* that will be denoted by $\mathbf{S}$ (for selection), and is not to be mistaken for a sample covariance matrix. Each row of $\mathbf{S}$ has a one in the position of a component to be retained, and the rest zeros. For example, if there were five principal components, the first two may be selected as follows.

$$\mathbf{S}\mathbf{y} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

If $\mathbf{A}$ is any $k \times k$ matrix, then $\mathbf{S}\mathbf{A}\mathbf{S}^\top$ is the $p \times p$ sub-matrix with rows and columns indicated by $\mathbf{S}$. A sub-matrix of the identity is another (smaller) identity matrix, so $\mathbf{S}\mathbf{S}^\top = \mathbf{I}_p$. Selection matrices are quite flexible and can even be used to re-order variables, but here they will just be used to select the first $p$ principal components.

Simply rotating a set of selected principal components is not a good choice, because the resulting linear combinations are correlated.

$$\begin{aligned} cov(\mathbf{R}\mathbf{S}\mathbf{y}) &= \mathbf{R}\mathbf{S}cov(\mathbf{y})\left(\mathbf{R}\mathbf{S}\right)^\top \\ &= \mathbf{R}\mathbf{S}\mathbf{D}\mathbf{S}^\top\mathbf{R}^\top, \end{aligned}$$

a matrix that in general will not be diagonal unless all the eigenvalues equal one. Because the eigenvalues are the variances of the principal components, this suggests standardizing the principal components before rotating them. It is more convenient (mathematically, not computationally) to standardize first, and then select. The result is

$$\begin{aligned} \mathbf{f} &= \mathbf{S}\mathbf{D}^{-1/2}\mathbf{y} \\ &= \mathbf{S}\mathbf{D}^{-1/2}\mathbf{C}^\top\mathbf{z}. \end{aligned}$$

The notation $\mathbf{f}$ is meant to suggest that the standardized principal components are analogous to factors, even though they are not really factors.

Applying a rotation to $\mathbf{f}$, we have $\mathbf{f}' = \mathbf{R}\mathbf{f}$, with covariance matrix

$$\begin{aligned} cov(\mathbf{f}') &= cov(\mathbf{R}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{C}^\top\mathbf{z}) \\ &= \mathbf{R}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{C}^\top cov(\mathbf{z})\left(\mathbf{R}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{C}^\top\right)^\top \\ &= \mathbf{R}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{C}^\top\boldsymbol{\Sigma}\mathbf{C}\mathbf{D}^{-1/2}\mathbf{S}^\top\mathbf{R}^\top \\ &= \mathbf{R}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{C}^\top\mathbf{C}\mathbf{D}\mathbf{C}^\top\mathbf{C}\mathbf{D}^{-1/2}\mathbf{S}^\top\mathbf{R}^\top \\ &= \mathbf{R}\mathbf{S}\mathbf{D}^{-1/2}\underbrace{\mathbf{C}^\top\mathbf{C}}_{\mathbf{I}}\mathbf{D}\underbrace{\mathbf{C}^\top\mathbf{C}}_{\mathbf{I}}\mathbf{D}^{-1/2}\mathbf{S}^\top\mathbf{R}^\top \\ &= \mathbf{R}\mathbf{S}\underbrace{\mathbf{D}^{-1/2}\mathbf{D}\mathbf{D}^{-1/2}}_{\mathbf{I}}\mathbf{S}^\top\mathbf{R}^\top \\ &= \mathbf{R}\underbrace{\mathbf{S}\mathbf{S}^\top}_{\mathbf{I}}\mathbf{R}^\top \\ &= \mathbf{R}\mathbf{R}^\top \\ &= \mathbf{I}. \end{aligned} \tag{2.26}$$

Thus, by scaling[27] the selected principal components and *then* rotating, we obtain linear combinations that are uncorrelated. Since their expected values are zero and their variances are one, they are still standardized after rotation.

The $k \times p$ matrix of correlations between the original variables and the rotated components is

$$\begin{aligned} corr(\mathbf{z}, \mathbf{f}') = cov(\mathbf{z}, \mathbf{f}') &= cov(\mathbf{z}, \mathbf{R}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{C}^\top\mathbf{z}) \\ &= cov(\mathbf{z})\left(\mathbf{R}\mathbf{S}\mathbf{D}^{-1/2}\mathbf{C}^\top\right)^\top \\ &= \boldsymbol{\Sigma}\,\mathbf{C}\mathbf{D}^{-1/2}\mathbf{S}^\top\mathbf{R}^\top \\ &= \mathbf{C}\mathbf{D}\underbrace{\mathbf{C}^\top\mathbf{C}}_{\mathbf{I}}\mathbf{D}^{-1/2}\mathbf{S}^\top\mathbf{R}^\top \\ &= \mathbf{C}\mathbf{D}\mathbf{D}^{-1/2}\mathbf{S}^\top\mathbf{R}^\top \\ &= \mathbf{C}\mathbf{D}^{1/2}\mathbf{S}^\top\mathbf{R}^\top \\ &= corr(\mathbf{z}, \mathbf{y})\mathbf{S}^\top\mathbf{R}^\top \end{aligned}$$

from (2.25).

Since $corr(\mathbf{z}, \mathbf{y})\mathbf{S}^\top$ is just the first $p$ columns of the $k \times k$ matrix $corr(\mathbf{z}, \mathbf{y})$, we can select principal components first and then compute the correlations, yielding

$$corr(\mathbf{z}, \mathbf{f}') = corr(\mathbf{z}, \mathbf{S}\mathbf{y})\mathbf{R}^\top. \tag{2.27}$$

Furthermore, scaling and rotation does not affect the amount of variance explained by the first $p$ components. By (2.2) and (2.3), the variance in $z_j$ explained by the first

[27] Scaling the components to have variance one is the same as standardizing, because they already have expected value zero.

$p$ components is the sum of the squared correlations between $z_j$ and those components. There are $k$ such quantities, one for each observed variable. They are the diagonal elements of the matrix $corr(\mathbf{z}, \mathbf{Sy})corr(\mathbf{z}, \mathbf{Sy})^\top$.

By (2.27), the corresponding sums of squared correlations between the variables and the scaled and rotated components are on the main diagonal of

$$\begin{aligned} corr(\mathbf{z}, \mathbf{f}')corr(\mathbf{z}, \mathbf{f}')^\top &= corr(\mathbf{z}, \mathbf{Sy})\mathbf{R}^\top \left(corr(\mathbf{z}, \mathbf{Sy})\mathbf{R}^\top\right)^\top \\ &= corr(\mathbf{z}, \mathbf{Sy}) \underbrace{\mathbf{R}^\top \mathbf{R}}_{\mathbf{I}} corr(\mathbf{z}, \mathbf{Sy})^\top \\ &= corr(\mathbf{z}, \mathbf{Sy})corr(\mathbf{z}, \mathbf{Sy})^\top. \end{aligned}$$

That is, for each variable, the sum of squared correlations with the first $p$ original components is the same as the sum of squared correlations with the scaled and rotated components $\mathbf{f}'$.

It remains to show that the sum of squared correlations of the variables with $\mathbf{f}'$ is the variance explained by $\mathbf{f}'$. This is true because

1. Following the calculations leading to (2.3), we have this general result. Let the random variable $w = a_1x_1 + \cdots + a_kx_k$, where $a_, \ldots, a_k$ are non-zero constants, $Var(x_j) = \sigma_j^2$, and $Cov(x_i, x_j) = 0$ for $i \neq j$. Then the variance in $w$ that is explained by a subset of $x$ variables is the sum of their squared correlations with $w$.

2. For $i = 1, \ldots, k$, $z_i = a_{i,1}f'_1 + \cdots + a_{i,p}f'_p + c_{i,p+1}y_{p+1} + \cdots + c_{i,k}y_k$, where $\mathbf{f}' = [f'_j]$. It is a homework problem to write a matrix expression for the $a_{i,j}$.

3. The matrix of covariances between $\mathbf{f}'$ and the principal components $y_{p+1}, \ldots, y_k$ is zero.

The conclusion is that for each variable, the variance explained by the rotated linear combinations $\mathbf{f}'$ is equal to the variance explained by the first $p$ original components.

To summarize, one can select the first $p$ out of $k$ principal components, and then scale them to have variance one. This yields $\mathbf{f}$. Applying a rotation (or reflection) yields $\mathbf{f}' = \mathbf{Rf}$. The random variables in $\mathbf{f}'$ have these properties:

- They are uncorrelated.
- They explain the same amount of variance as the first $p$ principal components.
- Their correlations with the observed variables are equal to the correlations of the first $p$ principal components with the observed variables, but post-multiplied by the transpose of the rotation matrix. This is equation (2.27).

All this holds for any $p \times p$ rotation matrix — that is, for any orthogonal matrix $\mathbf{R}$.

Now, it is not at all mandatory to scale and rotate the principal components, but it can be useful, because the original components, though unique, are often difficult to understand in terms of the input variables. Rotation to something approaching simple

structure can result in linear combinations of the variables that are uncorrelated, collectively just as good as the principal components in terms of explaining variance, and also easy to understand. The only thing that is lost is the property that the first one explains the most possible variance, and so on.

The mechanics of rotation can be directly borrowed from factor analysis. Recalling factor analysis with rotation,

$$\begin{aligned} \mathbf{z} &= \mathbf{\Lambda F} + \mathbf{e} \\ &= (\mathbf{\Lambda R}^\top)(\mathbf{RF}) + \mathbf{e} \\ &= (\mathbf{\Lambda R}^\top)\mathbf{F}' + \mathbf{e}, \end{aligned}$$

where $\mathbf{F}'$ denotes the rotated factors. Based on an initial solution $\widehat{\mathbf{\Lambda}}$, the rotation matrix $\mathbf{R}$ is chosen so that $\widehat{\mathbf{\Lambda}}\mathbf{R}^\top$ has a simple structure.

Comparing Equation (2.27) to the corresponding results for factor analysis,

$$corr(\mathbf{z}, \mathbf{f}') = corr(\mathbf{z}, \mathbf{Sy})\mathbf{R}^\top \quad corr(\mathbf{z}, \mathbf{F}') = corr(\mathbf{z}, \mathbf{F})\mathbf{R}^\top.$$

So, one can simply take the matrix of sample correlations between the variables and the first $p$ principal components, and hand it to a rotation algorithm like varimax. The result will be simplified matrix of correlations between the variables and and a set of rotated components $\mathbf{f}'$ – as well as the rotation matrix that gets the job done.

Illustrating with the Mind-body data, we begin with `pc2`, the earlier `prcomp` object that retained just the two principal components, the ones with eigenvalues greater than one.

```
> pc2 = prcomp(dat, scale = T, rank=2)
> ls(pc2)
[1] "center"   "rotation" "scale"    "sdev"     "x"
```

The list element `pc2$x` is an $n \times 2$ matrix of the two principal components that are retained. Looking at the correlations of these principal components with the variables,

```
> cor(dat,pc2$x)
               PC1        PC2
progmat -0.4709330 -0.6299014
reason  -0.4981509 -0.7277446
verbal  -0.5519561 -0.6910097
headlng -0.7500678  0.1156757
headbrd -0.6073970  0.3689507
headcir -0.9063741  0.1686041
bizyg   -0.8298157  0.2293757
weight  -0.7274347  0.2792455
height  -0.7364050  0.2490749
```

Correlations of raw (unrotated) principal components with variables are always hard to understand, but the minus signs make it worse. We can just flip the signs and everything still correct, because correlations between variables and principal components have the same signs as eigenvector elements. The definition of an eigenvector and corresponding eigenvalue is $\mathbf{Ax} = \lambda\mathbf{x}$. Thus, if $\mathbf{x}$ is an eigenvector corresponding to $\lambda$, so is $-\mathbf{x}$. The choice of sign is arbitrary.

```
> y = - pc2$x # Principal components (reflected, still unrotated)
> M1 = cor(dat,y); M1 # Correlations between variables and components
              PC1        PC2
progmat 0.4709330  0.6299014
reason  0.4981509  0.7277446
verbal  0.5519561  0.6910097
headlng 0.7500678 -0.1156757
headbrd 0.6073970 -0.3689507
headcir 0.9063741 -0.1686041
bizyg   0.8298157 -0.2293757
weight  0.7274347 -0.2792455
height  0.7364050 -0.2490749
```

Applying a rotation to these correlations is very easy.

```
> vmax1 = varimax(M1); print(vmax1, cutoff=0)
$loadings

Loadings:
        PC1    PC2
progmat  0.122  0.777
reason   0.100  0.876
verbal   0.165  0.869
headlng  0.717  0.248
headbrd  0.709 -0.042
headcir  0.880  0.274
bizyg    0.841  0.185
weight   0.774  0.093
height   0.767  0.124

                 PC1   PC2
SS loadings    3.739 2.323
Proportion Var 0.415 0.258
Cumulative Var 0.415 0.674

$rotmat
           [,1]      [,2]
[1,]  0.8841526 0.4671982
[2,] -0.4671982 0.8841526
```

The pattern of correlations is clear. After rotation, the first component represents physical size, and the second component represents performance on the mental tests. The 67.4% of variance explained is the same as the percentage of variance explained before rotation:

```
> sum(M1^2)/9
[1] 0.6735697
```

As a quick cross-check, we calculate the scaled principal components $\mathbf{f}$, apply the rotation from `vmax1` to obtain $\mathbf{f}'$, and verify that $corr(\mathbf{z}, \mathbf{f}')$ corresponds to the "loadings" produced by the `varimax` function. In `fprime = f %*% t(R)`, note the post-multiplication by $\mathbf{R}^\top$, rather than pre-multiplication by $\mathbf{R}$. This is because the $n$ random $\mathbf{f}$ vectors are in the rows of a matrix, and thus are transposed.

```
> f = scale(y)
> # Note that pc2$rotmat is the transpose of the rotation matrix that is applied to the factors
> R = t(vmax1$rotmat) # Transpose it for notation consistent with the text.
> fprime = f %*% t(R)
> round(cor(dat,fprime),3)
         [,1]   [,2]
progmat 0.122  0.777
reason  0.100  0.876
verbal  0.165  0.869
headlng 0.717  0.248
headbrd 0.709 -0.042
headcir 0.880  0.274
bizyg   0.841  0.185
weight  0.774  0.093
height  0.767  0.124
> print(vmax1$loadings,cutoff=0) # For comparison

Loadings:
        PC1    PC2
progmat  0.122  0.777
reason   0.100  0.876
verbal   0.165  0.869
headlng  0.717  0.248
headbrd  0.709 -0.042
headcir  0.880  0.274
bizyg    0.841  0.185
weight   0.774  0.093
height   0.767  0.124

                 PC1   PC2
SS loadings    3.739 2.323
Proportion Var 0.415 0.258
Cumulative Var 0.415 0.674
```

This works so well that I really can't see why anyone would want to do principal components *without* rotation. In fact, rotating principal components is a fairly common practice. Social scientists do it all the time. Many are led down this path by the default "factor analysis" method in SPSS and SAS being principal components (!) and the default rotation method being varimax.

It's interesting what these users do when they obtain a new data set with the same variables, or when they use a set of variables that have previously been "factor analyzed" by another author. Rather than using the weights (eigenvectors) from the first study, they tend to form "scales" by simply adding up the variables that correlate primarily with the same component, or possibly adding up $z$ values if the variables are on really different scales (as the physical variables are in our example). Thus, they would get a "size" variable and a "smart" variable from the Mind-body data. The reasoning is usually not explicit, but I believe they may be thinking that the particular weights may be quite specific to the sub-population from which they obtained the data, and the weights may also be subject to sampling error. They want something more portable and generalizable, so they go with a cruder linear combination. In my view, this may be pretty good practice.

Generally speaking, the more sophisticated the user, the less likely he or she is to apply a rotation to principal components. After all, rotation is a central tool in exploratory factor analysis, and principal components analysis definitely is not factor analysis. So why do it? This little section provides the answer. I hope it establishes that scaling and then rotating a set principal components makes them easier to interpret, without sacrificing anything important.

# Chapter 3

# Confirmatory Factor Analysis

In confirmatory factor analysis, as in exploratory factor analysis, a set of unobservable latent variables called "factors" give rise to a set of observable variables. The principal difference between exploratory and confirmatory factor analysis is in the treatment of parameter identifiability. Exploratory factor analysis models include a link between every factor and every observable variable, and attempt to deal with the resulting lack of identifiability by rotating the factor solutions. Confirmatory factor analysis behaves much more like a traditional statistical method. Based on substantive considerations and re-parameterizations, the dimension of the parameter space is reduced so as to make the parameters identifiable. Then, estimation and inference proceed as usual. Confirmatory factor analysis models are directly imported as the measurement model in the general two-stage model of Chapter 1.

Given a set of data (or proposed set of data), it is generally quite easy to come up with a confirmatory factor analysis model. Such a model may be blessed with identifiability, or it may not. If not, it's back to the drawing board. The primary objective of this chapter is to develop a set of rules that will allow the reader to determine the identifiability status of a model without elaborate calculation – usually by just examining the path diagram. As in Chapters 0 and 1, identifiable almost always means identifiable from the covariance matrix. The rules for parameter identifiability from throughout the book, including this chapter, are collected in Appendix D.

Using the conceptual framework of Chapter 1, underlying everything is a regression-like *original model*. The parameters of the original model will not be identifiable, so it is simplified and re-parameterized to obtain a *surrogate model* whose parameters may be identifiable. The parameters of the surrogate model bear a systematic relationship to the the parameters of the original model, and by keeping track of what that relationship is, it will be possible to draw conclusions about the parameters of the original model. For example, suppose a parameter $\theta_j$ of the surrogate model is a positive multiple of a parameter in the original model. Then if a test determines that $\theta_j > 0$, it can also be concluded that the parameter of the original model is positive.

Here is the original model for confirmatory factor analysis. It is a part of the general two-stage model (1.1). Independently for $i = 1, \ldots, n$, let

$$\mathbf{d}_i = \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \tag{3.1}$$

where

- $\mathbf{d}_i$ is a $k \times 1$ observable random vector. The expected value of $\mathbf{d}_i$ will be denoted by $\boldsymbol{\mu}$, and the covariance matrix of $\mathbf{d}_i$ will be denoted by $\boldsymbol{\Sigma}$.
- $\boldsymbol{\nu}$ is a $k \times 1$ vector of constants.
- $\boldsymbol{\Lambda}$ is a $k \times (p+q)$ matrix of constants.
- $\mathbf{F}_i$ ($F$ for Factor) is a $p \times 1$ latent random vector whose expected value is denoted by $\boldsymbol{\mu}_F$, and whose variance-covariance matrix is denoted by $\boldsymbol{\Phi}$.
- $\mathbf{e}_i$ is a $k \times 1$ vector of error terms that is independent of $\mathbf{F}_i$. It has expected value zero and covariance matrix $\boldsymbol{\Omega}$, which need not be positive definite.

This looks a lot like a multivariate regression model, and it is more or less acceptable for all the reasons that regression is acceptable. It may not be exactly correct, but there is hope that it's a reasonable approximation of the truth, at least within the range of the data.

As discussed in Section A.6.1 of Chapter 1, the parameter vectors $\boldsymbol{\nu}$ and $\boldsymbol{\mu}_F$ will almost never be identifiable separately based on $\boldsymbol{\mu}$, even if it were possible to identify $\boldsymbol{\Lambda}$, $\boldsymbol{\Phi}$ and $\boldsymbol{\Omega}$ from $\boldsymbol{\Sigma}$. Accordingly, we re-parameterize, obtaining a *surrogate centered model*. As a warm-up for what is to come, it is helpful to express the re-parameterization as a change of variables.

$$\begin{aligned}
& \mathbf{d}_i = \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i \\
\iff \quad & \mathbf{d}_i = \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + (\boldsymbol{\Lambda}\boldsymbol{\mu}_F - \boldsymbol{\Lambda}\boldsymbol{\mu}_F) + \mathbf{e}_i \\
\iff \quad & \mathbf{d}_i - (\boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F) = \boldsymbol{\Lambda}(\mathbf{F}_i - \boldsymbol{\mu}_F) + \mathbf{e}_i \\
\iff \quad & (\mathbf{d}_i - \boldsymbol{\mu}) = \boldsymbol{\Lambda}(\mathbf{F}_i - \boldsymbol{\mu}_F) + \mathbf{e}_i \\
\iff \quad & \overset{c}{\mathbf{d}}_i = \boldsymbol{\Lambda}\overset{c}{\mathbf{F}}_i + \mathbf{e}_i,
\end{aligned} \tag{3.2}$$

where the superscript $c$ indicates *centered* versions of the random vectors, in which $\mathbf{d}_i$ and $\mathbf{F}_i$ are expressed as deviations from their expected values. The centering notation is dropped, and the result is a model from which $\boldsymbol{\nu}$ and $\boldsymbol{\mu}_F$ have been eliminated. We are glad to see them go. They are not identifiable separately anyway, and the function of $\boldsymbol{\nu}$ and $\boldsymbol{\mu}_F$ that is identifiable, $\boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F$, is of very little interest. The parameters we really care about are the factor loadings in $\boldsymbol{\Lambda}$ and the correlations between factors in $\boldsymbol{\Phi}$. These quantities are unaffected by centering.

Here is a full statement of the *centered surrogate model*. Independently for $i = 1, \ldots, n$,

$$\mathbf{d}_i = \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \tag{3.3}$$

where

- All expected values are zero.
- $\mathbf{d}_i$ is a $k \times 1$ observable random vector, with $cov(\mathbf{d}_i) = \boldsymbol{\Sigma}$.
- $\boldsymbol{\Lambda}$ is a $k \times (p+q)$ matrix of constants (factor loadings).
- $\mathbf{F}_i$ ($F$ for Factor) is a $p \times 1$ latent random vector with $cov(\mathbf{F}_i) = \boldsymbol{\Phi}$.
- $\mathbf{e}_i$ is a $k \times 1$ random vector of error terms that is independent of $\mathbf{F}_i$. Its covariance matrix is $\boldsymbol{\Omega}$.

In practice, special cases of this model will be fit to data sets where the expected values of the variables are definitely not zero. There are two ways to justify this, equivalent in practice. The first solution is to leave $\mathbf{d}_i$ uncentered in the model, and estimate the nuisance parameters in $\boldsymbol{\mu} = \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F$ with the vector of sample means $\bar{\mathbf{d}}$. The other solution is to center $\mathbf{d}_i$ in the data set, by subtracting off $\bar{\mathbf{d}}$. In either case, inference about $\boldsymbol{\Lambda}$ and $\boldsymbol{\Phi}$ will be based on the sample covariance matrix $\widehat{\boldsymbol{\Sigma}}$.

Readers of Chapter 2 will recognize Model (3.3) as almost identical to the "general factor analysis model" (2.6) on page 213. The only difference is that here, $cov(\mathbf{e}_i) = \boldsymbol{\Omega}$ need not be diagonal, though it is diagonal in many of the simpler models. One could say that, recognizing exploratory factor analysis as a failure, we are starting over.

It was shown in Chapter 2 (especially Sections 2.2 and 2.3) that the parameters of the centered surrogate are not identifiable without some further restrictions on the parameter space. These restrictions are of two kinds. The first kind of restriction is substantive, based on the nature of the data. Setting parameters equal to one another (for example, equal factor loadings) or equal to zero are invariably substantive restrictions, and must be justified in terms of the data set.

The other kind of restriction involves setting certain parameters to the value one. Thinking of the original Model (3.1) as the "true model," this might seem like an arbitrary restriction of the parameter space. However, it will turn out that the resulting model is a surrogate model, in which the centered model (3.3) has been re-parameterized by a change of variables. The parameters of the surrogate model are identifiable *functions* of the original model parameters. By making the process of re-parameterization explicit, we will be able to tell what the surrogate model parameters mean.

Again, the primary objective of this chapter is to build up a set of simple rules for deciding whether the parameters of a proposed model are identifiable. Two important rules have already been established. They are the *Parameter Count Rule* (Rule 1, first stated on page 61) and the *Double Measurement Rule* (Rule 2a, page 178). The parameter count

rule gives a simple necessary condition for identifiability[1], while the double measurement rule, like most of the other standard rules in this book, describes a sufficient condition. The double measurement rule fits neatly into the next section.

## 3.1 Setting Some Factor Loadings to One

In both the original Model (3.1) and the centered surrogate model (3.3), the factor loadings in the matrix $\mathbf{\Lambda}$ are unrestricted. In this section, parameter identifiability will be obtained by setting some factor loadings to one. We will start by just accepting these models as given, focusing on the technical details of identifiability. Then later, it will be shown how these seemingly arbitrary restrictions of the parameter space are actually re-parameterizations that result in surrogate model, one whose parameters have a systematic relationship to the parameters of the original model.

**Double Measurement** Recall the double measurement model (1.22) on page 177, which arose in the course of checking identifiability for the brand awareness data. Figure 3.1 shows a simple scalar example.

Each factor is measured by two observable variables; the factor loadings are all equal to one. There are two sets of measurements, with potentially non-zero covariances within sets, but not between sets. As in the brand awareness Example 1.2 on page 139, common extraneous influences on the measurements within each set are to be expected, but pains have been taken to make the two sets of measurements independent. In general there can be any number of factors, but it becomes challenging to draw the path diagram. Figure 1.10 on page 184 is a try with five factors.

To re-state the double measurement model in matrix form, let

$$\begin{aligned} \mathbf{d}_{i,1} &= \mathbf{F}_i + \mathbf{e}_{i,1} \\ \mathbf{d}_{i,2} &= \mathbf{F}_i + \mathbf{e}_{i,2}, \end{aligned}$$

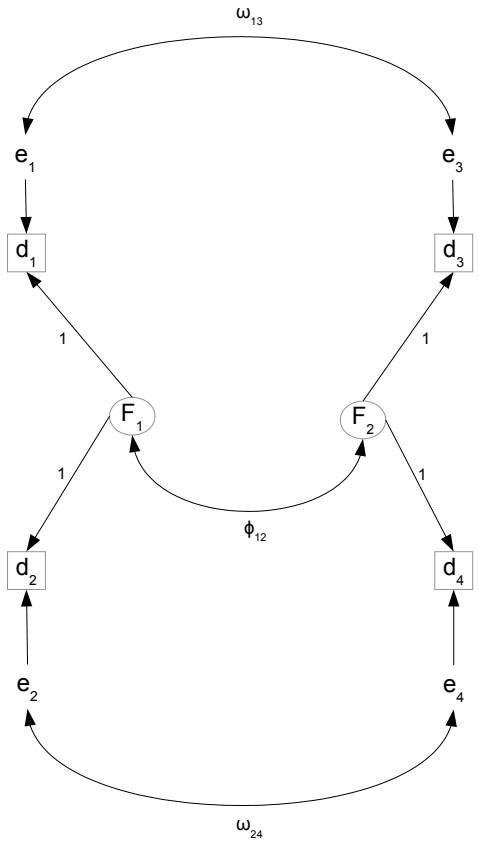
where $E(\mathbf{F}_i) = \mathbf{0}$, $cov(\mathbf{F}_i) = \mathbf{\Phi}$, $\mathbf{F}_i$ has zero covariance with $\mathbf{e}_{i,1}$ and $\mathbf{e}_{i,2}$, $cov(\mathbf{e}_{i,1}) = \mathbf{\Omega}_1$, $cov(\mathbf{e}_{i,2}) = \mathbf{\Omega}_2$ and $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) = \mathbf{O}$.

The parameters in this model (which will be most useful as part of a larger model) are the unique elements of the matrices $\mathbf{\Phi}$, $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$. The double measurement rule (Rule 2a) says that these parameters are identifiable.

**Three observed variables** We now develop an identifiability rule in which for each factor, there are three observable variables of a certain kind. Figure 3.2 shows the path diagram when there is one factor. Here is a statement of the model. Independently for

[1] "Suppose identifiability is to be decided based on a set of moment structure equations. If there are more parameters than equations, the parameter vector is identifiable on at most a set of volume zero in the parameter space."

Figure 3.1: Scalar Double Measurement



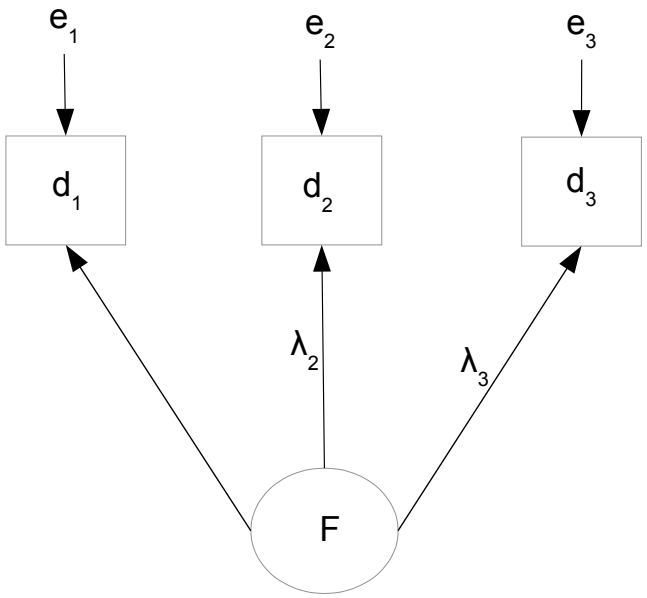

$i = 1, \ldots, n$, let

$$\begin{aligned} d_{i,1} &= F_i + e_{i,1} \\ d_{i,2} &= \lambda_2 F_i + e_{i,2} \\ d_{i,3} &= \lambda_3 F_i + e_{i,3}, \end{aligned} \tag{3.4}$$

with all expected values zero, $Var(F_i) = \phi > 0$, $Var(e_{i,j}) = \omega_j > 0$, and $F_i$ and $e_{i,j}$ all independent. Note that this is a centered model, and that in the first equation, a factor loading that would be denoted $\lambda_1$ has been set to one. Centered variables and parameters equal to one are signs that it's a surrogate model.

The parameter vector is $\boldsymbol{\theta} = (\phi, \lambda_2, \lambda_3, \omega_1, \omega_2, \omega_3)$. There are six unknown parameters, and the covariance matrix of $(d_{i,1}, d_{i,2}, d_{i,3})^\top$ has six unique elements. This means that there are six covariance structure equations in six unknown parameters. If the parameters

Figure 3.2: One Unstandardized Factor, Three Observed Variables




are identifiable, they are just identifiable. Calculating the covariance matrix,

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{pmatrix} = \begin{array}{c|ccc} & d_1 & d_2 & d_3 \\ \hline d_1 & \phi + \omega_1 & \lambda_2\phi & \lambda_3\phi \\ d_2 & & \lambda_2^2\phi + \omega_2 & \lambda_2\lambda_3\phi \\ d_3 & & & \lambda_3^2\phi + \omega_3 \end{array}.$$

The covariance structure equations are

$$\begin{aligned} \sigma_{11} &= \phi + \omega_1 \\ \sigma_{12} &= \lambda_2\phi \\ \sigma_{13} &= \lambda_3\phi \\ \sigma_{22} &= \lambda_2^2\phi + \omega_2 \\ \sigma_{23} &= \lambda_2\lambda_3\phi \\ \sigma_{33} &= \lambda_3^2\phi + \omega_3. \end{aligned}$$

If $\lambda_2 = \lambda_3 = 0$, that fact can be determined from $\sigma_{12} = \sigma_{13} = 0$, so that $\lambda_2$ and $\lambda_3$ are identifiable. The parameters $\omega_2 = \sigma_{22}$ and $\omega_3 = \sigma_{33}$ are also identifiable. However, only the equation $\sigma_{11} = \phi + \omega_1$ remains, and there are infinitely many solutions. This means that at points in the parameter space where $\lambda_2 = \lambda_3 = 0$, only four of the six parameters are identifiable.

Suppose just one of $\lambda_2$ and $\lambda_3$ equals zero, say, $\lambda_2$. In that case, $\lambda_2$ and $\omega_2$ are identifiable, but the equation $\sigma_{23} = 0$ is essentially lost. By the parameter count rule,

the remaining three equations in four unknowns do not have a unique solution, except possibly on a set of volume zero in that four-dimensional section of the parameter space. The conclusion is that the parameter vector is not identifiable at points where $\lambda_2 = 0$, $\lambda_3 = 0$, or both.

So assume that $\lambda_2 \neq 0$ and $\lambda_3 \neq 0$. This "assumption" means that we are considering points in the parameter space where both $\lambda_2$ and $\lambda_3$ are non-zero. In practical situations, it means that the variables $d_2$ and $d_3$ (and $d_1$ too, of course) need to be chosen so that they unquestionably reflect the underlying factor $F$. In this case, the covariance structure equations have the unique solution

$$\begin{aligned} \phi &= \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \\ \lambda_2 &= \sigma_{23}/\sigma_{13} \\ \lambda_3 &= \sigma_{23}/\sigma_{12} \\ \omega_1 &= \sigma_{11} - \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \\ \omega_2 &= \sigma_{22} - \frac{\sigma_{12}\sigma_{23}}{\sigma_{13}} \\ \omega_3 &= \sigma_{33} - \frac{\sigma_{13}\sigma_{23}}{\sigma_{13}}. \end{aligned} \tag{3.5}$$

Suppose we add another observed variable to the model: $d_{i,4} = \lambda_4 F_i + e_{i,4}$. The covariance matrix is now

$$\boldsymbol{\Sigma} = \begin{pmatrix} \phi + \omega_1 & \lambda_2\phi & \lambda_3\phi & \lambda_4\phi \\ & \lambda_2^2\phi + \omega_2 & \lambda_2\lambda_3\phi & \lambda_2\lambda_4\phi \\ & & \lambda_3^2\phi + \omega_3 & \lambda_3\lambda_4\phi \\ & & & \lambda_4^2\phi + \omega_4 \end{pmatrix}.$$

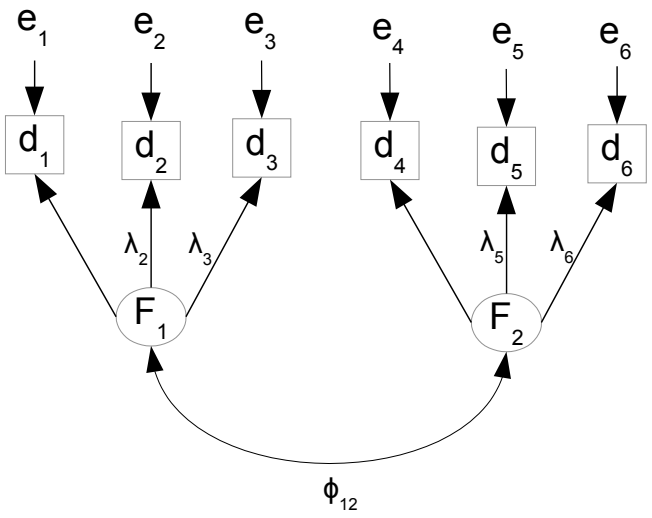
Whether or not $\lambda_4 = 0$, all the parameters are easily identifiable. For five observed variables, two loadings can be zero, and so on.

With more than three observed variables, the parameters are over-identified. In this case, testing model fit is a possibility. For example, if there are four observed variables, then there are eight parameters and ten covariance structure equations, giving rise to $10 - 8 = 2$ equality constraints on the covariance matrix.

Now add another factor to Model (3.4), as in Figure 3.3. A single factor loading has been set to one for each factor, $cov(\mathbf{F}_i) = \boldsymbol{\Phi} = [\phi_{\ell j}]$, and $Var(e_j) = \omega_j$ for $j = 1, \ldots, 6$. The model equations are

$$\begin{aligned} d_1 &= F_1 + e_1 \\ d_2 &= \lambda_2 F_1 + e_2 \\ d_3 &= \lambda_3 F_1 + e_3 \\ d_4 &= F_2 + e_4 \\ d_5 &= \lambda_5 F_2 + e_5 \\ d_6 &= \lambda_6 F_2 + e_6, \end{aligned}$$

Figure 3.3: Two Unstandardized Factors




and the covariance matrix of the observable variables is

$$\mathbf{\Sigma} = \begin{pmatrix} \omega_1 + \phi_{11} & \lambda_2\phi_{11} & \lambda_3\phi_{11} & \phi_{12} & \lambda_5\phi_{12} & \lambda_6\phi_{12} \\ & \lambda_2^2\phi_{11} + \omega_2 & \lambda_2\lambda_3\phi_{11} & \lambda_2\phi_{12} & \lambda_2\lambda_5\phi_{12} & \lambda_2\lambda_6\phi_{12} \\ & & \lambda_3^2\phi_{11} + \omega_3 & \lambda_3\phi_{12} & \lambda_3\lambda_5\phi_{12} & \lambda_3\lambda_6\phi_{12} \\ & & & \omega_4 + \phi_{22} & \lambda_5\phi_{22} & \lambda_6\phi_{22} \\ & & & & \lambda_5^2\phi_{22} + \omega_5 & \lambda_5\lambda_6\phi_{22} \\ & & & & & \lambda_6^2\phi_{22} + \omega_6 \end{pmatrix}$$

Typesetting that covariance matrix would have been a chore. `SageMath` kindly agreed to do it for me; then I manually removed the lower triangle to make the matrix easier to look at. Here is the code.

```
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
# Two unstandardized factors
L = ZeroMatrix(6,2)
L[0,0]= 1; L[1,0]= var('lambda2'); L[2,0]= var('lambda3')
L[3,1]= 1; L[4,1]= var('lambda5'); L[5,1]= var('lambda6'); L
P = SymmetricMatrix(2,'phi'); P
O = DiagonalMatrix(6,symbol='omega'); O
Sig = FactorAnalysisCov(L,P,O); Sig
print(latex(Sig))
```

Assuming $\lambda_2$, $\lambda_3$, $\lambda_5$ and $\lambda_6$ to be non-zero, these factor loadings along with $\phi_{11}, \phi_{22}$ and $\omega_1, \ldots \omega_6$ may be recovered as for the one-factor model. The remaining parameter, $\phi_{12}$, is identified from $\phi_{12} = \sigma_{14}$. Thus, all the parameters are identifiable. Identifiability

is preserved when more factors are added under these same conditions. Adding more variables in any set also does no harm.

**Reference variables** We are at the point of stating an important general rule, but first, please notice a special feature of the observed variables in the models we have been considering. Each observed variables is influenced by only one factor and an error term. This is almost never seen in exploratory factor analysis, except that it might be considered an extreme case of Thurstone's "simple structure." In confirmatory factor analysis models, such variables are quite common, and it helps to have a name for them. The term is taken from Jöreskog's (1969) classic article in *Psychmetrika* [36].

**Definition 3.1** *A* reference variable *for a latent variable is an observable variable that is a function only of that latent variable and an error term. The factor loading is non-zero.*

Obviously, not all observable variables are reference variables by this definition. For example, in the two-factor model of Figure 2.1 on page 214, there are no reference variables at all. In the latent variable regression model of Figure 13 on page 49, $W_1$ and $W_2$ are reference variables, but $Y$ is not. Reference variable are very useful for establishing identifiability, and many of the standard sufficient conditions for parameter identifiability involve reference variables for the latent variables.

Before the introduction of reference variables, the following rule was established. If the conditions seem overly restrictive, I agree. We can and will do better.

**Three-variable Rule for Unstandardized Factors** The parameters of a factor analysis model are identifiable provided

- There are at least three reference variables for each factor.
- For each factor, the factor loading of at least one reference variable is equal to one.
- Errors are independent of one another and of the factors.

**Only one reference variable per factor is really needed.** The three-variable rule is widely used in practice, but it is more restrictive than it needs to be. It is a lot to ask that each factor have *three* observed variables that are influenced by that factor and none of the others. It's tough enough to come up with one such pure measurement for each factor. Fortunately, it turns out that only one of the three observed variables for each factor needs to be a reference variable. The other two can be influenced by all the factors.

The reference variable rule is a matrix version of the three-variable rule. For it to apply, there must be at least three observable variables for every factor, including one reference variable per factor. The observable variables are collected into three or possibly four vectors. For case $i$ (there are $n$ cases), $\mathbf{d}_{i,1}$ contains reference variables for the factors, with the factor loadings for the reference variables set to one. The number of variables in $\mathbf{d}_{i,2}$ and $\mathbf{d}_{i,3}$ is also equal to the number of factors. If there are any more observable variables, they are placed in $\mathbf{d}_{i,4}$.

Here is the model. Independently for $i = 1, \ldots, n$,

$$\begin{array}{rcl} \mathbf{d}_{i,1} & = & \mathbf{F}_i + \mathbf{e}_{i,1} \\ \mathbf{d}_{i,2} & = & \boldsymbol{\Lambda}_2\mathbf{F}_i + \mathbf{e}_{i,2} \\ \mathbf{d}_{i,3} & = & \boldsymbol{\Lambda}_3\mathbf{F}_i + \mathbf{e}_{i,3} \\ \mathbf{d}_{i,4} & = & \boldsymbol{\Lambda}_4\mathbf{F}_i + \mathbf{e}_{i,4}, \end{array} \tag{3.6}$$

where

- $\mathbf{d}_{i,1}$, $\mathbf{d}_{i,2}$ and $\mathbf{d}_{i,3}$ are $p \times 1$ observable random vectors. If $\mathbf{d}_{i,4}$ is present, it is an $m \times 1$ observable random vector.
- $\mathbf{F}_i$ ($F$ for Factor) is a $p \times 1$ latent random vector with expected value zero $cov(\mathbf{F}_i) = \boldsymbol{\Phi}$.
- $\boldsymbol{\Lambda}_2$ and $\boldsymbol{\Lambda}_3$ are $p \times p$ non-singular matrices of constants.
- $\boldsymbol{\Lambda}_4$, if it is present, is an $m$ by $p$ matrix of constants.
- $\mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,4}$ are vectors of error terms, with expected value zero, covariance matrix $cov(\mathbf{e}_{i,j}) = \boldsymbol{\Omega}_{j,j}$ for $j = 1, \ldots, 4$, and
  - $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) = cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,3}) = cov(\mathbf{e}_{i,2}, \mathbf{e}_{i,3}) = \mathbf{O}$, all $p \times p$ matrices.
  - $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,4}) = \mathbf{O}$, a $p \times m$ matrix.
  - $cov(\mathbf{e}_{i,2}, \mathbf{e}_{i,4}) = \boldsymbol{\Omega}_{2,4}$ and $cov(\mathbf{e}_{i,3}, \mathbf{e}_{i,4}) = \boldsymbol{\Omega}_{3,4}$.

The parameters of this model are the unique elements of the matrices $\boldsymbol{\Phi}$, $\boldsymbol{\Lambda}_2$, $\boldsymbol{\Lambda}_3$, $\boldsymbol{\Omega}_{1,1}$, $\boldsymbol{\Omega}_{2,2}$ and $\boldsymbol{\Omega}_{3,3}$. If there are more than $3p$ observable variables and $\mathbf{d}_{i,4}$ is necessary, the list of parameter matrices also includes $\boldsymbol{\Lambda}_4$, $\boldsymbol{\Omega}_{2,4}$, $\boldsymbol{\Omega}_{3,4}$ and $\boldsymbol{\Omega}_{4,4}$.

Detailed discussion of this model is deferred until Section 3.4. For now, just note that while the three-variable rule allows observable variables to be influenced by only a single factor, Model (3.6) says that at least two-thirds of the variables can be influenced by all the factors, through the matrices $\boldsymbol{\Lambda}_2$ and $\boldsymbol{\Lambda}_3$ (and possibly $\boldsymbol{\Lambda}_4$). Also, the three-variable rule requires all error terms to have zero covariance. In Model (3.6), however, the first $3p$ variables are divided into sets; error terms are allowed to have non-zero covariance within sets, but not between sets. If there is a fourth set of variables, its set of error terms may be correlated with the error terms of sets two and three, as well as with each other — but not with the error terms of set one.

**Identifiability** The covariance matrix of the observable variables may be written as a partitioned matrix.

$$\boldsymbol{\Sigma} = cov\begin{pmatrix} \mathbf{d}_{i,1} \\ \hline \mathbf{d}_{i,2} \\ \hline \mathbf{d}_{i,3} \\ \hline \mathbf{d}_{i,4} \end{pmatrix} = \left(\begin{array}{c|c|c|c} \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} & \boldsymbol{\Sigma}_{1,3} & \boldsymbol{\Sigma}_{1,4} \\ \hline & \boldsymbol{\Sigma}_{2,2} & \boldsymbol{\Sigma}_{2,3} & \boldsymbol{\Sigma}_{2,4} \\ \hline & & \boldsymbol{\Sigma}_{3,3} & \boldsymbol{\Sigma}_{3,4} \\ \hline & & & \boldsymbol{\Sigma}_{4,4} \end{array}\right) \tag{3.7}$$

$$= \left(\begin{array}{c|c|c|c} \boldsymbol{\Phi} + \boldsymbol{\Omega}_{1,1} & \boldsymbol{\Phi}\boldsymbol{\Lambda}_2^\top & \boldsymbol{\Phi}\boldsymbol{\Lambda}_3^\top & \boldsymbol{\Phi}\boldsymbol{\Lambda}_4^\top \\ \hline & \boldsymbol{\Lambda}_2\boldsymbol{\Phi}\boldsymbol{\Lambda}_2^\top + \boldsymbol{\Omega}_{2,2} & \boldsymbol{\Lambda}_2\boldsymbol{\Phi}\boldsymbol{\Lambda}_3^\top & \boldsymbol{\Lambda}_2\boldsymbol{\Phi}\boldsymbol{\Lambda}_4^\top + \boldsymbol{\Omega}_{2,4} \\ \hline & & \boldsymbol{\Lambda}_3\boldsymbol{\Phi}\boldsymbol{\Lambda}_3^\top + \boldsymbol{\Omega}_{3,3} & \boldsymbol{\Lambda}_3\boldsymbol{\Phi}\boldsymbol{\Lambda}_4^\top + \boldsymbol{\Omega}_{3,4} \\ \hline & & & \boldsymbol{\Lambda}_4\boldsymbol{\Phi}\boldsymbol{\Lambda}_4^\top + \boldsymbol{\Omega}_{4,4} \end{array}\right).$$

Viewing (3.7) as a compact way to express the covariance structure equations, one obtains solutions that are directly analogous to (3.5). To avoid transpose signs, the solutions use $\boldsymbol{\Sigma}_{i,j}^\top = \boldsymbol{\Sigma}_{j,i}$.

$$\begin{aligned} \boldsymbol{\Phi} &= \boldsymbol{\Sigma}_{1,3}\boldsymbol{\Sigma}_{2,3}^{-1}\boldsymbol{\Sigma}_{2,1} \\ \boldsymbol{\Lambda}_2 &= \boldsymbol{\Sigma}_{2,3}\boldsymbol{\Sigma}_{1,3}^{-1} \\ \boldsymbol{\Lambda}_3 &= \boldsymbol{\Sigma}_{3,2}\boldsymbol{\Sigma}_{1,2}^{-1} \\ \boldsymbol{\Omega}_{1,1} &= \boldsymbol{\Sigma}_{1,1} - \boldsymbol{\Sigma}_{1,3}\boldsymbol{\Sigma}_{2,3}^{-1}\boldsymbol{\Sigma}_{2,1} \\ \boldsymbol{\Omega}_{2,2} &= \boldsymbol{\Sigma}_{2,2} - \boldsymbol{\Sigma}_{2,1}\boldsymbol{\Sigma}_{3,1}^{-1}\boldsymbol{\Sigma}_{3,2} \\ \boldsymbol{\Omega}_{3,3} &= \boldsymbol{\Sigma}_{3,3} - \boldsymbol{\Sigma}_{3,2}\boldsymbol{\Sigma}_{1,2}^{-1}\boldsymbol{\Sigma}_{1,3}. \end{aligned} \tag{3.8}$$

In case there are more than $3p$ observed variables and $\mathbf{d}_{i,4}$ is needed, solutions for the additional parameter matrices are

$$\begin{aligned} \boldsymbol{\Lambda}_4 &= \boldsymbol{\Sigma}_{4,1}\boldsymbol{\Sigma}_{2,1}^{-1}\boldsymbol{\Sigma}_{2,3}\boldsymbol{\Sigma}_{1,3}^{-1} \\ \boldsymbol{\Omega}_{2,4} &= \boldsymbol{\Sigma}_{2,4} - \boldsymbol{\Sigma}_{2,3}\boldsymbol{\Sigma}_{1,3}^{-1}\boldsymbol{\Sigma}_{1,4} \\ \boldsymbol{\Omega}_{3,4} &= \boldsymbol{\Sigma}_{3,4} - \boldsymbol{\Sigma}_{3,2}\boldsymbol{\Sigma}_{1,2}^{-1}\boldsymbol{\Sigma}_{1,4} \\ \boldsymbol{\Omega}_{4,4} &= \boldsymbol{\Sigma}_{4,4} - \boldsymbol{\Sigma}_{4,1}\boldsymbol{\Sigma}_{3,1}^{-1}\boldsymbol{\Sigma}_{3,2}\boldsymbol{\Sigma}_{1,2}^{-1}\boldsymbol{\Sigma}_{1,4}. \end{aligned} \tag{3.9}$$

This establishes identifiability of all the parameters, except on that set of volume zero in the parameter space where $\boldsymbol{\Lambda}_2$ and $\boldsymbol{\Lambda}_3$ do not have inverses. This is like the requirement that $\lambda_2$ and $\lambda_3$ be non-zero in the three-variable model (1usfactor), so that one may "divide" by them. It's a set of volume zero because if $\boldsymbol{\Lambda}_2$ or $\boldsymbol{\Lambda}_3$ were singular, then the columns would be linearly dependent, and at least one column would be a perfect linear combination of the others.

We have the following important rule. Full discussion will be deferred until Section 3.4, where an even stronger version will be given.

**The Reference Variable Rule for Unstandardized Factors** The parameters of a factor analysis model are identifiable except possibly on a set of volume zero in the parameter space, provided

- The number of observable variables (including reference variables) is at least three times the number of factors.
- For each factor, there is at least one reference variable, with a factor loading of one.
- Divide the observable variables into sets. The first set contains one reference variable for each factor; the factor loadings all equal one. The number of variables in the second set and the number in the third set is also equal to the number of factors. The fourth set may contain any number of additional variables, including zero. The error terms for the variables in the first three sets may have non-zero covariance within sets, but not between sets. The error terms for the variables in the fourth set may have non-zero covariance within the set, and with the error terms of sets two and three, but they must have zero covariance with the error terms of the reference variables.

**Two reference variables per factor** In some models, a factor may influence fewer than three observable variables, a condition that would force either $\mathbf{\Lambda}_2$ or $\mathbf{\Lambda}_3$ to be singular in the preceding discussion. If the model has at least two such factors and non-zero covariance between the factors, we can get away with two reference variables for each factor. Understanding that this may be only part of a larger model, the model equations would be

$$\begin{aligned} d_1 &= F_1 + e_1 \\ d_2 &= \lambda_2 F_1 + e_2 \\ d_3 &= F_2 + e_3 \\ d_4 &= \lambda_4 F_2 + e_4, \end{aligned}$$

with all expected values zero, $cov \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \mathbf{\Phi} = [\phi_{ij}]$, $Var(e_j) = \omega_j$, and the error terms independent of the factors and each other. An additional critical stipulation is that $Cov(F_1, F_2) = \phi_{12} \neq 0$.

The covariance matrix of the observable variables is

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ & & \sigma_{33} & \sigma_{34} \\ & & & \sigma_{44} \end{pmatrix} = \begin{pmatrix} \phi_{11} + \omega_1 & \lambda_2\phi_{11} & \phi_{12} & \lambda_4\phi_{12} \\ & \lambda_2^2\phi_{11} + \omega_2 & \lambda_2\phi_{12} & \lambda_2\lambda_4\phi_{12} \\ & & \phi_{22} + \omega_3 & \lambda_4\phi_{22} \\ & & & \lambda_4^2\phi_{22} + \omega_4 \end{pmatrix} \tag{3.10}$$

Provided $\phi_{12} \neq 0$, all the parameters are easily identifiable. If $\phi_{12} = 0$, then that parameter is identifiable, but then identifying the other parameters would require a unique solution to six equations in eight unknowns. By the parameter count rule, this is impossible in most of the parameter space. Thus, identifiability requires $\phi_{12} \neq 0$.

With more factors and two observed variables per factor, identifiability is maintained provided that each factor has a non-zero covariance with at least one other factor. Naturally, three or more variables for some of the factors is okay. We have the following rule.

**Two-varisble Rule for Unstandardized Factors** The parameters of a factor analysis model are identifiable provided

- There are at least two factors.
- There are at least two reference variables per factor.
- For each factor, the factor loading of at least one reference variable is equal to one.
- Each factor has a non-zero covariance with at least one other factor.
- Errors are independent of one another and of the factors.

**Re-parameterization and surrogate models** Setting some of the factor loadings to one is a useful technical device for obtaining identifiability, but does the resulting model make sense? When a factor loading is set to one, it means that the observed variable is just the factor plus a piece of random noise. Models like this were common in Chapter 0, but the regression-like "original" model with a slope possibly not equal to unity is much easier to believe. While it may be true that "all models are wrong[2]," it is still not a good idea adopt models that are obviously unrealistic, unless there is a good reason.

A common justification for setting factor loadings to one to to describe the process as "setting the scale," as in Bollen [10] (p. 198). Suppose the latent variable is length, and it is measured twice. One measurement is in inches, and the other is in centimeters. What's the scale of measurement of the latent variable? This is un-knowable[3] and not very interesting anyway, so the scale of the latent variable is arbitrarily made to agree with one of the observed variables, by setting its factor loading to one.

As I see it, this "setting the scale" interpretation does not really hold up. Suppose the latent variable is amount of debt, measured in dollars. One of the observed variables is reported debt, also in dollars. Clearly, the latent and observed variables are on the same scale. I think the factor loading could easily be a constant strictly less than one, so that, for example, for every one dollar increase in true debt, the average person might report 75 cents. Setting the factor loading to one when it is really 0.75 would be to model an interesting phenomenon out of existence. There must be some other explanation for setting a factor loading to one.

To see what is really going on, consider the following one-factor example. None of the factor loadings is necessarily equal to one.

**Example 3.1.1** *A centered model with one factor.*

[2]The famous quote is from Box and Draper (1987, p. 424) who said, "Essentially all models are wrong, but some are useful." [11]

[3]A symptom of non-identifiability.

Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} d_{i,1} &= \lambda_1 F_i + e_{i,1} \\ d_{i,2} &= \lambda_2 F_i + e_{i,2} \\ d_{i,3} &= \lambda_3 F_i + e_{i,3} \\ d_{i,4} &= \lambda_4 F_i + e_{i,4}, \end{aligned}$$

with all expected values zero, $Var(F_i) = \phi$, $Var(e_{i,j}) = \omega_j$, and $F_i$ and $e_{i,j}$ all independent.

As usual, identifiability is to be established by solving the covariance structure equations for the unknown parameters. There are nine unknown parameters, and the covariance matrix of the observable variables has ten unique variances and covariances. The parameter count rule says that identifiability is possible, but not guaranteed. The covariance matrix is

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ & & \sigma_{33} & \sigma_{34} \\ & & & \sigma_{44} \end{pmatrix} = \begin{pmatrix} \lambda_1^2\phi + \omega_1 & \lambda_1\lambda_2\phi & \lambda_1\lambda_3\phi & \lambda_1\lambda_4\phi \\ & \lambda_2^2\phi + \omega_2 & \lambda_2\lambda_3\phi & \lambda_2\lambda_4\phi \\ & & \lambda_3^2\phi + \omega_3 & \lambda_3\lambda_4\phi \\ & & & \lambda_4^2\phi + \omega_4 \end{pmatrix} \tag{3.11}$$

If two distinct parameter sets yield the same covariance matrix, the parameter vector is not identifiable. Table 3.1 shows two such parameter sets — actually, infinitely many.

Table 3.1: Non-identifiability

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{\theta}_1$ | $\phi$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ |
| $\boldsymbol{\theta}_2$ | $\phi/c^2$ | $c\lambda_1$ | $c\lambda_2$ | $c\lambda_3$ | $c\lambda_4$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ |

For any $c \neq 0$, both $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ yield the covariance matrix in (3.11).

As usual when parameters are not identifiable, this is a serious problem. Regardless of what the true parameter values are, there are infinitely many sets of *untrue* parameter values that yield exactly the same $\boldsymbol{\Sigma}$. Since inference is based on the covariance matrix of the observable data, there is no way to even approach the full truth based on the data, no matter how large the sample size. However, there is a way to get at the partial truth, because certain *functions* of the parameter vector are identifiable. For example, at points in the parameter space where $\lambda_1, \lambda_2, \lambda_3 \neq 0$,

- $\sigma_{11} - \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} = \omega_1$, and the other error variances are identifiable by a similar calculation.
- $\frac{\sigma_{13}}{\sigma_{23}} = \frac{\lambda_1\lambda_3\phi}{\lambda_2\lambda_3\phi} = \frac{\lambda_1}{\lambda_2}$, so *ratios* of factor loadings are identifiable.

- If the sign of one factor loading is known (say by naming the factor[4]), then the signs of the others can be identified from the covariances in (3.11).
- $\frac{\sigma_{12}\sigma_{13}}{\sigma_{23}\sigma_{11}} = \frac{\lambda_1^2\phi}{\lambda_1^2\phi+\omega_1}$, the reliability of $d_1$ as a measure of $F$. Reliabilities are identifiable for this model.

The point here is that while the entire parameter vector may not be identifiable, the covariance matrix still contains useful information about the parameters. It's difficult to get at, though. If we try to fit a model like the one in Example (3.1) by maximum likelihood, lack of identifiability will cause the likelihood function to have a maximum that is not unique, and unpleasant numerical things will happen.

The solution is re-parameterization. It cannot be a one-to-one re-parametrization, because that would leave the identifiability of the model parameters unchanged. Instead, it's a sort of collapsing re-parameterization, one that results in a parameter space of lower dimension. It is accomplished by a change of variables, and the resulting model is a surrogate model. We have already seen how a change of variables is used to transform the original model (3.1) into the centered surrogate model (3.3).

In the model of Example 3.1, assume $\lambda_1 \neq 0$. It can be made positive by naming the factor appropriately, so let $\lambda_1 > 0$. Setting $F' = \lambda_1 F$, we have $d_1 = F' + e_1$, and it *looks* as if $\lambda_1$ has been set to one (if you ignore the prime). The consequences for the other factor loadings are, for example,

$$\begin{aligned} d_2 &= \lambda_2 F + e_2 \\ &= \left(\frac{\lambda_2}{\lambda_1}\right)(\lambda_1 F) + e_2 \\ &= \lambda_2' F' + e_2, \end{aligned}$$

and we have

$$\begin{aligned} d_1 &= F' + e_1 \\ d_2 &= \lambda_2' F' + e_2 \\ d_3 &= \lambda_3' F' + e_3, \text{ etc.} \end{aligned}$$

Losing the primes, the result looks exactly like Model (3.4). It is a surrogate for the model of Example 3.1, except that there are three factors instead of four. In terms of the original model parameters, the parameter $\lambda_2$ is really $\lambda_2/\lambda_1$. The variance parameter $\phi$ is really $\lambda_1^2\phi$. As shown in Table 3.2, these are identifiable functions of the parameters of the original model.

[4]Suppose that the factor is left-right political orientation. Do extremely high scores reflect right-wing ideology, or left-wing ideology? Nobody knows. However, you have an observed variable, score on a questionnaire asking about politics. It is *scored* so that agreement with certain statements gets you a higher Left score or a higher Right score. Which one? It's up to the investigator. So just make a choice, and assume that the factor loading is positive. This way, you have decided whether to call the factor "Left-wing orientation" or "Right-wing orientation." This always works, but you only want to do it when the connection between the factor and the observable variable is completely clear and non-controversial.

Table 3.2: Identifiable Functions in Model (3.4)

| Function of $\boldsymbol{\Sigma}$ | Value under model | |
|---|---|---|
| | Surrogate | Original |
| $\sigma_{23}/\sigma_{13}$ | $\lambda_2$ | $\lambda_2/\lambda_1$ |
| $\sigma_{23}/\sigma_{12}$ | $\lambda_3$ | $\lambda_3/\lambda_1$ |
| $\sigma_{12}\sigma_{13}/\sigma_{23}$ | $\phi$ | $\lambda_1^2\phi$ |

Suppose there is more than one factor, with a factor loading set to one for each factor. Then

$$\begin{aligned} \phi'_{12} &= Cov(F'_1, F'_2) \\ &= Cov(\lambda_1 F_1, \lambda_4 F_2) \\ &= \lambda_1\lambda_4 Cov(F_1, F_2) \\ &= \lambda_1\lambda_4\phi_{12}. \end{aligned}$$

To summarize, setting a factor loading to one for each factor is not an arbitrary restriction of the parameter space. It is a very useful re-parameterization, resulting in a surrogate model. The parameters of the surrogate model are identifiable functions of the original model parameters. Their meanings are limited but clear. Everything is relative to the values of the parameters that have apparently been suppressed. The error variances $\omega_j$ are unaffected, but all the other parameters are positive multiples of the corresponding parameters of the original model. Any estimated factor loading or covariance is really an estimate of that quantity times an unknown positive constant. If the latent variable model has a causal structure (rather than just covariances between factors), the re-parameterization has cascading effects that run down the chain of causality.

Unless one is actually interested in ratios of factor loadings, point and interval estimates of the surrogate model parameters are not very meaningful. However, a test of whether a surrogate model parameter is positive, negative or zero is also a valid test about the original model parameter. This is good enough for many applications. In the social and biological sciences, the primary research question is often whether a relationship between variables exists, and if so, whether the relationship is positive or negative. In such cases, setting factor loadings to one can be an excellent way to achieve parameter identifiability and get on with the data analysis.

## 3.2 Standardized Factors

Setting a factor loading to one for each factor is path to identifiability. The other common trick is to set the variances of the factors to one. Please consider again the one-factor model of Example 3.1 on page 288. Recall that this model is centered, but otherwise it has not been re-parameterized, and its parameters are not identifiable. To obtain $Var(F') = 1$

in a re-parameterized model, let $F' = \frac{1}{\phi^{1/2}}F$. Then $Var(F') = \frac{1}{\phi}Var(F) = 1$ as desired, and

$$\begin{aligned} d_j &= \lambda_j F + e_j \\ &= \lambda_j \phi^{1/2} \frac{1}{\phi^{1/2}} F + e_j \\ &= \lambda'_j F' + e_j. \end{aligned}$$

Under the new, re-parameterized model, the factor loading is expressed as a multiple of the unknown standard deviation of the factor; $\lambda'_j = \lambda_j \phi^{1/2}$ is the expected increase in $d_j$ when $F$ is increased by one standard deviation unit. Since the standard deviation is unknown (and un-knowable) except for being positive, this means that an estimate of $\lambda'_j$ could be informative about the sign of the original factor loading, but that's all.

Discarding the primes, we have a surrogate model. Consider the following three-variable version. Independently for $i = 1, \ldots, n$, let

$$\begin{aligned} d_{i,1} &= \lambda_1 F_i + e_{i,1} \\ d_{i,2} &= \lambda_2 F_i + e_{i,2} \\ d_{i,3} &= \lambda_3 F_i + e_{i,3}, \end{aligned}$$

with all expected values zero, $Var(F_i) = 1$, $Var(e_{i,j}) = \omega_j$ and $F_i$ and $e_{i,j}$ all independent.

The covariance matrix of an observable data vector is,

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{pmatrix} = \begin{pmatrix} \lambda_1^2 + \omega_1 & \lambda_1\lambda_2 & \lambda_1\lambda_3 \\ & \lambda_2^2 + \omega_2 & \lambda_2\lambda_3 \\ & & \lambda_3^2 + \omega_3 \end{pmatrix}. \tag{3.12}$$

There are six covariance structure equations in six unknown parameters. If two or three of the factor loadings are equal to zero, all three covariances equal zero, it's impossible to tell whether two loadings or three equal zero, and none of the parameters is identifiable. So, consider what happens when just one factor loading equals zero – say, $\lambda_1$. Since $\sigma_{12} = \sigma_{13} = 0$ but $\sigma_{23} \neq 0$, it is clear that $\lambda_1 = 0$. That is, its value is identifiable. Also, $\sigma_{11} = \omega_1$, and $\omega_1$ is identifiable. However, the covariance structure equation $\sigma_{23} = \lambda_2\lambda_3$ has infinitely many solutions; identification of $\omega_2$ and $\omega_3$ is also impossible.

Accordingly, assume that $\lambda_1$, $\lambda_2$ and $\lambda_3$ are all non-zero. As in Section 1.3 of Chapter 1, this is an acknowledgement that parameter identifiability need not be the same in different regions of the parameter space. Viewing (3.12) as a compact statement of the covariance

structure equations and trying to solve, we have

$$\begin{aligned}
\lambda_1^2 &= \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} = \frac{\lambda_1\lambda_2\,\lambda_1\lambda_3}{\lambda_2\lambda_3} \\
\lambda_2^2 &= \frac{\sigma_{12}\sigma_{23}}{\sigma_{13}} \\
\lambda_3^2 &= \frac{\sigma_{13}\sigma_{23}}{\sigma_{12}} \\
\omega_1 &= \sigma_{11} - \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \\
\omega_2 &= \sigma_{22} - \frac{\sigma_{12}\sigma_{23}}{\sigma_{13}} \\
\omega_3 &= \sigma_{33} - \frac{\sigma_{13}\sigma_{23}}{\sigma_{12}}
\end{aligned} \tag{3.13}$$

The error variances are identifiable, but only the squares of the factor loadings can be uniquely identified. To see this clearly, note that if all three $\lambda_j$ are replaced with $-\lambda_j$, we get same $\mathbf{\Sigma}$. The likelihood function will have two maxima, of the same height. Which one is located will depend on where the numerical search starts.

The solution is to decide on the sign of one factor loading. It really is a decision that is up to the user, and it's based on the *meaning* of the hypothesized factor. If the three variables are scores on three math tests, is $F$ math ability, or math inability? You decide. Once the sign of one loading is fixed, the signs of the other two may be determined from the signs of the $\sigma_{ij}$. Identifiability is purchased by cutting the parameter space in half, but it really doesn't cost anything.

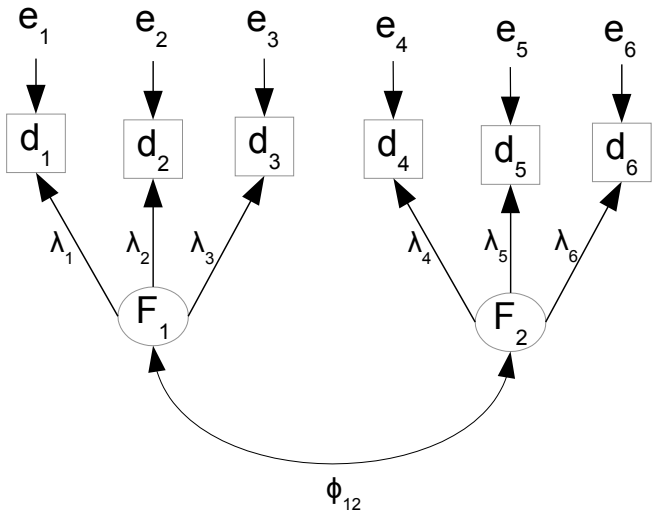
Now suppose we add another observed variable to the model: $d_{i,4} = \lambda_4 F_i + e_{i,4}$. The covariance matrix is

$$\mathbf{\Sigma} = \begin{pmatrix}
\lambda_1^2+\omega_1 & \lambda_1\lambda_2 & \lambda_1\lambda_3 & \lambda_1\lambda_4 \\
 & \lambda_2^2+\omega_2 & \lambda_2\lambda_3 & \lambda_2\lambda_4 \\
 & & \lambda_3^2+\omega_3 & \lambda_3\lambda_4 \\
 & & & \lambda_4^2+\omega_4
\end{pmatrix}. \tag{3.14}$$

The parameters will all be identifiable as long as three out of four loadings are non-zero, and one sign is known. For example, if only $\lambda_1 = 0$ then the top row $= 0$, and it is possible to solve for $\lambda_2, \lambda_3, \lambda_4$ as before. For five observed variables, two loadings can be zero, and so on. With more than three observed variables, the parameters are over-identified. For example, with four observed variables, there are eight parameters and ten covariance structure equations, giving rise to $10 - 8 = 2$ equality constraints on the covariance matrix.

Returning to three observed variables per factor, add another factor as in Figure 3.4. The variances of both factors equal one, and $Var(e_j) = \omega_j$ for $j = 1, \ldots, 6$. The model

Figure 3.4: Two factors




equations are

$$\begin{aligned} d_1 &= \lambda_1 F_1 + e_1 \\ d_2 &= \lambda_2 F_1 + e_2 \\ d_3 &= \lambda_3 F_1 + e_3 \\ d_4 &= \lambda_4 F_2 + e_4 \\ d_5 &= \lambda_5 F_2 + e_5 \\ d_6 &= \lambda_6 F_2 + e_6, \end{aligned}$$

and the covariance matrix of the observable variables is

$$\mathbf{\Sigma} = \begin{pmatrix} \lambda_1^2 + \omega_1 & \lambda_1\lambda_2 & \lambda_1\lambda_3 & \lambda_1\lambda_4\phi_{12} & \lambda_1\lambda_5\phi_{12} & \lambda_1\lambda_6\phi_{12} \\ & \lambda_2^2 + \omega_2 & \lambda_2\lambda_3 & \lambda_2\lambda_4\phi_{12} & \lambda_2\lambda_5\phi_{12} & \lambda_2\lambda_6\phi_{12} \\ & & \lambda_3^2 + \omega_3 & \lambda_3\lambda_4\phi_{12} & \lambda_3\lambda_5\phi_{12} & \lambda_3\lambda_6\phi_{12} \\ & & & \lambda_4^2 + \omega_4 & \lambda_4\lambda_5 & \lambda_4\lambda_6 \\ & & & & \lambda_5^2 + \omega_5 & \lambda_5\lambda_6 \\ & & & & & \lambda_6^2 + \omega_6 \end{pmatrix}$$

Assuming that all the factor loadings are non-zero and the sign of one factor loading is known in each set (one set per factor), $\lambda_1, \lambda_2, \lambda_3$ may be identified from set One and $\lambda_4, \lambda_5, \lambda_6$ may be identified from set Two. Then $\phi_{12}$ may be identified from any unused covariance, and the $\omega_j$ are identifiable from the variances. Thus, all the parameters are identifiable.

Adding more standardized factors, the parameters remain identifiable provided there are at least three variables for each factor with non-zero factor loadings, and the sign of

one factor loading is known in each set. Adding more variables in any set also does no harm.

This establishes the three-variable rule for standardized factors. The parameters will be identifiable provided that there are at least three reference variables per factor, and the errors are independent of one another and of the factors. Comparing these conditions to the three-variable rule for unstandardized factors on page 284, we see the only difference is that the variance of the factor equal to one (and one sign known) is substituted for the factor loading of one (in which case its sign is positive). The result is the following widely-used rule.

**Rule 2b: Three-variable Rule** The parameters of a factor analysis model are identifiable provided

- There are at least three reference variables for each factor.
- For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading for at least one reference variable is equal to one.
- Errors are independent of one another and of the factors.

## 3.3 Equivalence of the Surrogate Models

The three-variable rules for standardized and unstandardized factors were very similar, and it was quite easy to combine them into a single rule. The extreme similarity suggests that the two common surrogate models — the one with a factor loading set to one for each factor, and the one with the variances of the factors set to one — might be the same thing in disguise. In fact, this is correct. The surrogate models in question are one-to-one.

What this means is that if the parameters of the two surrogate models are expressed in terms of the parameters of the original model, then there is a one-to-one (injective) function connecting their parameter vectors. There are two important consequences. First, if the parameters of one surrogate model are shown to be a function of $\boldsymbol{\Sigma}$ and hence identifiable, then the parameters of the other surrogate model are immediately identified as well. This means that it is permissible to check identifiability for one surrogate model even when you intend to fit the other one to your data. Usually, this means doing calculations for a model with factor loadings set to one.

The other consequence is that since the parameter vectors of the two surrogate models are one to one, they capture the same information about the the parameters of the original model — and again, the original model is what we really care about. In this sense, the two surrogate models are equally good. However, the *form* of the information may be more convenient for one of the models, depending on the interests and research objectives of the investigator. Section 3.8 includes examples of extracting the same information the easy way and the hard way.

### 3.3.1 Demonstration of Equivalence

The following is fully rigorous, but much too long and chatty to be called a proof. It is based on three surrogate models. All three models have $p$ factors and $k$ observable variables, and error terms independent of the factors. The first model will be called the *centered original model*. In this model, the intercepts and non-zero expected values in the original model have been swallowed by a re-parameterization, as in the "centered surrogate model" (3.3) on page 278. Note that while the centered original model is a surrogate model, the factor loadings, the covariance matrix of the factors, and the covariance matrix of the error terms are all identical to their counterparts in the original model.

In the centered original model, the observed variables are sorted into two vectors; this is the only difference between the present centered original model and the earlier centered surrogate model. For case $i$, (there are $n$ cases), $\mathbf{d}_{i,1}$ consists of $p$ reference variables for the factors. These are the best available representatives of the factors. If the factors are named appropriately, the factor loadings linking each factor to its reference variable may be assumed strictly positive. The remaining $k-p$ observed variables are collected into $\mathbf{d}_{i,2}$. In the equations for the centered original model, the subscript $i$ is suppressed[5] to reduce notational clutter. Implicitly, everything is independent for $i = 1, \ldots, n$. The model equations are

$$\begin{array}{rcl} \mathbf{d}_1 & = & \boldsymbol{\Lambda}_1\mathbf{F} + \mathbf{e}_1 \\ \mathbf{d}_2 & = & \boldsymbol{\Lambda}_2\mathbf{F} + \mathbf{e}_2, \end{array} \tag{3.15}$$

where all expected values are zero, the $p \times p$ matrix $\boldsymbol{\Lambda}_1$ is diagonal with positive diagonal elements, $cov(\mathbf{F}) = \boldsymbol{\Phi}$, $cov(\mathbf{e}_1) = \boldsymbol{\Omega}_1$, $cov(\mathbf{e}_2) = \boldsymbol{\Omega}_2$ and $cov(\mathbf{e}_1, \mathbf{e}_2) = \boldsymbol{\Omega}_{1,2}$. The parameter vector for this model is

$$\boldsymbol{\theta} = (\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \boldsymbol{\Phi}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{\Omega}_{1,2}).$$

Of course, only the non-redundant elements of the covariance matrices are intended as part of the parameter vector.

The centered original model is then re-parameterized in two different ways, leading to two further surrogate models. These models will cleverly be called Model One and Model Two. Figure 3.5 indicates the process.

In Model One, the number one figures prominently, because it looks like the non-zero factor loadings in $\boldsymbol{\Lambda}_1$ have all been set to one. That is, a factor loading appears to have been set to one for each factor. Actually, it is a re-parameterization accomplished by a change of variables. Letting $\mathbf{F}' = \boldsymbol{\Lambda}_1\mathbf{F}$ yields

$$\begin{array}{rcl} \mathbf{d}_1 & = & \mathbf{F}' + \mathbf{e}_1 \\ \mathbf{d}_2 & = & \boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_1^{-1}\boldsymbol{\Lambda}_1\mathbf{F} + \mathbf{e}_2 \\ & = & \boldsymbol{\Lambda}_2'\mathbf{F}' + \mathbf{e}_2, \end{array} \tag{3.16}$$

[5] For $\mathbf{d}_{i,1}$, $\mathbf{d}_{i,2}$, $\mathbf{F}_i$, $\mathbf{e}_{i,1}$ and $\mathbf{e}_{i,2}$

Figure 3.5: Surrogate Models


where $\mathbf{\Lambda}_2' = \mathbf{\Lambda}_2\mathbf{\Lambda}_1^{-1}$. The covariance matrix of the transformed factors is

$$\mathbf{\Phi}' = cov(\mathbf{\Lambda}_1\mathbf{F}) = \mathbf{\Lambda}_1\mathbf{\Phi}\mathbf{\Lambda}_1^\top,$$

and the covariance matrices of the error terms are the same as for the centered original model. The parameters of Model One are

$$\begin{array}{rcl} \boldsymbol{\theta}_1 & = & (\quad \mathbf{\Lambda}_2', \quad \mathbf{\Phi}', \quad \mathbf{\Omega}_1', \quad \mathbf{\Omega}_2', \quad \mathbf{\Omega}_{1,2}' \quad ) \\ & = & (\quad \mathbf{\Lambda}_2\mathbf{\Lambda}_1^{-1}, \quad \mathbf{\Lambda}_1\mathbf{\Phi}\mathbf{\Lambda}_1^\top, \quad \mathbf{\Omega}_1, \quad \mathbf{\Omega}_2, \quad \mathbf{\Omega}_{1,2} \quad ). \end{array} \tag{3.17}$$

In Model Two, the factors are scaled to have variance one. Since they are already have expected value zero, this means they are standardized. Let $\mathbf{V}$ denote a diagonal matrix with the variances of the factors (the diagonal elements of $\mathbf{\Phi}$) on the main diagonal. Transforming the factors by $\mathbf{F}'' = \mathbf{V}^{-1/2}\mathbf{F}$,

$$\begin{array}{rcl} \mathbf{d}_1 & = & \mathbf{\Lambda}_1\mathbf{F} + \mathbf{e}_1 \\ & = & \mathbf{\Lambda}_1\mathbf{V}^{1/2}\mathbf{V}^{-1/2}\mathbf{F} + \mathbf{e}_1 \\ & = & \mathbf{\Lambda}_1''\mathbf{F}'' + \mathbf{e}_1, \end{array}$$

and

$$\begin{array}{rcl} \mathbf{d}_2 & = & \mathbf{\Lambda}_2\mathbf{F} + \mathbf{e}_2 \\ & = & \mathbf{\Lambda}_2\mathbf{V}^{1/2}\mathbf{V}^{-1/2}\mathbf{F} + \mathbf{e}_2 \\ & = & \mathbf{\Lambda}_2''\mathbf{F}'' + \mathbf{e}_2. \end{array}$$

Summarizing, the equations for Model Two are

$$\begin{array}{rcl} \mathbf{d}_1 & = & \mathbf{\Lambda}_1''\mathbf{F}'' + \mathbf{e}_1 \\ \mathbf{d}_2 & = & \mathbf{\Lambda}_2''\mathbf{F}'' + \mathbf{e}_2, \end{array} \tag{3.18}$$

where $\mathbf{\Lambda}_1'' = \mathbf{\Lambda}_1\mathbf{V}^{1/2}$ and $\mathbf{\Lambda}_2'' = \mathbf{\Lambda}_2\mathbf{V}^{1/2}$. The covariance matrix of the transformed factors for Model Two is

$$\mathbf{\Phi}'' = cov(\mathbf{V}^{-1/2}\mathbf{F}) = \mathbf{V}^{-1/2}\mathbf{\Phi}\mathbf{V}^{-1/2},$$

and the parameter vector is

$$\begin{array}{rcl} \boldsymbol{\theta}_2 & = & ( \quad \boldsymbol{\Lambda}_1'', \quad \boldsymbol{\Lambda}_2'', \quad \boldsymbol{\Phi}'', \quad \boldsymbol{\Omega}_1'', \quad \boldsymbol{\Omega}_2'', \quad \boldsymbol{\Omega}_{1,2}'' \quad ) \\ & = & ( \quad \boldsymbol{\Lambda}_1\mathbf{V}^{1/2}, \quad \boldsymbol{\Lambda}_2\mathbf{V}^{1/2}, \quad \mathbf{V}^{-1/2}\boldsymbol{\Phi}\mathbf{V}^{-1/2}, \quad \boldsymbol{\Omega}_1, \quad \boldsymbol{\Omega}_2, \quad \boldsymbol{\Omega}_{1,2} \quad ). \end{array} \tag{3.19}$$

The objective here is to show that $\boldsymbol{\theta}_1$ in Expression (3.17) and $\boldsymbol{\theta}_2$ in Expression (3.19) are connected by a one-to-one function; that is, $\boldsymbol{\theta}_1$ is a function of $\boldsymbol{\theta}_2$, and $\boldsymbol{\theta}_2$ is a function of $\boldsymbol{\theta}_1$.

To find $\boldsymbol{\theta}_1$ as a function of $\boldsymbol{\theta}_2$, it is enough to express $\boldsymbol{\Lambda}_2'$ and $\boldsymbol{\Phi}'$ in terms of the elements of $\boldsymbol{\theta}_2$. We have

$$\begin{array}{rcl} \boldsymbol{\Lambda}_2''\boldsymbol{\Lambda}_1''^{-1} & = & \boldsymbol{\Lambda}_2\mathbf{V}^{1/2}\left(\boldsymbol{\Lambda}_1\mathbf{V}^{1/2}\right)^{-1} \\ & = & \boldsymbol{\Lambda}_2\mathbf{V}^{1/2}\mathbf{V}^{-1/2}\boldsymbol{\Lambda}_1^{-1} \\ & = & \boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_1^{-1} \\ & = & \boldsymbol{\Lambda}_2' \end{array} \tag{3.20}$$

and

$$\begin{array}{rcl} \boldsymbol{\Lambda}_1''\boldsymbol{\Phi}''\boldsymbol{\Lambda}_1''^{\top} & = & \left(\boldsymbol{\Lambda}_1\mathbf{V}^{1/2}\right)\left(\mathbf{V}^{-1/2}\boldsymbol{\Phi}\mathbf{V}^{-1/2\top}\right)\left(\boldsymbol{\Lambda}_1\mathbf{V}^{1/2}\right)^{\top} \\ & = & \boldsymbol{\Lambda}_1\boldsymbol{\Phi}\mathbf{V}^{-1/2}\mathbf{V}^{1/2}\boldsymbol{\Lambda}_1^{\top} \\ & = & \boldsymbol{\Lambda}_1\boldsymbol{\Phi}\boldsymbol{\Lambda}_1^{\top} \\ & = & \boldsymbol{\Phi}'. \end{array} \tag{3.21}$$

Notice that going in this direction, the assumption that $\boldsymbol{\Lambda}_1$ is diagonal is not used. All that's necessary is the existence of an inverse. Also notice that by the invariance principle of maximum likelihood estimation, one could simply place hats on the parameter matrices of (3.20) and (3.21) to obtain estimates for Model One from those for Model Two, without re-fitting the model. Similarly, expressions (3.22), (3.23) and (3.24) below may be used to obtain Model Two estimates directly from Model One estimates.

To go from Model One to Model Two, a bit of background is required. Let $\mathbf{A}$ and $\mathbf{B}$ be diagonal (and square) matrices of the same size. Then $\mathbf{AB} = \mathbf{BA}$, and if all the elements are non-negative, $(\mathbf{AB})^{1/2} = \mathbf{A}^{1/2}\mathbf{B}^{1/2}$. Also, let $\mathbf{A}$ be a square matrix, and let $dg(\mathbf{A})$ denote the diagonal matrix with diagonal elements equal to the diagonal elements of $\mathbf{A}$. For example, in the current problem, $dg(\boldsymbol{\Phi}) = \mathbf{V}$.

Now, suppose the diagonal elements of $\boldsymbol{\Lambda}_1$ are labelled $\lambda_1, \ldots, \lambda_p$. Because $\boldsymbol{\Lambda}_1$ is diagonal, the $j$th diagonal element of $\boldsymbol{\Phi}' = \boldsymbol{\Lambda}_1\boldsymbol{\Phi}\boldsymbol{\Lambda}_1^{\top}$ is $\lambda_j\phi_{j,j}\lambda_j = \lambda_j^2\phi_{j,j}$. This is also the $j$th diagonal element of $\boldsymbol{\Lambda}_1\mathbf{V}\boldsymbol{\Lambda}_1$, which is diagonal because the product of diagonal matrices is diagonal. In short, $dg(\boldsymbol{\Phi}') = \boldsymbol{\Lambda}_1\mathbf{V}\boldsymbol{\Lambda}_1$.

The task is to find $\boldsymbol{\theta}_2$ as a function of $\boldsymbol{\theta}_1$. That is, we need to express $\boldsymbol{\Lambda}_1''$, $\boldsymbol{\Lambda}_2''$ and $\boldsymbol{\Phi}''$ in terms of single-prime quantities. The variances and covariances in $\boldsymbol{\Omega}_1''$, $\boldsymbol{\Omega}_2''$ and $\boldsymbol{\Omega}_{1,2}''$ are automatic, because the transformations considered here do not affect these error matrices.

Using the special properties of diagonal matrices indicated above,

$$\begin{aligned} dg(\mathbf{\Phi}')^{1/2} &= (\mathbf{\Lambda}_1\mathbf{V}\mathbf{\Lambda}_1)^{1/2} \\ &= \left(\mathbf{\Lambda}_1\mathbf{V}^{1/2}\mathbf{V}^{1/2}\mathbf{\Lambda}_1\right)^{1/2} \\ &= \left(\mathbf{\Lambda}_1\mathbf{V}^{1/2}\mathbf{\Lambda}_1\mathbf{V}^{1/2}\right)^{1/2} \\ &= \mathbf{\Lambda}_1\mathbf{V}^{1/2} \\ &= \mathbf{\Lambda}_1'' \end{aligned} \tag{3.22}$$

and

$$\begin{aligned} \mathbf{\Lambda}_2' dg(\mathbf{\Phi}')^{1/2} &= (\mathbf{\Lambda}_2\mathbf{\Lambda}_1^{-1})(\mathbf{\Lambda}_1\mathbf{V}^{1/2}) \\ &= \mathbf{\Lambda}_2\mathbf{V}^{1/2} \\ &= \mathbf{\Lambda}_2'' \end{aligned} \tag{3.23}$$

and

$$\begin{aligned} dg(\mathbf{\Phi}')^{-1/2}\,\mathbf{\Phi}'\,dg(\mathbf{\Phi}')^{-1/2} &= (\mathbf{\Lambda}_1\mathbf{V}^{1/2})^{-1}(\mathbf{\Lambda}_1\mathbf{\Phi}\mathbf{\Lambda}_1^\top)(\mathbf{\Lambda}_1\mathbf{V}^{1/2})^{-1} \\ &= \mathbf{V}^{-1/2}\mathbf{\Lambda}_1^{-1}\mathbf{\Lambda}_1\mathbf{\Phi}\mathbf{\Lambda}_1\mathbf{V}^{-1/2}\mathbf{\Lambda}_1^{-1} \\ &= \mathbf{V}^{-1/2}\mathbf{\Phi}\mathbf{\Lambda}_1\mathbf{\Lambda}_1^{-1}\mathbf{V}^{-1/2} \\ &= \mathbf{V}^{-1/2}\mathbf{\Phi}\mathbf{V}^{-1/2} \\ &= \mathbf{\Phi}''. \end{aligned} \tag{3.24}$$

This establishes that the parameters of Models One and Two are one to one. In Figure 3.5, there could be a two-headed arrow between Model One and Model Two[6]. As a corollary, we have the following useful rule.

**Rule 2i: The Equivalence Rule** For a centered factor analysis model with at least one reference variable for each factor, suppose that surrogate models are obtained by either standardizing the factors, or by setting the factor loading of a reference variable equal to one for each factor. Then the parameters of one surrogate model are identifiable if and only if the parameters of the other surrogate model are identifiable.

[6]The student may be like, Okay, this is all correct, but how would anyone even think of some of these functions, especially the formula for $\mathbf{\Phi}''$ in (3.24)? The key is that surprisingly, if you standardize $\mathbf{F}'$ you get $\mathbf{F}''$. This makes it easy to write the double-prime matrices in terms of the single-prime matrices. Going in the other direction, try a change of variables in which $\mathbf{\Lambda}_1''$ is absorbed into $\mathbf{F}''$, effectively setting a factor loading to one for each factor. The change of variables is $\mathbf{F}''' = \mathbf{\Lambda}_1''\mathbf{F}''$, which happens to equal $\mathbf{F}'$.

This is so remarkable that it bears repeating. If you set a factor loading to one for each factor in the centered original model, you get Model One. If you standardize the factors in the centered original model, you get Model Two. If you standardize the factors in Model One, you get Model Two. If you set a factor loading to one for each factor in Model Two, you get Model One. It feels like a projection of some kind.

### 3.3.2 Choosing a Surrogate Model

So, the two standard surrogate models are equivalent. Their identifiability status is the same, and they contain the same information about the parameters of the original model. In actual data analysis, which one should you use?

**An advantage of unit factor loadings** Certainly, when factor loadings are set to one it's easier to calculate $\mathbf{\Sigma}$ by hand as a function of the surrogate model parameters. It's also easier to solve for model parameters in terms of $\sigma_{ij}$ quantities – again, if the calculations are done by hand. For models to which the standard identifiability rules do not apply, this can be very helpful.

**An advantage of standardized factors** Recall that the identifiable parameters of a surrogate model are actually identifiable *functions* of the parameters of the original model. It's helpful if these functions correspond directly to something you want to know about. When a factor loading is seemingly set to one, the other factor loadings (which would appear on the other arrows coming from that factor) are actually *ratios* of factor loadings, with the invisible factor loading in the denominator. Thus, all the other factor loadings are relative to the one that disappeared, making the invisible factor loading a kind of reference quantity. When such ratios of factor loadings are of interest, setting a factor loading to one for each factor is a good choice.

On the other hand, the variances and covariances of the factors under the surrogate model are the original quantities multiplied by the reference factor loadings. This preserves nothing of interest from the original model, apart from the signs of the covariances.

In contrast, consider the covariances between factors when the factors are standardized. For a general factor analysis model, suppose that the factor $F_j$ has been standardized. Using double primes for consistency with the notation of Section 3.3.1, let $F_j'' = \frac{1}{\sqrt{\phi_{j,j}}}F_j$. As well, the factor $F_\ell$ has been standardized by $F_\ell'' = \frac{1}{\sqrt{\phi_{\ell,\ell}}}F_\ell$. Then the covariance between the transformed factors is

$$\begin{aligned}
\phi''_{j,\ell} &= cov(F_j'', F_\ell'') \\
&= cov(\frac{1}{\sqrt{\phi_{j,j}}}F_j, \frac{1}{\sqrt{\phi_{\ell,\ell}}}F_\ell) \\
&= \frac{1}{\sqrt{\phi_{j,j}\phi_{\ell,\ell}}}cov(F_j, F_\ell) \\
&= \frac{\phi_{j,\ell}}{\sqrt{\phi_{j,j}\phi_{\ell,\ell}}} \\
&= Corr(F_j, F_\ell).
\end{aligned}$$

The covariances between factors under the surrogate model are not just correlations, which they must be since the factors have variance one. They are *the* correlations — that is, they are exactly the correlations between factors under the original model. As such, those $\phi''_{j,\ell}$ quantities are very easy to understand and interpret. Confidence intervals are

meaningful. This is a significant advantage to standardizing the factors, though it's more helpful for pure factor analysis than for general structural equation models with a causal structure in the latent variables.

While it might be tempting to set a factor loading to one for a factor and also standardize that same factor, it's a very bad idea. You can do it, but this reduction of the original parameter space cannot be accomplished by a change of variables. Consequently the connection of the resulting model parameters to the parameters of the original model would be mysterious. Furthermore, doing both at once usually implies equality constraints on $\mathbf{\Sigma}$ that do not follow from the original model, invalidating the goodness of fit test. It's something you just should not do.

To summarize, setting a factor loading to one for each factor (Model One) is attractive because it makes calculations easier. Standardizing factors (Model Two) is attractive because the resulting covariances between factors are the correlations between factors under the original model. As the following example shows, it is possible to enjoy the benefits of both surrogate models. If identifiability is unclear and you prefer the interpretability of the model with standardized factors, you can safely show identifiability for Model One and then Fit Model Two to the data.

**Example 3.3.1** *The Political Democracy Example*

This data set is discussed by Bollen [10] and other authors. Based on news reports and other sources, a panel of experts rated a sample of 72 developing countries on the following variables.

$d_1$: Freedom of the press in 1960

$d_2$: Freedom of political opposition in 1960

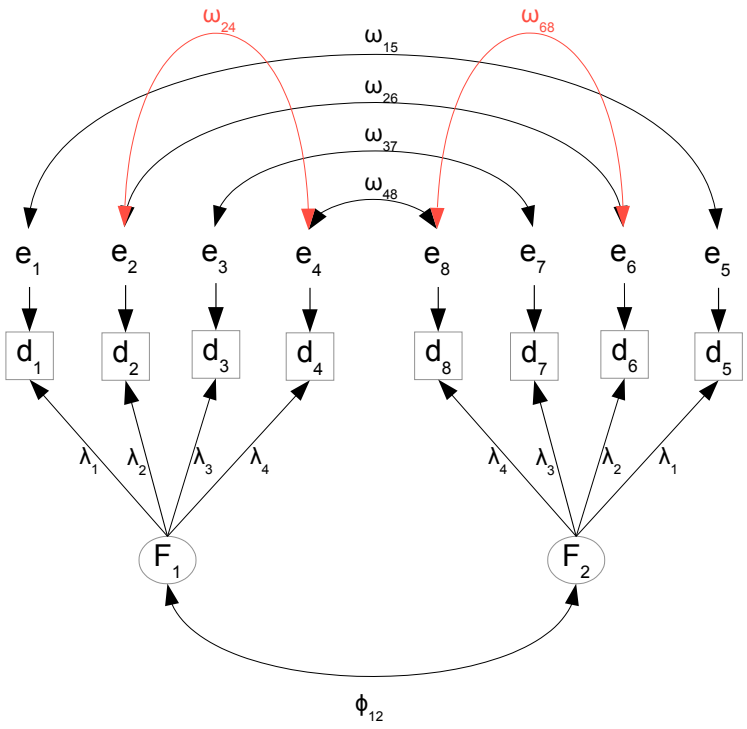
$d_3$: Fairness of elections in 1960

$d_4$: Effectiveness of the elected legislature in 1960

The variables $d_5$ through $d_8$ represent the same quantities for the year 1965. There are two hypothesized factors, strength of political democracy in 1960, and strength of political democracy in 1965. Figure 3.6 shows a path diagram, which in my humble opinion is an improvement on Bollen's Figure 7.3 on page 235 — even though they contain the same information. The factor $F_1$ is political democracy in 1960, and $F_2$ is political democracy in 1965. The factor loadings are hypothesized to be the same in 1960 and 1965, though the variances of the error terms might not be. Though the variables $d_5$ through $d_8$ correspond directly to $d_1$ through $d_4$, they are sorted in the opposite order to allow for the curved arrows between arrow terms.

It is those curved arrows, representing covariances between error terms, that make this model unusual. There are *two* sources of covariance between the 1960 and 1965 variables, only one of which the covariance between factors. Even so, it turns out that all the parameters are identifiable.

Figure 3.6: Political Democracy Factor Model




The curved arrows between error terms reflect really thoughtful, good modelling. For example, freedom of the press in 1960 and freedom of the press in 1965 are assessed based on similar information from mostly the same sources, so that both observed variables are impacted by similar sources of bias. The latent variables involved are not part of the model, so they are represented by a covariance between error terms. The same applies to the other three pairs of variables ($d_2$ and $d_6$, $d_3$ and $d_7$, $d_4$ and $d_8$). Covariances within years are in red just for visual contrast. The measurement of $d_2$ and $d_4$ have something extra in common, as do $d_6$ and $d_8$. I'm not sure what it is. Anyway, this is good. Most people just assume error terms uncorrelated without really thinking about it.

We are interested in a surrogate model with standardized factors, and we need to verify identifiability before trying to fit the model. Identifiability will be a lot easier to check for a surrogate model with $\lambda_1 = 1$. The equivalence rule says that it's okay to check one model and then feel comfortable fitting the other one.

Without the curved arrows, this model would be identifiable at a glance by the three-variable rule. With the curved arrows, it will be possible to get the job done by combining rules and a few simple calculations. Bear in mind that once a parameter has been identified, it may be used in the solutions for other parameters.

Here are the model equations for the model with factor loadings set to one.

$$\begin{aligned} d_1 &= F_1 + e_1 \\ d_2 &= \lambda_2 F_1 + e_2 \\ d_3 &= \lambda_3 F_1 + e_3 \\ d_4 &= \lambda_4 F_1 + e_4 \\ d_5 &= F_2 + e_5 \\ d_6 &= \lambda_2 F_2 + e_6 \\ d_7 &= \lambda_3 F_2 + e_7 \\ d_8 &= \lambda_4 F_2 + e_8 \end{aligned}$$

First, apply the three-variable rule to the submodel with $F_1$, $d_1$, $d_2$ and $d_3$[7]. The parameters $\lambda_2$, $\lambda_3$, $\phi_{11}$, $\omega_{11}$, $\omega_{22}$ and $\omega_{33}$ are identified.

Adding $d_4$ to the system is non-standard, because of the covariance between $e_4$ and $e_2$. However,

$$\begin{aligned} \sigma_{14} &= cov(d_1, d_4) \\ &= cov(F_1 + e_1)(\lambda_4 F_1 + e_4) \\ &= \lambda_4 Var(F_1) + 0 + 0 + 0 \\ &= \lambda_4 \phi_{11}, \end{aligned}$$

and $\lambda_4 = \sigma_{14}/\phi_{11}$ is identified. Then,

$$\begin{aligned} \sigma_{24} &= cov(d_2, d_4) \\ &= cov(\lambda_2 F_1 + e_2)(\lambda_4 F_1 + e_4) \\ &= \lambda_2 \lambda_4 Var(F_1) + 0 + 0 + cov(e_2, e_4) \\ &= \lambda_2 \lambda_4 \phi_{11} + \omega_{24}, \end{aligned}$$

and $\omega_{24} = \sigma_{24} - \lambda_2 \lambda_4 \phi_{11}$ is identified.

Repeating these operations for the submodel with $F_2$, $d_5$, $d_6$, $d_7$ and $d_8$, the variance parameters $\omega_{55}, \ldots, \omega_{88}$ are identified. Also, it is clear that if the factor loadings for 1965 were different from 1960, they would be identified as well.

Now we turn to the sources of covariance between the 1960 and 1965 measurements.

$$\begin{aligned} \sigma_{18} &= cov(d_1, d_8) \\ &= cov(F_1 + e_1)(\lambda_4 F_2 + e_8) \\ &= \lambda_4 cov(F_1, F_2) \\ &= \lambda_4 \phi_{12}. \end{aligned}$$

Then, $\phi_{12} = \sigma_{18}/\lambda_4$ is identified.

[7]What about the curved arrows? There are no curved arrows connecting $e_1$, $e_2$ and $e_3$, so the calculations for this subsystem, if we had to re-do them, would be unaffected.

Now it's straightforward to solve for the remaining covariances between errors.

$$\begin{aligned} \sigma_{15} &= cov(d_1, d_5) \\ &= cov(F_1 + e_1)(F_2 + e_5) \\ &= cov(F_1, F_2) + cov(e_1, e_5) \\ &= \phi_{12} + \omega_{15} \\ \Longrightarrow \quad & \omega_{15} = \sigma_{15} - \phi_{12}, \end{aligned}$$

and

$$\begin{aligned} \sigma_{26} &= cov(d_2, d_6) \\ &= cov(\lambda_2 F_1 + e_2)(\lambda_2 F_2 + e_6) \\ &= \lambda_2^2 cov(F_1, F_2) + cov(e_2, e_6) \\ &= \lambda_2^2 \phi_{12} + \omega_{26} \\ \Longrightarrow \quad & \omega_{26} = \sigma_{26} - \lambda_2^2 \phi_{12}, \end{aligned}$$

and similarly for $\omega_{37}$ and $\omega_{48}$.

I got a bit carried away here, and showed elementary details that you are probably able to do in your head. This may obscure the fact that establishing identifiability for this interesting model is really pretty easy, especially when working with the surrogate model in which factor loadings are set to one. It's not necessary to calculate the whole covariance matrix $\mathbf{\Sigma}$, and all the calculations that are really needed could be done on a sheet of scratch paper.

## 3.4 The Reference Variable rule

This rule comes from applying the equivalence rule to the reference variable rule for unstandardized factors, on page 286, so that it holds for both the common surrogate factor analysis models. It says that under some other conditions that are fairly mild and easy to satisfy, the parameters of a model with three observable variables per factor will be identifiable, provided that one of the variables is a reference variable. The other two variables may be influenced by all the factors. Here is the rule.

**Rule 2c: The Reference Variable Rule** The parameters of a factor analysis model are identifiable except possibly on a set of volume zero in the parameter space, provided

- The number of observable variables (including reference variables) is at least three times the number of factors.
- There is at least one reference variable for each factor.
- For each factor, either the variance equals one and the sign of the reference variable's factor loading is known, or the factor loading of the reference variable is equal to one.
- Divide the observable variables into sets. The first set contains one reference variable for each factor. The number of variables in the second set and the number in the third set is also equal to the number of factors. The fourth set may contain any number of additional variables, including zero. The error terms for the variables in the first three sets may have non-zero covariance within sets, but not between sets. The error terms for the variables in the fourth set may have non-zero covariance within the set, and with the error terms of sets two and three, but they must have zero covariance with the error terms of the reference variables.

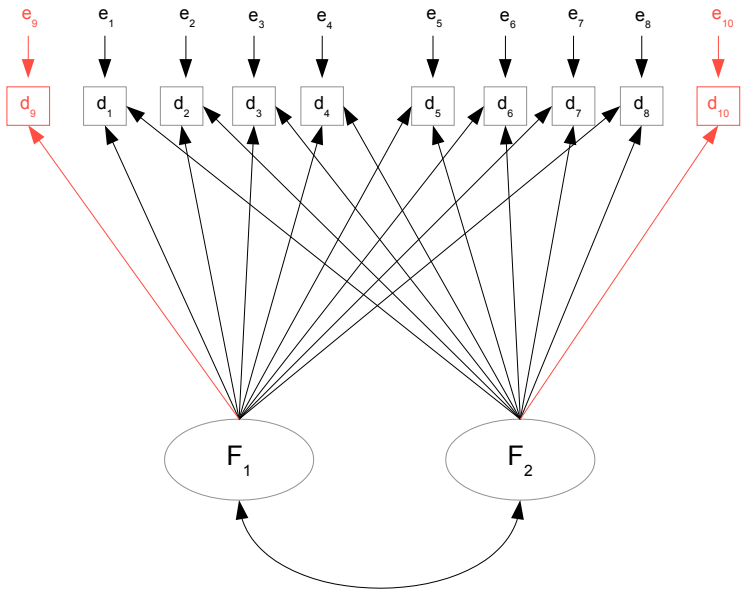
The last condition is unusually long. It describes patterns of permissible covariances between error terms. That's important and we will get back to it, but for now just observe that the condition is satisfied for models in which all the error terms are independent – something that is almost the default for factor analysis models[8].

**The rule with independent errors** Figure 3.7 illustrates the reference variable rule with independent errors, and also gives an idea of the modelling flexibility the rule permits. The black part of the model is a direct copy of the unrestricted exploratory factor analysis model of Figure 2.1 in Chapter 2. Then, reference variables for the factors (the observable variables $d_9$ and $d_{10}$) have been added in red. The resulting model is immediately identifiable, assuming the factors are standardized or the factor loadings on the red arrows are set to one.

This shows that with a few extra variables of the right kind, the parameters of an exploratory factor analysis can be estimated without any fuss. If the factors are standardized, the covariances between factors are the correlations between factors under the original model. The factor loadings under the surrogate model are positive multiples of the the corresponding factor loadings under the original model. While the actual values of the original factor loadings are not knowable, it is possible to estimate and test whether their signs are positive, negative or zero. That's enough for many purposes. All the technical gymnastics from Chapter 2, like rotation to simple structure, viewing the resulting factor solution as a scientific theory and invoking Occam's Razor from the philosophy of

[8]Independent errors are universal in exploratory factor analysis, and many confirmatory factor analysis models seem to have inherited this feature. In Chapter 2 on page 214, independent errors are traced to Spearman's (1904) "Law of the Universal Unity of the Intellective Function." [60]

Figure 3.7: Adding reference variables to an unrestricted factor model




science to justify it on the grounds of simplicity — all of that is unnecessary if you have the right kind of data set. The reference variable rule tells you what kind of data set you need.

There is a general point here. Lack of identifiability is often a problem with the study design, not the model. This makes sense. Identifiability is literally about what can be known. Naturally, there is an intimate connection to research design.

One other observation is that while the black part of Figure 3.7 is an exploratory factor analysis model, the whole analysis can't be completely exploratory. You really need to have a good idea of what the factors are before designing measurement procedures (reference variables) that clearly tap one factor but not any of the others.

**Statement of the model** Rule 2c goes on and on about covariances between error terms. To clarify the discussion, a full statement of the model will be helpful. This is an adaptation of Model 3.1 on page 286. Independently for $i = 1, \ldots, n$,

$$\begin{aligned} \mathbf{d}_{i,1} &= \boldsymbol{\Lambda}_1 \mathbf{F}_i + \mathbf{e}_{i,1} \\ \mathbf{d}_{i,2} &= \boldsymbol{\Lambda}_2 \mathbf{F}_i + \mathbf{e}_{i,2} \\ \mathbf{d}_{i,3} &= \boldsymbol{\Lambda}_3 \mathbf{F}_i + \mathbf{e}_{i,3} \\ \mathbf{d}_{i,4} &= \boldsymbol{\Lambda}_4 \mathbf{F}_i + \mathbf{e}_{i,4}, \end{aligned} \tag{3.25}$$

where

- $\mathbf{d}_{i,1}$, $\mathbf{d}_{i,2}$ and $\mathbf{d}_{i,3}$ are $p \times 1$ observable random vectors.

- $\mathbf{d}_{i,4}$ need not be present. If it is present, it is an $m \times 1$ observable random vector.
- $\mathbf{F}_i$ ($F$ for Factor) is a $p \times 1$ latent random vector with expected value zero $cov(\mathbf{F}_i) = \mathbf{\Phi}$.
- $\mathbf{\Lambda}_1$ is a $p \times p$ diagonal of constants, with non-zero diagonal elements. The diagonal elements may be assumed positive.
- $\mathbf{\Lambda}_2$ and $\mathbf{\Lambda}_3$ are $p \times p$ non-singular matrices of constants.
- $\mathbf{\Lambda}_4$, if it is present, is an $m$ by $p$ matrix of constants.
- $\mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,4}$ are vectors of error terms, with expected value zero, covariance matrix $cov(\mathbf{e}_{i,j}) = \mathbf{\Omega}_{j,j}$ for $j = 1, \ldots, 4$, and
  - $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) = cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,3}) = cov(\mathbf{e}_{i,2}, \mathbf{e}_{i,3}) = \mathbf{O}$, all $p \times p$ matrices.
  - $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,4}) = \mathbf{O}$, a $p \times m$ matrix.
  - $cov(\mathbf{e}_{i,2}, \mathbf{e}_{i,4}) = \mathbf{\Omega}_{2,4}$ and $cov(\mathbf{e}_{i,3}, \mathbf{e}_{i,4}) = \mathbf{\Omega}_{3,4}$.
- Either the diagonal elements of $\mathbf{\Lambda}_1$ or the diagonal elements of $\mathbf{\Phi}$ are equal to one.

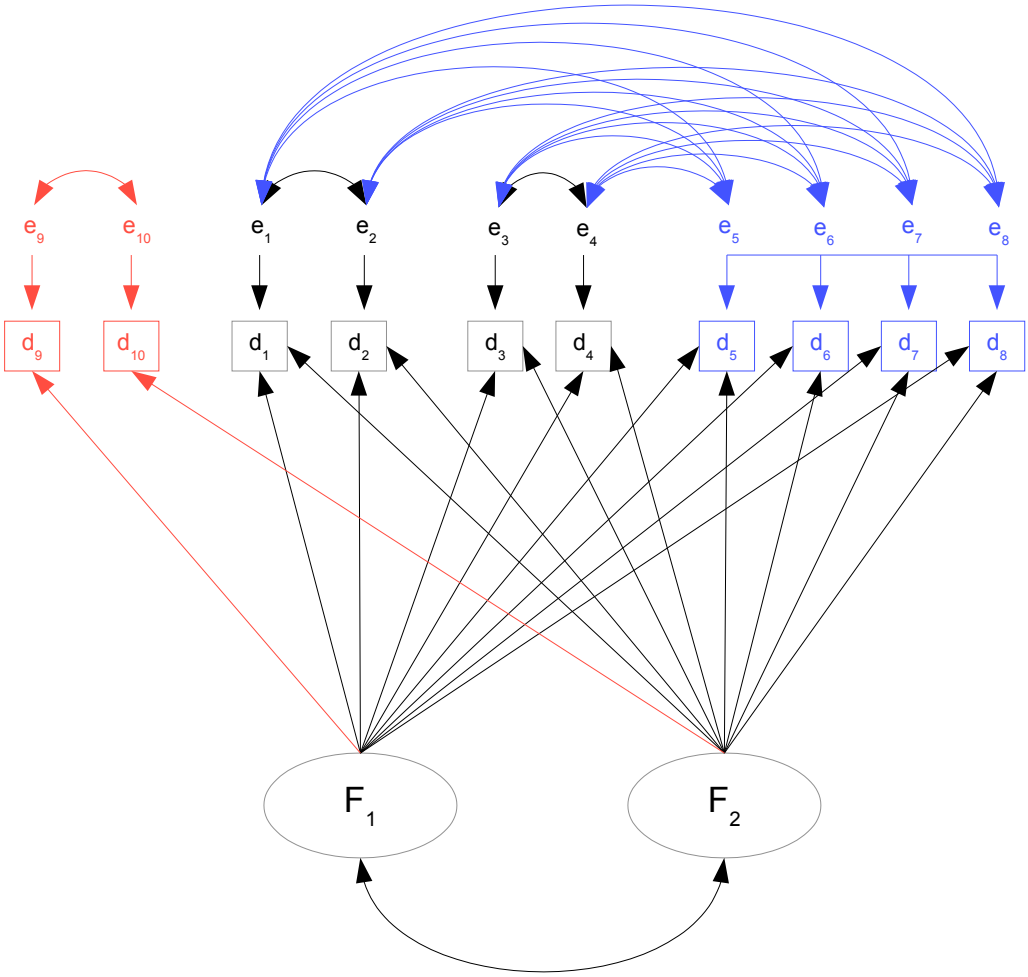
What's happening here is that the reference variables for the factors are being placed in $\mathbf{d}_{i,1}$, and then the remaining observable variables are being allocated to $\mathbf{d}_{i,2}$, $\mathbf{d}_{i,3}$, and possibly $\mathbf{d}_{i,4}$, depending on the potential for non-zero covariance between their error terms.

Figure 3.8 is a re-arranged version of Figure 3.7, showing the covariances between errors that the rule allows. The reference variables $d_9$ and $d_{10}$ are grouped together in $\mathbf{d}_{i,1}$, while $\mathbf{d}_{i,2}$ contains $d_1$ and $d_2$, and $\mathbf{d}_{i,3}$ contains $d_3$ and $d_4$. The remaining observed variables, $d_5$ through $d_8$, are placed in $\mathbf{d}_{i,4}$. With the colour coding, perhaps you can see it. $e_9$ and $e_{10}$ are correlated, $e_1$ and $e_2$ are correlated, $e_3$ and $e_4$ are correlated, $e_5$ through $e_8$ are correlated, and there are four blue connectors running to each of $e_1$, $e_2$, $e_3$ and $e_4$.

**Correlated error terms** To understand how error terms might be correlated, consider what an error term represents. In a path diagram, suppose that a variable $y$ has three arrows pointing toward it from $x_1$, $x_2$ and $x_3$, and one more arrow coming from $e$, an error term. The model is saying that $y$ is influenced by the $x$ variables, but it's not completely determined by them. There are other, unmeasured variables that affect $y$. We don't know what they all are, or even how many there are. Anyway, we roll them together and call them $e$. That is, the error term in a model equation is *everything else* that affects the endogenous variable, apart from the other variables on the right side of the equation.

Thinking of an error term as a giant linear combination of unmeasured and perhaps even unimagined variables (probably not a bad approximation), it is clear that if any variables appear in more than one such linear combination, or if some of the variables in two different linear combinations have non-zero covariance, then the error terms will have non-zero covariance as well. This is how the curved arrows between error terms arise. "Everthing else" includes some of the same influences, or related influences.

Figure 3.8: Allowable covariances between error terms




When observable variables are recorded at roughly the same time and by the same method, then correlated errors of measurement are practically unavoidable. For example, suppose that a sample of high school students takes a standardized test, consisting of sub-tests on mathematical and verbal material. Scores on the sub-tests will be two different observable variables. Some students will suffer from test anxiety more than others, some will be more test-wise than others, some will have gotten more sleep the night before, and some students will simply be having a better day than others. The list goes on. The point is that these unmeasured factors are not explicitly part of the model, but they will influence performance on both the math test and the verbal test. They are a source of covariance between the two measures, over and above any covariance between the *factors* (say, verbal ability and mathematical ability) that the tests seek to measure. All this would be represented by a curved, double-headed arrow between the error terms.

If the variables in a study come from questionnaires, the case for correlated error terms

is even stronger. Consider a questionnaire with a lot of questions about the respondent's workplace. Mixed together are questions from several sub-scales that seek to assess the quality of relations with co-workers, the perceived overall fairness of management, opportunities for advancement, and the respondent's job satisfaction. In the model, these sub-scales are going to be separate observed variables, each with its own error term. The respondent's current mood will certainly affect all the responses, as may happy or unhappy events outside the workplace. Some respondents will not really believe their responses will not get back to the employer, and will play it safe by saying that everything is great — on *all* the questions. Others will take the opportunity to vent their frustrations, and paint a picture of everything that is darker than the one they actually experience from day to day.

Also, one should not minimize the extent to which social science research (including market research and behavioural economics) is a social transaction between the participant and the investigator. Many people answering questionnaires certainly seek to represent themselves in a favourable light [19],[52] and often politely tell the investigator what they think the investigator wants to hear [49]. All these dynamics (which are only rarely what the investigator wants to study) push the responses to clusters of questions up or down together. In the path diagram they are represented by curved, double-headed arrows connecting error terms.

It would be nice if all error terms could have covariances with one another that are unknown parameters, and not assumed zero. This is how it goes in ordinary multivariate regression, with all variables observable. Once there are latent variables, however, identifiability becomes an issue. Certainly, if all the error terms in a factor analysis model have non-zero covariance with each other, then the parameter count rule establishes that all the parameters of the model cannot be identifiable. So, what should we do?

One alternative is to assume the covariances are zero, and hope. Just hope that the processes involving the variables in the model are a lot stronger than the processes leading to correlated error terms. The model is not quite correct and everyone knows it, but it should not be too misleading. I think it's fair to say that almost all the usual factor analysis models with independent error terms are based on this kind of hope. Too often, the model does not fit; this can include negative variance estimates, the so-called Heywood case described on page 226. Note that the negative variance in Example 1.5.1 was produced by correlated error terms.

There is another, better solution: careful research design. This means doing some thinking about the model to *before* collecting the data. The first thing to note is that some error terms can legitimately be assumed to have zero covariance – on the basis of reasonable modelling, not just hope. For example, suppose that a medical technician records the height of a patient, and also asks about occupation (later to be converted into a numerical index of occupational prestige). There is surely measurement error in both operations, but no particular reason to suspect that the errors might vary systematically with one another. Again, suppose a participant in a study fills out several questionnaires designed to assess racism and other social attitudes. The error terms are correlated, without a doubt. But if the person also grants access to her cell phone data, then a racism measure derived from Facebook likes (again imperfect, as always) could arguably

have an error term independent of the error terms of the self-report data.

As another example, here's a quote from page 84 in Section 0.10.3 on the double measurement design in Chapter 0: "...farmers who overestimate their number of pigs may also overestimate their number of cows. On the other hand, if the number of pigs is counted once by the farm manager at feeding time and on another occasion by a research assistant from an areal photograph, then it would be fair to assume that the errors of measurement for the different methods are uncorrelated." There are more examples in the BMI Health Study (Section 0.10.4 of Chapter 0, page 89). The point is that error terms need not *always* be correlated. If two observable variables are measured by different methods, on different occasions and ideally by different personnel, it's usually reasonable to assume that their errors are independent.

This is where the reference variable rule comes in. Like the double measurement rule, it allows correlated errors *within* certain sets of observed variables, as long as there is zero covariance *between* sets — and identifiability is still preserved. It requires advance planning, and the data collection will inevitably be more demanding. However, it's not really a lot to ask. In experimental research (with random assignment of cases to treatment conditions), it is quite common to plan the data collection and statistical analysis at the same time, and to take a lot of care about the details of procedure. The same thing applies to good research using strictly observational data. It's not enough to just hand out a bunch of questionnaires.

**Example 3.4.1** *Student Mental Health*

Let's give some content to Figure 3.8. The result will be a re-arrangement of the observed variables, with some of the curved, double-headed arrows eliminated. Suppose it's a study of student mental health. The investigators believe that anxiety and depression are the two main mental health problems that many young people face. They mean long-lasting, chronic anxiety and depression, not just getting anxious or sad about something, and then the feeling passes. The investigators are interested in how these traits are related to one another. Specifically, they want to estimate the correlation between true (not just reported) long-term anxiety and true long-term depression.

The participants are volunteer High School students. They all take part in a one-on-one interview with a clinical psychologist, who asks some very carefully chosen questions, and assesses them on level of persisting anxiety and level of persisting depression. I am willing to believe that the anxiety assessment reflects true anxiety plus error, and is not directly influenced by true depression. I also can believe that the depression assessment reflects true depression plus error, and is not directly influenced by true anxiety. So both clinical assessments are reference variables.

Regardless of what the clinical psychologist might claim, it's unavoidable that common extraneous factors will affect both assessments. For example, regardless of how skilled and non-threatening the psychologist might be, some people will just be less likely to report symptoms; it's a matter of personal style. The measurement errors of the two clinical

assessments are correlated, but we can live with it. The variables in the first set are:

$d_9$: Clinical rating of anxiety.

$d_{10}$: Clinical rating of depression.

Using security camera recordings of students eating lunch in the cafeteria (with everyone's permission, of course), the investigators record four social behaviour variables during a designated twenty-minute period. Correlated errors within this set are very likely.

$d_1$: Speaking time (not on phone).

$d_2$: Listening time (head turned toward speaker).

$d_5$: Number of smiles/laughs while not on cell phone solo[9].

$d_6$: Time solo on cell phone.

The following variables are obtained from school records. Measurement errors may not be correlated within this set, but we will be conservative, and assume they might be. In any case, it wll be testable.

$d_3$: Grade point average last academic session.

$d_4$: Attendance last academic session.

$d_7$: Hours per week playing school sports.

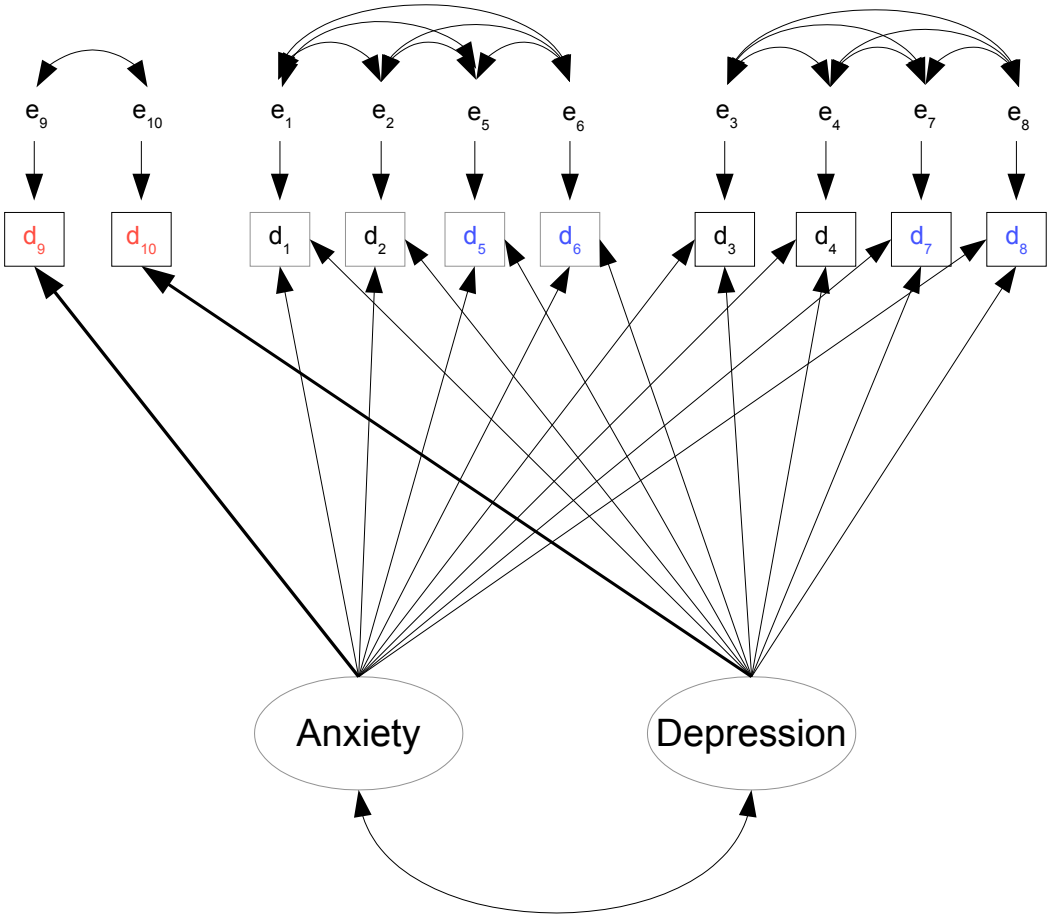
$d_8$: Hours per week spent on extra-curricular activities, not including school sports.

Comparing the variable numbering and colour coding to Figure 3.8, it can be seen that two blue variables ($d_5$ and $d_6$) have been grouped with the social behaviour variables, and the other two blue ones ($d_7$ and $d_8$) have been grouped with the school record variables. The flexibility of the reference variable rule has been exploited to assemble a model that makes substantive sense, and still has identifiable parameters because it's a special case of what's allowed. The result is the model of Figure 3.9. This is a good way to apply the reference variable rule in practice. The proof requires three sets of observed variables, each with as many observed variables as there are factors, and it allows an additional set with as many variables as you like. But in practice, one may have an arbitrary number of variable sets, each with error terms correlated only within the set — provided the following conditions are met.

- One set consists of a reference variable for each factor.
- Two or more of the other sets of variables have at least as many variables as there are factors.

[9]If two people are looking at a phone together, it's not "solo," and if they smile or laugh it would be counted

Figure 3.9: Model for the student mental health example (Example 3.4)




Again, the sets of observed variables are defined by having error terms that are correlated with one another, and uncorrelated with the error terms of variables in the other sets. The uncorrelated error terms are to be justified by specific features of the research design. This is both an opportunity and an obligation.

The reference variable rule is much stronger than the three-variable rules (also called three-indicator rules) given in other textbooks I have seen. For example, in Bollen's classic text [10] the "three-indicator" rule on p. 244 is exactly our three-variable rule (Rule 2b). All the observed variables are reference variables, and the covariance matrix of the error terms is diagonal. The result is a very restrictive model like the one in Figure 3.4, where observable variables can be influenced by only one factor. Surely it is better to *hypothesize* that certain factor loadings are zero and then test the hypothesis, than to simply assume that they are zero. Of course the assumption of independent errors is hard to justify as well, for most data sets. The reference variable rule is a welcome alternative.

**Overfitting** There is potential for abuse here. Suppose that as usual, data are collected without much thought to the confirmatory factor analysis model that will be fit. The error terms all could be correlated; who knows? All the factors could potentially affect all the observable variables; who knows? So the data analyst (who knows about the reference variable rule) picks some variables to be reference variables for the factors, assumes all the error terms to be independent, and runs the software. The model does not fit. So he picks some different variables as reference variables, and also semi-arbitrarily groups the observable variables into clusters, allowing non-zero covariance between error terms within a cluster. Now the fit is a lot better. The chi-squared test for lack of fit might even be non-significant. If it is still significant and the investigator keeps trying different combinations, then sooner or later, one of the models will almost surely fit the data. It is sort of like ordinary p-hacking[10] in reverse. The data analyst keeps trying different things until the result is *not* statistically significant.

Has something real been discovered, or is it just an exploitation of random features of the data? The boundary between data snooping and legitimate exploratory data analysis is often fuzzy, and this is no exception. The solution, if you engage in this kind of practice, is replication and cross-validation. An example will be given in Section 3.8.

## 3.5 More Identification Rules

Combining the two-variable rule for unstandardized factors (page 288) with the equivalence rule yields

**Rule 2d: Two-variable Rule** The parameters of a factor analysis model are identifiable provided

- There are at least two factors.
- There are at least two reference variables for each factor.
- For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading of at least one reference variable is equal to one.
- Each factor has a non-zero covariance with at least one other factor.
- Errors are independent of one another and of the factors.

The two-variable rule requires at least two factors, each with two reference variables. In practice, factors that influence only two observable variables are often part of a larger system, and there might be only one such factor in the model. The following shows how a factor with two reference variables can be combined with a model whose parameters have already been identified in some other way.

[10] Simonsohn et al. [59] deserve credit for the catchy term "*p*-hacking. I do not necessarily endorse their work on the "*p*-curve."

**Rule 2e: Two-variable Addition Rule** A factor with just two reference variables may be added to a measurement model whose parameters are identifiable, and the parameters of the combined model will be identifiable provided

- The errors for the two additional reference variables are independent of one another and of the error terms already in the model.
- For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading of at least one reference variable is equal to one.
- In the existing model with identifiable parameters,
  - There is at least one reference variable for each factor, and
  - At least one factor has a non-zero covariance with the new factor.

The proof of this rule will be given for standardized factors; the equivalence rule says that it also applies when a factor loading is set to one for each factor. Assume that there are already $p$ factors and $k$ observable variables in the model. The additional factor is $F_{p+1}$, and its reference variables are $d_{k+1}$ and $d_{k+2}$.

In the existing model, there is a factor that has non-zero covariance with $F_{p+1}$. Without loss of generality, label this factor $F_1$, and let its reference variable be $d_1$. We have

$$\begin{aligned} d_1 &= \lambda_1 F_1 + e_1 \\ d_{k+1} &= \lambda_{k+1} F_{p+1} + e_{k+1} \\ d_{k+2} &= \lambda_{k+2} F_{p+1} + e_{k+2}. \end{aligned} \tag{3.26}$$

The new parameters that need to be identified are $\lambda_{k+1}$, $\lambda_{k+2}$, $\omega_{k+1}$, $\omega_{k+2}$, and the covariances between the existing factors and the new factor: $\phi_{j,p+1}$ for $j = 1, \ldots, p$.

The covariance matrix of $\begin{pmatrix} d_1 \\ d_{k+1} \\ d_{k+2} \end{pmatrix}$ is

$$\begin{pmatrix} \sigma_{1,1} & \sigma_{1,k+1} & \sigma_{1,k+2} \\ & \sigma_{k+1,k+1} & \sigma_{k+1,k+2} \\ & & \sigma_{k+2,k+2} \end{pmatrix} = \begin{pmatrix} \lambda_1^2 + \omega_1 & \lambda_1\lambda_{k+1}\phi_{1,p+1} & \lambda_1\lambda_{k+2}\phi_{1,p+1} \\ & \lambda_{k+1}^2 + \omega_{k+1} & \lambda_{k+1}\lambda_{k+2} \\ & & \lambda_{k+2}^2 + \omega_{k+2} \end{pmatrix}.$$

Since the signs of $\lambda_1$ and $\lambda_{k+1}$ are known, the sign of $\phi_{1,p+1}$ can be determined from $\sigma_{1,k+1}$. Also, note that since $\lambda_1$ is already identified, it may used along with the $\sigma_{i,j}$ to solve for new parameters. Then,

$$\frac{\sigma_{1,k+1}\sigma_{1,k+2}}{\sigma_{k+1,k+2}} = \frac{\lambda_1^2\lambda_{k+1}\lambda_{k+1}\phi_{1,p+1}^2}{\lambda_{k+1}\lambda_{k+2}} = \lambda_1^2\phi_{1,p+1}^2.$$

Assuming $\lambda_1$ and $\lambda_{k+1}$ are positive (which they can always be, by naming the factors appropriately), $\phi_{1,p+1} = sign(\sigma_{1,k+1})\sqrt{\frac{\sigma_{1,k+1}\sigma_{1,k+2}}{\lambda_1^2\sigma_{k+1,k+2}}}$. Since $\phi_{1,p+1}$ is now identified, it can be

used to solve for other parameters, and

$$\begin{aligned}
\lambda_{k+1} &= \frac{\sigma_{1,k+1}}{\lambda_1 \phi_{1,p+1}} \\
\lambda_{k+2} &= \frac{\sigma_{1,k+2}}{\lambda_1 \phi_{1,p+1}} \\
\omega_{k+1} &= \sigma_{k+1,k+1} - \lambda_{k+1}^2 \\
\omega_{k+2} &= \sigma_{k+2,k+2} - \lambda_{k+2}^2.
\end{aligned}$$

To identify the covariances of the other factors with $F_{p+1}$, place the primary reference variables for those factors into positions $2, \ldots, p$ of the covariance matrix of observable variables. Then, for $j = 2, \ldots, p$,

$$cov(d_j, d_{k+1}) = \sigma_{j,k+1} = \lambda_j \lambda_{k+1} \phi_{j,p+1} \implies \phi_{j,p+1} = \frac{\sigma_{j,k+1}}{\lambda_j \lambda_{k+1}}.$$

This establishes the two-variable addition rule.

**The Combination Rule** The two-variable addition rule reflects how parameter identifiability is usually established in practice for big measurement models. Parts of the model are identified, and then they are combined with other factors and variables to produce larger sub-models whose parameters are identifiable. Then the sub-models are combined. The combination rule says that sub-models with identifiable parameters may be combined, provided that the error terms of the two models have zero covariance.

**Rule 2f: Combination Rule** Suppose that two factor analysis models are based on non-overlapping sets of observable variables from the same data set, and that the parameters of both models are identifiable. The two models may be combined into a single model provided that the error terms of the first model are independent of the error terms in the second model. The additional parameters of the combined model are the covariances between the two sets of factors. These are all identifiable, except possibly on a set of volume zero in the parameter space.

**Proof.** Let the first model have $p_1$ factors and $k_1$ observable variables, and let the second model have $p_2$ factors and $k_2$ observable variables. Separate the first set of observable variables into two subsets, with $p_1$ variables in the first subset, and $k_1 - p_1$ variables in the other subset. Do the same thing for the other model. The criteria for separating the variables into subsets will be described presently. The model equations are now

$$\begin{aligned}
\mathbf{d}_1 &= \mathbf{\Lambda}_1 \mathbf{F}_1 + \mathbf{e}_1 \\
\mathbf{d}_2 &= \mathbf{\Lambda}_2 \mathbf{F}_1 + \mathbf{e}_2 \\
\mathbf{d}_3 &= \mathbf{\Lambda}_3 \mathbf{F}_2 + \mathbf{e}_3 \\
\mathbf{d}_4 &= \mathbf{\Lambda}_4 \mathbf{F}_2 + \mathbf{e}_4
\end{aligned}$$

The matrix of factor loadings $\mathbf{\Lambda}_1$ is $p_1 \times p_1$. The variables in $\mathbf{d}_1$ are selected to ensure that $\mathbf{\Lambda}_1$ has an inverse. The variables in $\mathbf{d}_3$ are selected so that $\mathbf{\Lambda}_3$ has an inverse. Suppose it is impossible to select a subset of observable variables so that $\mathbf{\Lambda}_1$ has an inverse. If so, the columns of the *combined* matrix of factor loadings for the first model are linearly dependent. This holds only a set of volume zero in the parameter space. The same applies to the second model.

In the combined model, the only new parameters are contained in the $p_1 \times p_2$ matrix $cov(\mathbf{F}_1, \mathbf{F}_2)$, which will be denoted by $\mathbf{\Phi}_{12}$. We have

$$\begin{aligned} cov(\mathbf{d}_1, \mathbf{d}_3) &= cov(\mathbf{\Lambda}_1\mathbf{F}_1 + \mathbf{e}_1, \mathbf{\Lambda}_3\mathbf{F}_2 + \mathbf{e}_3) \\ &= \mathbf{\Lambda}_1 cov(\mathbf{F}_1, \mathbf{F}_2)\mathbf{\Lambda}_3^\top + \mathbf{O} + \mathbf{O} + \mathbf{O} \\ &= \mathbf{\Lambda}_1\mathbf{\Phi}_{12}\mathbf{\Lambda}_3^\top \end{aligned}$$

Since the matrices $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_3$ are already identified, they may be used to solve for $\mathbf{\Phi}_{12}$. Denoting $cov(\mathbf{d}_1, \mathbf{d}_3)$ by $\mathbf{\Sigma}_{13}$,

$$\begin{aligned} \mathbf{\Lambda}_1^{-1}\mathbf{\Sigma}_{13}\left(\mathbf{\Lambda}_3^\top\right)^{-1} &= \mathbf{\Lambda}_1^{-1}\mathbf{\Lambda}_1\mathbf{\Phi}_{12}\mathbf{\Lambda}_3^\top\left(\mathbf{\Lambda}_3^\top\right)^{-1} \\ &= \mathbf{\Phi}_{12}, \end{aligned}$$

completing the proof.

Note that if the factor analysis sub-models have been identified using any of the rules given so far in this chapter, then there is at least one reference variable for each factor. In this case, $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_3$ are diagonal matrices with non-zero diagonal elements, and both inverses exist. If factor loadings have been set to one in the surrogate models, then $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_3$ are identity matrices. In practice, the part of the combination rule that says "except possibly on a set of volume zero" does not come into play.

**The Extra Variables Rule** The extra variables rule says that if the parameters of a factor analysis model are identifiable, more observable variables may be added to the model without adding any new factors. Identifiability is preserved, provided that the error terms for the new variables are uncorrelated with the error terms for observable variables already in the model (as well as being uncorrelated with the factors, of course). It is okay for the error terms of the additional variables to be correlated with one another. Straight arrows with factor loadings on them may point from each existing factor to each new variable. It is not necessary to include all such arrows. There are no restriction on the factor loadings of the variables that are being added to the model.There are no restriction on the covariances of error terms for the new set of variables, except that they must not be correlated with error terms already in the model.

The extra variable rule and the reference variable rule have something in common. They both allow inclusion of an additional set of observable variables that are influenced by all factors, and whose error terms need not be independent. When both rules apply, the reference variable rule may be preferable, because it allows some covariances between the error terms of the new variables and the error terms of variables already in the model;

hence; it is more flexible. On the other hand, to add more observable variables to a non-standard model like the one in the political democracy Example 3.3.2, the extra variables rule is the way to go.

**Rule 2g: Extra Variables Rule** If the parameters of a factor analysis model are identifiable, then a set of additional observable variables (without any new factors) may be added to the model. In the path diagram, straight arrows with factor loadings on them may point from each existing factor to each new variable. Error terms for the new variables may have non-zero covariances with each other. If the error terms of the new set have zero covariance with the error terms of the initial set and with the factors, then the parameters of the combined model are identifiable, except possibly on a set of volume zero in the parameter space.

**Proof.** In the initial model, there are $p$ factors and $k_1$ observed variables. All parameters of the initial model are identifiable. The observed variables of the initial model are divided into two subsets, one with $p$ variables, and the other with $k_1 - p$ variables. The model equations are

$$\begin{aligned} \mathbf{d}_1 &= \mathbf{\Lambda}_1\mathbf{F} + \mathbf{e}_1 \\ \mathbf{d}_2 &= \mathbf{\Lambda}_2\mathbf{F} + \mathbf{e}_2 \\ \mathbf{d}_3 &= \mathbf{\Lambda}_3\mathbf{F} + \mathbf{e}_3, \end{aligned}$$

where the observed variables from the initial model are in $\mathbf{d}_1$ and $\mathbf{d}_2$, and the new variables are in $\mathbf{d}_3$. The variables in $\mathbf{d}_1$ are chosen so that the $p \times p$ matrix $\mathbf{\Lambda}_1$ has an inverse. This will be impossible if and only if the entire matrix of factor loadings for the initial model has columns that are linearly dependent, a condition that holds on a set of volume zero in the parameter space.

We have $cov(\mathbf{F}) = \mathbf{\Phi}$ and

$$cov\begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{pmatrix} = \left(\begin{array}{c|c|c} \mathbf{\Omega}_{11} & \mathbf{\Omega}_{12} & \mathbf{0} \\ \hline & \mathbf{\Omega}_{22} & \mathbf{0} \\ \hline & & \mathbf{\Omega}_{33} \end{array}\right).$$

The parameters to be identified are in the matrices $\mathbf{\Lambda}_3$ and $\mathbf{\Omega}_{33}$. The covariance matrix of the observable variables is

$$\begin{aligned} cov\begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \end{pmatrix} = \mathbf{\Sigma} &= \left(\begin{array}{c|c|c} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} & \mathbf{\Sigma}_{13} \\ \hline & \mathbf{\Sigma}_{22} & \mathbf{\Sigma}_{23} \\ \hline & & \mathbf{\Sigma}_{33} \end{array}\right) \\ &= \left(\begin{array}{c|c|c} \mathbf{\Phi} + \mathbf{\Omega}_{11} & \mathbf{\Phi}\mathbf{\Lambda}_2^\top & \mathbf{\Phi}\mathbf{\Lambda}_3^\top \\ \hline & \mathbf{\Lambda}_2\mathbf{\Phi}\mathbf{\Lambda}_2^\top + \mathbf{\Omega}_{22} & \mathbf{\Lambda}_2\mathbf{\Phi}\mathbf{\Lambda}_3^\top \\ \hline & & \mathbf{\Lambda}_3\mathbf{\Phi}\mathbf{\Lambda}_3^\top + \mathbf{\Omega}_{33} \end{array}\right). \end{aligned}$$

The parameters of the initial model are all identifiable, so they may be used to solve for the matrices $\mathbf{\Lambda}_3$ and $\mathbf{\Omega}_{33}$. This is straightforward:

$$\begin{aligned} \mathbf{\Lambda}_3 &= \mathbf{\Sigma}_{13}^\top \mathbf{\Phi}^{-1} \\ \mathbf{\Omega}_{33} &= \mathbf{\Sigma}_{33} - \mathbf{\Lambda}_3 \mathbf{\Phi} \mathbf{\Lambda}_3^\top \end{aligned} \quad \blacksquare$$

**The Error-free Rule** Starting with a factor analysis model with identifiable parameters, add an observable variable *to the factors*. Often it's an observed exogenous variable (like sex or a dummy variable for experimental condition) that is hypothesized to affect some of the latent variables in a general structural equation model. It is convenient to make such variables part of the latent variable model.

Suppose parameters of an existing factor analysis model with $p$ factors) are all identifiable. Add an observable scalar variable $x$ that is independent of the error terms, and may have non-zero covariances with the factors. Thinking of $x$ as an additional factor, we are adding a row (and column) to $\mathbf{\Sigma}$, and a row (and column) to $\mathbf{\Phi}$. There are $p+1$ additional parameters that need to be identified. One of these is the variance of $x$, which is obtained immediately as $\phi_{p+1,p+1} = \sigma_{k+1,k+1}$. The other new parameters are covariances between $x$ and the factors, which are identified as follows.

As in a couple of earlier proofs, the observed variables from the existing model are divided into two vectors $\mathbf{d}_1$ and $\mathbf{d}_2$, yielding the model equations

$$\begin{aligned} \mathbf{d}_1 &= \mathbf{\Lambda}_1 \mathbf{F} + \mathbf{e}_1 \\ \mathbf{d}_2 &= \mathbf{\Lambda}_2 \mathbf{F} + \mathbf{e}_2 \end{aligned}$$

where the variables in $\mathbf{d}_1$ are chosen so that the $p \times p$ matrix $\mathbf{\Lambda}_1$ has an inverse. This will be impossible if and only if the entire matrix of factor loadings for the existing model has columns that are linearly dependent, a condition that holds on a set of volume zero in the parameter space.

Let $\mathbf{\Sigma}_{x,d_1}$ denote the vector of covariances between $x$ and the variables in $\mathbf{d}_1$, and let $\mathbf{\Phi}_{x,F}$ denote the vector of covariances between $x$ and the other factors. $\mathbf{\Sigma}_{x,d_1}$ is part of the last row (column) of $\mathbf{\Sigma}$, and $\mathbf{\Phi}_{x,F}$ is part of the last row (column) of $\mathbf{\Phi}$. We have

$$\begin{aligned} \mathbf{\Sigma}_{x,d_1} &= cov(x, \mathbf{d}_1) \\ &= cov(x, \mathbf{\Lambda}_1 \mathbf{F} + \mathbf{e}_1) \\ &= \mathbf{\Lambda}_1 cov(x, \mathbf{F} + cov(x, \mathbf{e}_1)) \\ &= \mathbf{\Lambda}_1 \mathbf{\Phi}_{x,F} + \mathbf{0}, \end{aligned}$$

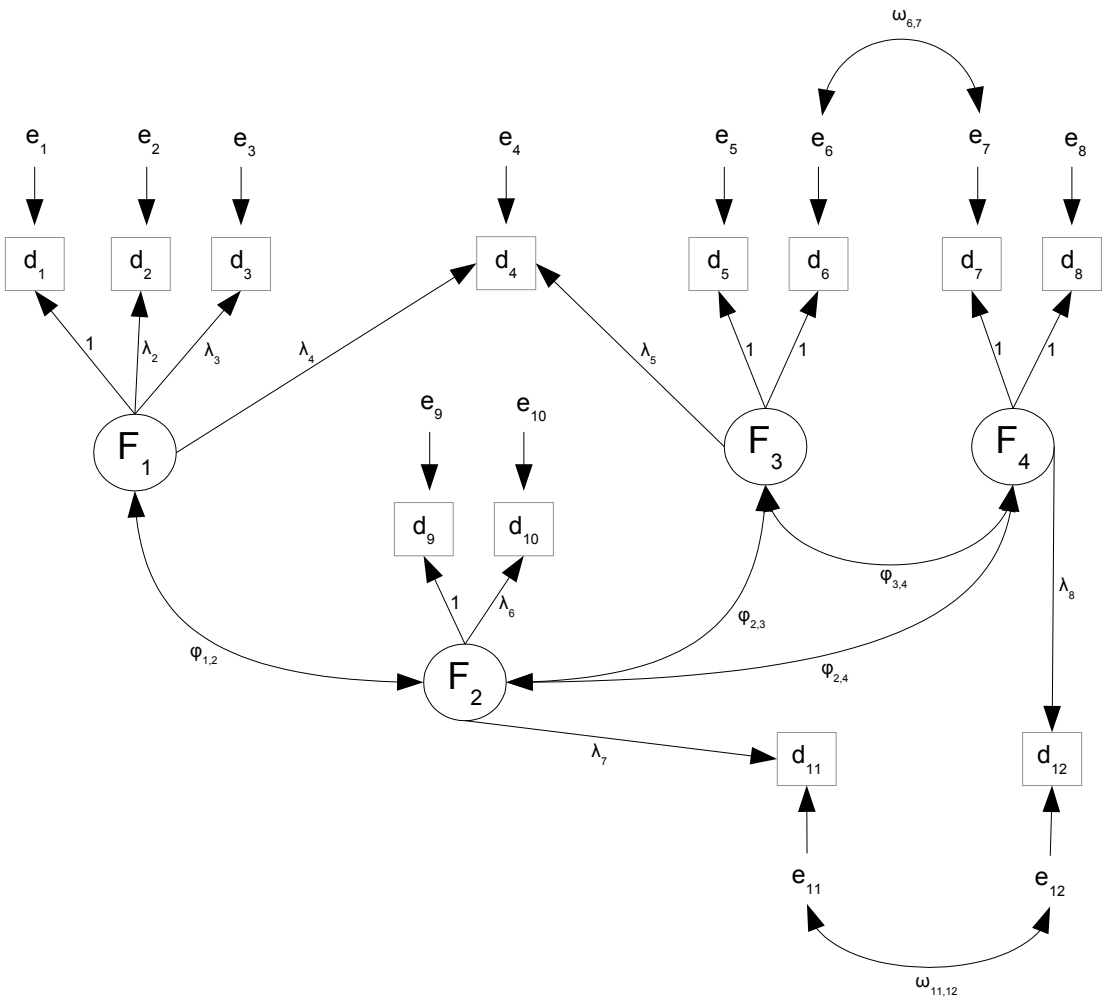
so that $\mathbf{\Phi}_{x,F} = \mathbf{\Lambda}_1^{-1} \mathbf{\Sigma}_{x,d_1}$. Since $\mathbf{\Lambda}_1$ is already identified, this completes the proof of the error-free rule. The rule will be stated as it applies to a vector of new observed variables.

**Rule 2h: The Error-free Rule** A set of observable variables may be added to the factors of a measurement model whose parameters are identifiable, provided that the new observed variables are independent of the error terms that are already in the model. The parameters of the resulting model are identifiable, except possibly on a set of volume zero in the parameter space.

## 3.6 Putting the Rules Together

Figure 3.10 shows a big, hairy confirmatory factor analysis model. Trying to establish

Figure 3.10: A Confirmatory Factor Analysis Model




identifiability by solving covariance structure equations would be a huge clerical task; instead, we will use the identifiability rules. See Appendix D for a collection of the identifiability rules in outline form.

There are twelve observable variables, so that $\mathbf{\Sigma}$ has $12(12+1)/2 = 78$ unique elements. The number one on some of the straight arrows tells us that this is a surrogate model in which at least one factor loading has been set to one for each factor. Counting parameters, there are 4 variances of the factors (denoted $\phi_{j,j}$), 4 possibly non-zero covariances between factors, and 7 factor loadings that are not fixed to the value one. There are 12 error variances (denoted $\omega_{j,j}$), and 2 possibly non-zero covariances between error terms. In all,

that's 78 covariance structure equations in $4+4+7+12+2 = 29$ unknown parameters. Because there are more covariance structure equations than parameters, the model passes the test of the parameter count rule, and identifiability cannot be ruled out.

We will establish identifiability in two ways, first without using the reference variable rule, and then using it.

**Without using the reference variable rule** The strategy will be to apply the rules to parts of the model, and then put the sub-models together. First, consider the sub-model involving $F_1$, $d_1$, $d_2$ and $d_3$. Its parameters are identifiable by the three-variable rule, provided that $\lambda_2$ and $\lambda_3$ are both non-zero. This could be verified empirically by testing $H_0 : Corr(d_1, d_2) = 0$ and $H_0 : Corr(d_1, d_3) = 0$. We have identified six parameters: $\phi_{1,1}$, $\lambda_2$, $\lambda_3$, $\omega_{1,1}$, $\omega_{2,2}$ and $\omega_{3,3}$.

Next, look at the part involving $F_3$, $F_4$, and $d_5$ through $d_8$. The double measurement rule covers this, including the covariance between $e_6$ and $e_7$. Just consider $d_6$ and $d_7$ to be part of the same "set" of measurements, perhaps conducted at the same time by the same personnel. This identifies eight more parameters: $\phi_{3,3}$, $\phi_{4,4}$, $\phi_{3,4}$, $\omega_{5,5}$, $\omega_{6,6}$, $\omega_{7,7}$, $\omega_{8,8}$ and $\omega_{6,7}$.

Now put the two sub-models together using the combination rule. Notice that the variables $d_4$ and $d_{12}$ are not included yet; they are being saved for later. Also, the zero covariance between $F_1$ and the other factors presents no obstacle. No new parameters have been identified in this case, but merging the sub-models helps with the next step.

The next step is to add the part involving $F_2$, $d_9$ and $d_{10}$ to the combined sub-model. The two-variable addition rule allows this, provided $\phi_{1,2}$, $\phi_{2,3}$ and $\phi_{2,4}$ are not all zero. According to the model, if $\phi_{1,2}$ were zero, then $d_9$ and $d_{10}$ would be uncorrelated with $d_1$, $d_2$ and $d_3$; this is testable. The conditions $\phi_{2,3} \neq 0$ and $\phi_{2,4} \neq 0$ could be verified in a similar way, and only one of the three covariances with $F_2$ needs to be non-zero for the two-variable addition rule to apply. In this way, seven more parameters are identified: $\phi_{2,2}$, $\phi_{1,2}$, $\phi_{2,3}$, $\phi_{2,4}$, $\lambda_6$, $\omega_{9,9}$ and $\omega_{10,10}$.

At this point, we have a (big) sub-model whose parameters are all identifiable, and which includes all the factors. Now use the extra variables rule to add the remaining observable variables $d_4$, $d_{11}$ and $d_{12}$, quickly checking that their error terms are not correlated with any of the error terms already in the model. Eight more parameters are identified: $\lambda_4$, $\lambda_5$, $\lambda_7$, $\lambda_8$, $\omega_{4,4}$, $\omega_{11,11}$, $\omega_{12,12}$ and $\omega_{11,12}$.

That does it. There were 29 parameters to identify, and we identified $6+8+7+8 = 29$. Notice how, at several points in the argument, empirical tests were proposed to verify that the true parameter vector was in a region of the parameter space where the parameters involved were identifiable. One can extend this dual strategy of identification checking and empirical testing, by testing sub-models for fit, and then testing fit again as the sub-models are combined. This way, if the final model does not fit the data, you probably will have a good idea why.

**Checking identifiability using the reference variable rule** The rule requires that the number of observable variables be at lest three times the number of factors. The model

has four factors and twelve observable variables, so the first requirement is satisfied — just barely. The next requirement is that every factor have at least one reference variable. A quick glance verifies this condition. In fact, every factor has at least two reference variables. At least one reference variable for every factor has a factor loading of one, so this is a nice unstandardized surrogate model; the third condition of the rule is satisfied.

The model has only two non-zero covariances between error terms, so as long as $d_6$ and $d_7$ go in the same set of variables and $d_{11}$ and $d_{12}$ do in the same set, all the parameters are identifiable except possibly on a set of volume zero in the parameter space. Let's take a closer look at this issue.

Referring back to Model (3.25) on page 306, observe that the lowerdimensional set of parameter values where identifiability fails is the set where the square sub-matrices $\mathbf{\Lambda}_2$ and $\mathbf{\Lambda}_3$ do not have inverses. Mathematically, this could happen just because of the values that the factor loadings happen to have, and there's really nothing we can do about it. More concerning would be if it happened because of definite zeros in a model we more or less believe, like the model of Figure 3.10. To check this, I wrote down the factor matrix, shown in (3.27). The rows are re-arranged (there is more than one way to do it) so that $\mathbf{\Lambda}_2$ and $\mathbf{\Lambda}_3$ both have inverses, provided that most of the $\lambda_j$ are non-zero. The only exception I see is that $\lambda_4$ could be zero. The other way of proving identifiability (without the reference variable rule) also requires that most of the $\lambda_j$ be non-zero.

$$\left(\begin{array}{c} \mathbf{\Lambda}_1 \\ \hline \mathbf{\Lambda}_2 \\ \hline \mathbf{\Lambda}_3 \end{array}\right) = \begin{array}{c|cccc} & F_1 & F_2 & F_3 & F_4 \\ \hline d_1 & 1 & 0 & 0 & 0 \\ d_9 & 0 & 1 & 0 & 0 \\ d_6 & 0 & 0 & 1 & 0 \\ d_7 & 0 & 0 & 0 & 1 \\ \hline d_2 & \lambda_2 & 0 & 0 & 0 \\ d_{10} & 0 & \lambda_6 & 0 & 0 \\ d_5 & 0 & 0 & 1 & 0 \\ d_8 & 0 & 0 & 0 & 1 \\ \hline d_4 & \lambda_4 & 0 & \lambda_5 & 0 \\ d_3 & \lambda_3 & 0 & 0 & 0 \\ d_{11} & 0 & \lambda_7 & 0 & 0 \\ d_{12} & 0 & 0 & 0 & \lambda_8 \end{array} \tag{3.27}$$

Most of the time, it is not necessary to write down the complete factor matrix in order to verify that the reference variable rule applies — but it's quite informative here. The main lesson is that while the model of Figure 3.10 seems to have a lot of arrows, it is actually a very sparse special case of the model (Model 3.25) that underlies the reference variable rule. In (3.27), 25 factor loadings are set to zero or one, while they are unconstrained under Model (3.25). These all represent testable null hypotheses[11] rather than assumptions. In addition, Figure 3.10 has only two potentially non-zero covariances between error terms, while Model (3.25) allows $3 \times p(p-1) = 36$. In all, that's $25 + 34 = 59$ ways in which the

[11] A factor loading will equal one under this surrogate model if and only if two factor loadings are equal under the original model.

model of Figure 3.10 might fail to fit the data, while Model (3.25) could fit very well.

This raises a question. What should be done if the model does not fit? If one is using the reference variable rule, the answer is pretty obvious. Fit a model with the factor loadings in $\mathbf{\Lambda}_2$ and $\mathbf{\Lambda}_3$ unconstrained, and test the 25 null hypotheses with $z$-tests. Any null hypothesis that is rejected points to a constraint on the parameter values that is contributing to the lack of fit. If the model with unconstrained factor loadings still does not fit, a second line of attack is to start testing hypotheses about covariances between error terms. This is tough to do in an honest way without more information about how the data were collected. The observable variables may not naturally divide themselves into subsets whose error terms can be assumed independent, because the study may not have been planned with this in mind.

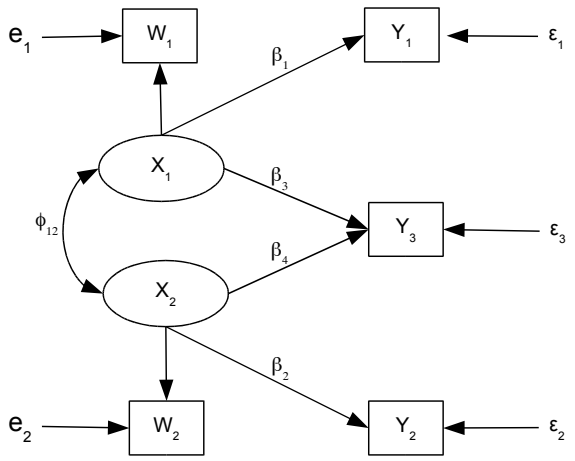
This is all possible with the reference variable rule in hand. Without the rule, it would be hard to know what to do. The only real choice would be to start guessing and trying to solve equations. Good luck.

### More examples of applying the rules

**Example 3.6.1** *A latent variable regression*

This example is based on the fact that a regression model with latent explanatory variables and observed response variables may be viewed as a confirmatory factor analysis model. Figure 3.11 reproduces Figure 19 on page 117. The reference variable rule does not apply

Figure 3.11: Regression with latent explanatory variables as a confirmatory factor analysis (Reproduction of Figure 19)

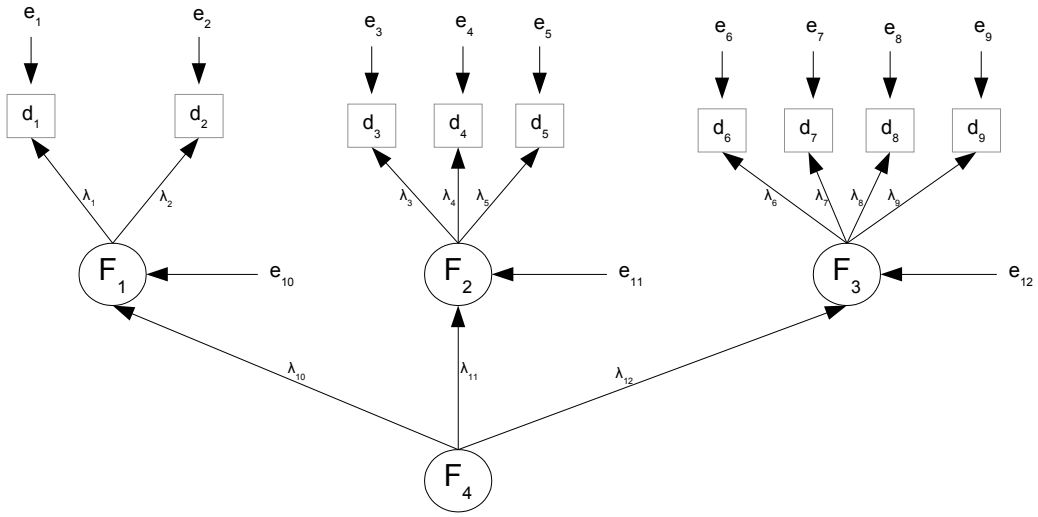



because there are two factors and only $5 < 6$ observable variables, but the parameters are immediately identifiable by the two-variable rule, except at points in the parameter space where $\phi_{1,2} = 0$. Detailed calculations like the ones in Chapter 0 are usually unnecessary if you know some identifiability rules.

**Example 3.6.2** *A second-order factor analysis*

Figure 3.12 shows a simple *second-order factor analysis* model. The idea behind higher order factor analysis is that the observed variables reflect a set of unobservable factors, and those factors in turn reflect the operation of another set of factors at a deeper level. In principle, there could be third-order factors influencing the second-order factors, and

Figure 3.12: Second-order factor analysis




so on.

In a higher-order factor analysis model, the higher-order factors (second order and above) have no direct influence on the observed variables. Perhaps surprisingly, it is still possible to apply the identifiability rules we have. In Figure 3.12, none of the factor loadings is explicitly set to one, so assume the factors are standardized, and that the sign of one factor loading is known for each factor. This includes the set $\lambda_{10}$, $\lambda_{11}$ and $\lambda_{12}$.

To check identifiability, adopt a two-stage approach. First, look at the first-order factors $F_1$, $F_2$ and $F_3$. Imagine curved, double headed arrows connecting them. The covariances will be determined by $\lambda_{10}$, $\lambda_{11}$ and $\lambda_{12}$, but ignore $F_4$ and the straight arrows from $F_4$ to the first-order factors for now.

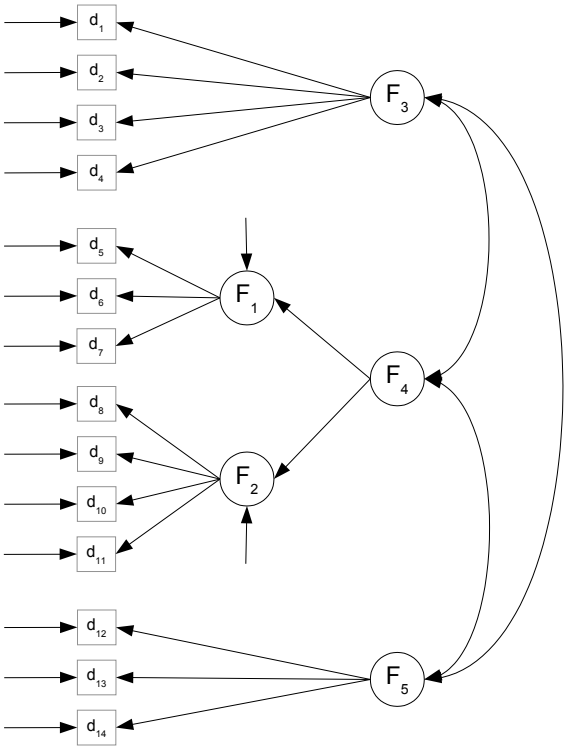
It's clear that the system involving $F_2$, $F_3$ and $d_3$ through $d_8$ is identified by the three-variable rule, and then the system involving $F_1$, $d_1$ and $d_2$ can be brought in with the two-variable addition rule. The factor loadings $\lambda_1$ through $\lambda_9$ and the error variances $\omega_1$ through $\omega_9$ have all been identified, as have the covariances (correlations) of $F_1$, $F_2$ and $F_3$.

Now think of $F_1$ through $F_3$ as observed variables, and let their correlation matrix (identified by the argument above) play the role of $\boldsymbol{\Sigma}$. By the three-variable rule, the factor loadings $\lambda_{10}$ through $\lambda_{12}$ are all identifiable, provided they are non-zero. That's it, and that's how it goes in general. The stages in the identifiability proof follow the stages in the model.

**Example 3.6.3** *Another higher-order model*

Figure 3.13 shows another confirmatory factor analysis model. This one is a sort of hybrid, with both first-order and second-order features. A picture like this could arise

Figure 3.13: Mixed first-order and second-order factor analysis




quite naturally in the course of model development. The investigator has several factors in mind, and several observed variables designated to measure each one. For example in Figure 3.13, $d_1$ through $d_4$ could be measures of left-right political orientation, $d_5$ through $d_{11}$ could be measures of academic performance (which would be called "intelligence" by some), and $d_{12}$ through $d_{14}$ could be measures of self-esteem. To check uni-dimensionality, the investigator carries out separate exploratory factor analyses (yes, *exploratory*) on the three subsets of observable variables. If everything is okay, a single-factor model should fit each one.

It works out okay for political orientation and self esteem, but for $d_5$ through $d_{11}$, two factors are required. After rotation, it looks like $d_5$ through $d_7$ load primarily on one factor, while $d_8$ through $d_{11}$ load on the other. The first set of variables depend on solving puzzles and math problems, while the second set depend on knowing the definitions of words and on reading a brief passage and then answering questions about it. One could call these factors "Math" and "Verbal," and nobody would argue.

Unfortunately, the factors are orthogonal, because it's a generic exploratory factor analysis model. It may fit the data, but only because of the arrows running from $F_1$ to

$d_8$ through $d_{11}$, and from $F_2$ to $d_5$ through $d_7$. This crossover pattern is not identifiable (the extra variables rule does not apply), and it's incompatible with the path diagram in Figure 3.13. Also, separate Math and Verbal factors do not accord with the investigator's theory or research questions, which are about a single thing called "intelligence." Figure 3.13 shows a really nice solution, which allows the Math and Verbal factors to be somewhat distinct, but correlated because they both reflect a second-order factor — and that factor is what the investigator wants to study.

Two comments are in order. First, I did not think of this cute data analysis trick. I saw it in a low-grade empirical research paper, and I am still searching for the source of the idea so I can give proper credit. Second, the forgoing discussion points out the fact that like most statistical methods, confirmatory factor analysis is often used in an exploratory way. In practice, the user will try quite a few models until finding one that fits the data adequately, and then carry out a boatload of statistical tests. In the end, only one model and a few of the tests will be reported, and the discussion will make it seem like it was planned all along. There is lots of opportunity for overfitting, and for apparent findings that actually reflect coincidences in the data. The solution is to replicate the results on a second, independent set of data. Without this kind of cross-validation, the so-called "conclusions" should be treated as data-driven hypotheses. Again, this situation is not limited to confirmatory factor analysis and structural equation models. It is true of most statistical applications.

Now consider identifiability for Figure 3.13. Parameters of the first-order system involving $F_1$ and $F_2$ (with a curved, double-headed arrow between the factors) and $d_5$ through $d_{11}$ are identifiable by the three-variable rule. Now bring $d_1$ through $d_4$ and $d_{12}$ through $d_{14}$ into the first-order model, using the error-free rule. That is, treat these observable variables as factors that are measured without error. The result is an ordinary second-order factor analysis model in which the second-order factors are $F_3$, $F_4$ and $F_5$. The system involving $F_3$ and $F_5$ is identified[12] by the three-variable rule. The system involving $F_1$, $F_2$ and $F_4$ is then brought in with the two-variable addition rule.

All the parameters are identifiable except on a set of volume zero in the parameter space, so it's mission accomplished — sort of. In this case, the set of volume zero where identifiability fails happens to include some interesting points, namely the points where $Cov(F_3, F_4) = 0$ and $Cov(F_4, F_5) = 0$. At least one of these covariances needs to be nonzero in order for the two-variable addition rule to work in the last stage of the proof.

The whole point of the study is probably the connections between $F_3$, $F_4$ and $F_5$. If the investigator tries to test the null hypothesis that all three covariances are zero using a likelihood ratio test, the process will fail. It will be impossible to fit the restricted model, because the likelihood function will have a non-unique maximum on an infinte connected set. If in reality part of the null hypothesis is true, with both $Cov(F_3, F_4) = 0$ and $Cov(F_4, F_5) = 0$, then there could easily be numerical difficulties in fitting the *unrestricted* model. Fortunately, the model says that $Cov(F_3, F_4) = 0$ if and only if the matrix of covariances between $(d_1, d_2, d_3, d_4)^\top$ and $(d_5, \ldots, d_{11})^\top$ consists only of zeros. This can

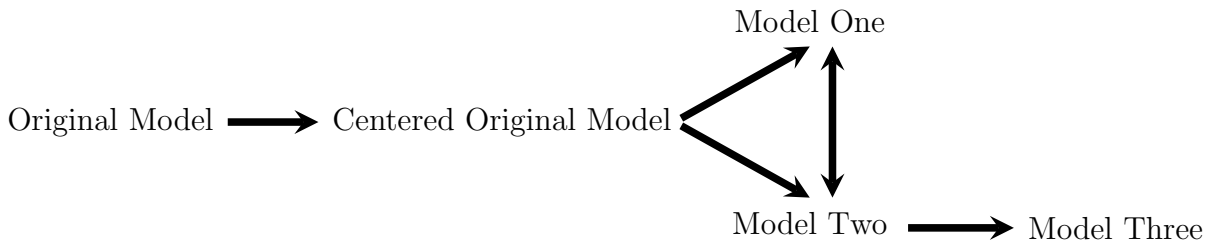
[12]That is, the parameters are a function of the variances and covariances of the first-order factors, which in turn are functions of the variances and covariances of the observable variables.

be tested with off-the-shelf canonical correlation methods (see R's `CCP` package), and $Cov(F_4, F_5) = 0$ can be diagnosed in a similar way.

## 3.7 Standardized Observed Variables

Standardizing the observed variables is familiar from exploratory factor analysis; see Chapter 2. In confirmatory factor analysis, it is a change of variables that leads to another level of surrogate model, beyond the standard choices of standardizing factors or setting factor loadings to one. Figure 3.14 shows the most common situation. Standardizing

Figure 3.14: Standardizing the Observed Variables




the observed variables just means dividing them by their standard deviations, since they already have expected value zero under the common centered surrogate models. While this operation may be applied at any point in the re-parameterization process, it is most commonly applied to a model with standardized factors (Model Two). The surrogate model with both standardized factors and standardized observed variables will be called Model Three.

Consider the model equations $\mathbf{d} = \mathbf{\Lambda F} + \mathbf{e}$, with the double primes (if any) hidden, and not bothering to separate the data vector into $\mathbf{d}_1$ and $\mathbf{d}_2$. Let $\mathbf{W} = dg(\mathbf{\Sigma})$, where as usual, $\mathbf{\Sigma} = cov(\mathbf{d})$. Then,

$$\begin{aligned} \mathbf{z} &= \mathbf{W}^{-1/2}\mathbf{d} \\ &= \mathbf{W}^{-1/2}(\mathbf{\Lambda F} + \mathbf{e}) \\ &= (\mathbf{W}^{-1/2}\mathbf{\Lambda})\mathbf{F} + (\mathbf{W}^{-1/2}\mathbf{e}) \\ &= \mathbf{\Lambda}'''\,\mathbf{F} + \mathbf{e}''', \end{aligned}$$

where

$$\mathbf{\Lambda}''' = \mathbf{W}^{-1/2}\mathbf{\Lambda} \quad \text{and} \quad cov(\mathbf{e}''') = \mathbf{\Omega}''' = \mathbf{W}^{-1/2}\mathbf{\Omega}\mathbf{W}^{-1/2}. \tag{3.28}$$

One thing to notice about standardizing the observed variables is that while it affects $\mathbf{\Lambda}$ and $\mathbf{\Omega}$, the covariances between factors in the matrix $\mathbf{\Phi}$ are unaffected. This is a fortunate, since observed variables are usually standardized only if the factors are standardized, and when the factors are standardized, covariances between factors equal correlations under

the original model. The nice interpretation is preserved – so at least, standardizing the observed variables does no harm.

It can also do some good. We will now see that when the observed variables are standardized, the factor loading for a reference variable is the correlation of the reference variable with the latent variable it measures, under the original model. Also, the variance of the error term (for all observed variables, not just reference variables) is the proportion of variance in that variable that is due to error — again, under the original model.

**Correlations between factors and their reference variables** Let $d_\ell$ be a reference variable for factor $j$, so that under the centered original model, $d_\ell = \lambda_{\ell,j} F_j + e_\ell$. Choosing explicitness over simplicity, we will employ the notation of Section 3.3.1, and use double primes to indicate quantities under Model Two, in which the factors have been standardized. We have

$$\begin{aligned} cov(F_j, d_\ell) &= cov(F_j, \lambda_{\ell,j} F_j + e_\ell) \\ &= cov(F_j, \lambda''_{\ell,j} F_j + e''_\ell) \\ &= \lambda_{\ell,j} cov(F_j, F_j) + cov(F_j, e_\ell) \\ &= \lambda_{\ell,j} \phi_{j,j}, \end{aligned}$$

so that

$$\begin{aligned} corr(F_j, d_\ell) &= \frac{\lambda_{\ell,j}\phi_{j,j}}{\sqrt{\phi_{j,j}}\sqrt{\sigma_{\ell,\ell}}} \\ &= \frac{\lambda_{\ell,j}\sqrt{\phi_{j,j}}}{\sqrt{\sigma_{\ell,\ell}}}. \end{aligned} \tag{3.29}$$

Consider the model with both $F_j$ and $d_\ell$ standardized. Recalling how we got there,

$$\begin{aligned} d_\ell &= \lambda_{\ell,j} F_j + e_\ell \\ &= \lambda_{\ell,j}\sqrt{\phi_{j,j}} \left( \frac{1}{\sqrt{\phi_{j,j}}} \right) F_j + e_\ell \\ &= \lambda''_{\ell,j} F''_j + e_\ell. \end{aligned}$$

Then standardizing $d_\ell$ as well,

$$\begin{aligned} z_\ell &= \frac{1}{\sqrt{\sigma_{\ell,\ell}}} \lambda''_{\ell,j} F''_j + \frac{1}{\sqrt{\sigma_{\ell,\ell}}} e_\ell \\ &= \lambda'''_{\ell,j} F''_j + e'''_\ell. \end{aligned}$$

Now un-wrap $\lambda'''_{\ell,j}$, the factor loading of the reference variable under this "completely standardized" model[13].

$$\begin{aligned} \lambda'''_{\ell,j} &= \frac{1}{\sqrt{\sigma_{\ell,\ell}}}\lambda''_{\ell,j} \\ &= \frac{1}{\sqrt{\sigma_{\ell,\ell}}}\lambda_{\ell,j}\sqrt{\phi_{j,j}}, \end{aligned}$$

which is exactly expression (3.29) for the correlation between the factor and its reference variable, under the original model. Squaring the factor loading yields the reliability of the reference variable — the proportion of variance in the reference variable that arises from the quantity it is measuring, and not error. It is always helpful when the parameters of a surrogate model correspond to something important about the original model.

**Uniqueness** As discussed in Chapter 2, the uniqueness of an observed variable is the proportion of its variance that comes from error (the unique factor) and not the common factors. For reference variables, the uniqueness is one minus the reliability.

Let $\boldsymbol{\lambda}_\ell$ denote row $\ell$ of the factor matrix $\boldsymbol{\Lambda}$ in the original model. This is the row corresponding to the observed variable $d_\ell$. If $d_\ell$ is a reference variable, $\boldsymbol{\lambda}_\ell$ has only one non-zero element, but that need not be the case here. If the observed variables are *not* standardized, $d_\ell = \boldsymbol{\lambda}_\ell\mathbf{F} + e_\ell$. When the observed variables are standardized (whether or not the factors are standardized as well), $z_\ell = \boldsymbol{\lambda}'''_\ell\mathbf{F} + e'''_\ell$, and

$$\begin{aligned} Var(z_\ell) &= cov(\boldsymbol{\lambda}'''_\ell\mathbf{F} + e'''_\ell) \\ &= \boldsymbol{\lambda}'''_\ell\boldsymbol{\Phi}\boldsymbol{\lambda}'''^{\top}_\ell + \omega'''_{\ell,\ell}. \end{aligned} \tag{3.30}$$

By (3.28), $\omega'''_{\ell,\ell} = \omega_{\ell,\ell}/\sigma_{\ell,\ell}$. It is exactly the uniqueness of $d_\ell$ under the original model. That is, it is the proportion of variance in the observed variable $d_\ell$ that comes from error (the unique factor) and not the common factors. For example, if the value of such a parameter is something like 0.85, it means that the variable in question is 85% noise. Uniqueness is worth estimating, and standardizing the observed variables makes the process more convenient.

**Reduction of the parameter space** Because $var(z_\ell = 1)$, expression (3.30) says that $\omega'''_{\ell,\ell} = Var(e'''_\ell) = 1 - \boldsymbol{\Lambda}'''_\ell\boldsymbol{\Phi}\boldsymbol{\Lambda}'''^{\top}_\ell$. That is, the variances of the error terms are functions of the other parameters in the model. The dimension of the parameter space has been reduced by $k$, the number of observed variables. We will now see that for almost all models used in practice, this reduction of the parameter space has no effect on identifiability or model fit.

[13]Some software, including lavaan, calls models and their estimates "completely" standardized when both the factors and the observable variables are standardized

**A modest assumption** As in 3.30, $d_\ell = \boldsymbol{\lambda}_\ell \mathbf{F} + e_\ell$ implies $Var(d_\ell) = \sigma_{\ell,\ell} = \boldsymbol{\lambda}_\ell \boldsymbol{\Phi} \boldsymbol{\lambda}_\ell^\top + \omega_{\ell,\ell}$. That is, the variances of the observed variables are something plus an $\omega_{\ell,\ell}$. Suppose that this is the only place in $\boldsymbol{\Sigma}$ where $\omega_{\ell,\ell}$ appears, and also suppose that for $\ell = 1, \ldots, j$, the error variance $\omega_{\ell,\ell}$ is not subject to any constraints, such as some of them being equal to one another or to other model parameters. This is typical of most models used in practice, and it leads to some useful conclusions.

**Identifiability** When solving covariance structure equations to prove identifiability, it is natural to set the diagonal elements of $\boldsymbol{\Omega}$ aside and solve for the other parameters first. If it works, one can then obtain the error variances by subtraction. When the observed variables are standardized, the whole process is the same except that the last step is omitted. This implies that the identifiability status of a model is not changed if the observed variables are standardized — given the "modest assumption" of the paragraph above.

**Equal diagonals** Recall the meaning of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. It's just the covariance matrix of the observable variables (that is, $\boldsymbol{\Sigma}$), written as a function of the model parameters $\boldsymbol{\theta}$.

As mentioned back on page 171, maximum likelihood estimation often proceeds by minimizing the objective function $g(\boldsymbol{\theta}) = tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}| - k$, which is equivalent to minimizing the minus log likelihood. The function $g(\boldsymbol{\theta})$ is a lot like a distance between $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ and $\widehat{\boldsymbol{\Sigma}}$[14]. Other things being equal, anything that brings $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ closer to $\widehat{\boldsymbol{\Sigma}}$ will reduce the value of $g(\boldsymbol{\theta})$. In particular, for any fixed values of the matrices $\boldsymbol{\Lambda}$ and $\boldsymbol{\Phi}$ (and regardless of the the off-diagonal elements of $\boldsymbol{\Omega}$), letting $\omega_{\ell,\ell} = \widehat{\sigma}_{\ell,\ell} - \boldsymbol{\lambda}_\ell \boldsymbol{\Phi} \boldsymbol{\lambda}_\ell^\top$ for $\ell = 1, \ldots, k$ will make the main diagonals of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ and $\widehat{\boldsymbol{\Sigma}}$ coincide, resulting in a lower value of $g(\boldsymbol{\theta})$. This also holds when $\boldsymbol{\Lambda} = \widehat{\boldsymbol{\Lambda}}$ and $\boldsymbol{\Phi} = \widehat{\boldsymbol{\Phi}}$. The conclusion is that for $\ell = 1, \ldots, k$, we have $\widehat{\sigma}_{\ell,\ell} = \widehat{\boldsymbol{\lambda}}_\ell \widehat{\boldsymbol{\Phi}} \widehat{\boldsymbol{\lambda}}_\ell^\top + \widehat{\omega}_{\ell,\ell}$. The right-hand side is a diagonal element of $\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$, so that $dg(\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})) = dg(\widehat{\boldsymbol{\Sigma}})$. Another way to express this is

$$\mathbf{W}(\widehat{\boldsymbol{\theta}}) = \widehat{\mathbf{W}}. \tag{3.31}$$

This equality will come in handy very shortly. Once again, it holds when the error variances $\omega_{\ell,\ell}$ appear only in the diagonal of $\boldsymbol{\Omega}$, and are otherwise unconstrained[15].

[14] It is non-negative, and it equals zero if and only if $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \widehat{\boldsymbol{\Sigma}}$. I'm not sure whether it obeys the triangle inequality. This gap makes the argument less rigorous.

[15] Of course the model implies some constraints on the $\omega_{\ell,\ell}$. Since they are variances, they are must be non-negative. Also, if the covariance matrix $\boldsymbol{\Omega}$ has any non-zero off-diagonal elements, the fact that it must be non-negative definite places additional limitations on the possible values of $\omega_{\ell,\ell}$. However, these constraints are not automatically enforced in a numerical search for the MLE, unless the user explicitly specifies inequality constraints. The result is that as the numerical optimization forces $dg(\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}}))$ toward $dg(\widehat{\boldsymbol{\Sigma}})$, an $\widehat{\omega}_{\ell,\ell}$ or two can easily become negative for some models and some data sets. This is the dreaded Heywood case (see p. 226). For a sufficiently large sample size, the consistency of maximum likelihood estimation guarantees that it cannot happen if the model is correct and the true parameter vector is in the interior of the parameter space. Negative variance estimates are a sign of poor model fit.

**Estimation** Parameter estimates for a model in which the observed variables are standardized may be obtained without re-fitting the model. The key is Expression (3.28), which is reproduced here for convenience. In most applications, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$ contain parameters from a surrogate model with standardized factors, so one could say they have invisible double primes.

$$\boldsymbol{\Lambda}''' = \mathbf{W}^{-1/2}\boldsymbol{\Lambda} \quad \text{and} \quad cov(\mathbf{e}''') = \boldsymbol{\Omega}''' = \mathbf{W}^{-1/2}\boldsymbol{\Omega}\mathbf{W}^{-1/2}.$$

It is tempting to just put hats on everything and invoke invariance, but you need to watch out. While $\widehat{\mathbf{W}} = dg(\widehat{\boldsymbol{\Sigma}})$ and $\widehat{\boldsymbol{\Sigma}}$ is an MLE, it's an MLE based on a generic multivariate normal model, not the same as the factor analysis model with $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$. What we really want is

$$\widehat{\boldsymbol{\Lambda}}''' = \mathbf{W}(\widehat{\boldsymbol{\theta}})^{-1/2}\widehat{\boldsymbol{\Lambda}} \quad \text{and} \quad \widehat{\boldsymbol{\Omega}}''' = \mathbf{W}(\widehat{\boldsymbol{\theta}})^{-1/2}\,\widehat{\boldsymbol{\Omega}}\,\mathbf{W}(\widehat{\boldsymbol{\theta}})^{-1/2}. \tag{3.32}$$

The distinction between $\widehat{\mathbf{W}}$ and $\mathbf{W}(\widehat{\boldsymbol{\theta}})$ does not matter when (3.31) holds, which is most of the time. Still, it's nice to know that lavaan uses (3.32). It took me a fair amount of work to verify this, because it's not that easy to come up with a model where (3.31) fails badly enough to have a noticeable effect.

To obtain standard errors and tests for a model with standardized observed variables, it is necessary to re-fit the model. There are two natural ways to proceed. The most obvious way is to literally standardize the observed variables; subtract off the sample means and then divide by the sample standard deviations[16]. The same results may be obtained by analyzing the sample correlation matrix rather than the covariance matrix. This will be illustrated in Section 3.8.

**Testing goodness of fit** When Expression (3.31) holds, standardizing the observed variables has no effect on the likelihood ratio test for model fit. This is established in the following theorem.

**Theorem 3.1** *For a centered confirmatory factor analysis model, let $\boldsymbol{\theta}$ denote the parameter vector, and let $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\theta})$ denote the $k \times k$ variance covariance matrix of the observable variables. The unique MLE of $\boldsymbol{\theta}$ is $\widehat{\boldsymbol{\theta}}$, and the sample variance-covariance matrix of the observable variables is $\widehat{\boldsymbol{\Sigma}}$. Let $\widehat{\mathbf{W}} = dg(\widehat{\boldsymbol{\Sigma}})$ and $\mathbf{W}(\widehat{\boldsymbol{\theta}}) = dg(\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}}))$. If $\widehat{\mathbf{W}} = \mathbf{W}(\widehat{\boldsymbol{\theta}})$, then the test statistic of the likelihood ratio test for goodness of model fit is unchanged when the observed variables are standardized.*

**Proof** As given in (1.18), the test statistic for a model with unstandardized observed variables is $G^2 = n(tr\{\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\} - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}| - k)$. In the model with standardized observed variables, $\widehat{\boldsymbol{\Sigma}}$ is replaced by the sample correlation matrix $\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\widehat{\mathbf{W}}^{-\frac{1}{2}}$, and

[16] Make sure you have $n$ rather than $n-1$ in the denominators of the standard deviations. This way, you are working with true MLEs.

$\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$ is replaced by $\mathbf{W}(\widehat{\boldsymbol{\theta}})^{-\frac{1}{2}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\,\mathbf{W}(\widehat{\boldsymbol{\theta}})^{-\frac{1}{2}}$. The resulting test statistic is

$$
\begin{aligned}
G_s^2 &= n\left(tr\left\{\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\widehat{\mathbf{W}}^{-\frac{1}{2}}\left(\mathbf{W}(\widehat{\boldsymbol{\theta}})^{-\frac{1}{2}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\,\mathbf{W}(\widehat{\boldsymbol{\theta}})^{-\frac{1}{2}}\right)^{-1}\right\} - \log\left|\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\widehat{\mathbf{W}}^{-\frac{1}{2}}\left(\mathbf{W}(\widehat{\boldsymbol{\theta}})^{-\frac{1}{2}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\,\mathbf{W}(\widehat{\boldsymbol{\theta}})^{-\frac{1}{2}}\right)^{-1}\right| - k\right) \\
&= n\left(tr\left\{\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\widehat{\mathbf{W}}^{-\frac{1}{2}}\left(\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\,\widehat{\mathbf{W}}^{-\frac{1}{2}}\right)^{-1}\right\} - \log\left|\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\widehat{\mathbf{W}}^{-\frac{1}{2}}\left(\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\,\widehat{\mathbf{W}}^{-\frac{1}{2}}\right)^{-1}\right| - k\right) \\
&= n\left(tr\left\{\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\underbrace{\widehat{\mathbf{W}}^{-\frac{1}{2}}\widehat{\mathbf{W}}^{\frac{1}{2}}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\,\widehat{\mathbf{W}}^{\frac{1}{2}}\right\} - \log\left|\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\underbrace{\widehat{\mathbf{W}}^{-\frac{1}{2}}\widehat{\mathbf{W}}^{\frac{1}{2}}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\,\widehat{\mathbf{W}}^{\frac{1}{2}}\right| - k\right) \\
&= n\left(tr\left\{\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\,\widehat{\mathbf{W}}^{\frac{1}{2}}\right\} - \log\left|\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\,\widehat{\mathbf{W}}^{\frac{1}{2}}\right| - k\right) \\
&= n\left(tr\left\{\widehat{\mathbf{W}}^{\frac{1}{2}}\widehat{\mathbf{W}}^{-\frac{1}{2}}\,\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\right\} - \log\left(\left|\widehat{\mathbf{W}}^{-\frac{1}{2}}\right|\,\left|\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\right|\,\left|\widehat{\mathbf{W}}^{\frac{1}{2}}\right|\right) - k\right) \\
&= n\left(tr\left\{\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\right\} - \log\left(\frac{\left|\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\right|\,\left|\widehat{\mathbf{W}}^{\frac{1}{2}}\right|}{\left|\widehat{\mathbf{W}}^{\frac{1}{2}}\right|}\right) - k\right) \\
&= n\left(tr\left\{\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\right\} - \log\left|\widehat{\boldsymbol{\Sigma}}\,\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\right| - k\right). \\
&= G^2 \quad \blacksquare
\end{aligned}
$$

**Normal inference is unaffected by standardizing** Theorem 3.1 depends on $\mathbf{W}(\widehat{\boldsymbol{\theta}})$ being equal to $\widehat{\mathbf{W}}$. As indicated in the discussion leading up to (3.31), this condition holds when the error variances $\omega_{\ell,\ell}$ appear only in the diagonal of $\boldsymbol{\Omega}$ and are not functions of one another or of other parameters in the model. Most confirmatory factor analysis models employed in practice enjoy this property. Likelihood ratio tests are differences in $G^2$ fit statistics between a restricted and an unrestricted model. Wald tests are asymptotically equivalent to likelihood ratio tests under the null hypothesis. Confidence intervals can be obtained by inverting tests. The result is that for most confirmatory factor analyses, inference based on the normal model is unaffected by standardizing the observable variables. The choice to standardize or not is entirely a matter of convenience and interpretability.

## 3.8 The Holzinger and Swineford Data with `lavaan`

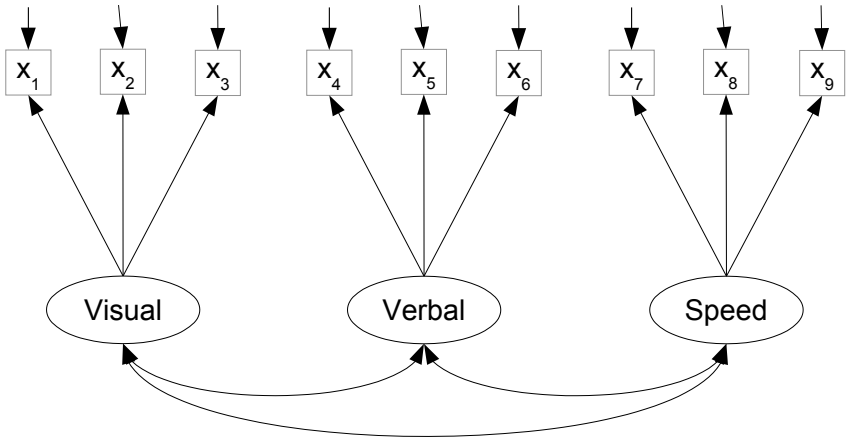
The Holzinger and Swineford (1939) data is a classic data set that is used in multiple textbooks and journal articles. It is included in the `lavaan` package, and is used in a confirmatory factor analysis example in the lavaan tutorial. The data were collected on students in grades seven and eight from two different schools. As in the lavaan tutorial, attention will be limited to nine tests of "mental ability" that are thought to reflect three factors: visualization (tests 1, 2 and 3), verbal or text processing (test 4, 5 and 6) and speed (tests 7, 8 and 9).

| Visual | | Verbal | | Speed | |
|---|---|---|---|---|---|
| $x_1$ | Visual Perception | $x_4$ | Paragraph Comprehension | $x_7$ | Addition |
| $x_2$ | Cubes | $x_5$ | Sentence Completion | $x_8$ | Counting Dots |
| $x_3$ | Lozenges | $x_6$ | Word Meaning | $x_9$ | Straight-Curved Capitals |

The students actually took 24 tests; the full data set is available in the MBESS package.

Figure 3.15 shows a path diagram. It's pretty straightforward; all the parameters are identifiable at a glance by the three-variable rule. As in Jöreskog's 1969 article [36], the analyses here will be limited to just the 145 children from the Grant-White school. This will provide a valuable cross-check of the numbers we obtain. Figure 3.15 represents Jöreskog's model (d).

Figure 3.15: Holzinger and Swineford Mental Test Data




As usual, the R code that follows is not just an example of how to do the job efficiently. Instead, it explores the capabilities of the software, and seeks to make connections between the computations and the ideas in the rest of the text. Students who imitate all these operations to do an assignment are missing the point. The examples you are most likely to want to follow tend to come near the end. The hope is by that point, you will know what's going on.

### Acquiring the data

```
> rm(list=ls())
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
> # help(HolzingerSwineford1939)
> hs = HolzingerSwineford1939
> hs = subset(hs,school=='Grant-White'); dim(hs) # 145 rows, 15 columns
[1] 145  15
> print(head(hs),digits=3)
     id sex ageyr agemo      school grade   x1   x2   x3   x4   x5   x6   x7   x8   x9
157 201   1    13     0 Grant-White     7 3.83 4.75 0.50 3.33 4.25 1.43 3.00 4.10 4.33
158 202   2    11    10 Grant-White     7 5.50 5.50 2.12 2.67 4.25 1.43 2.83 4.90 5.42
```

```
159 203   1    12     6 Grant-White     7 5.67 6.00 2.75 3.67 4.75 2.71 2.17 4.30 6.33
160 204   1    11    11 Grant-White     7 4.83 5.75 1.12 3.00 4.75 1.57 4.96 5.15 4.00
161 205   1    12     5 Grant-White     7 2.67 6.25 1.25 2.67 6.25 3.43 4.87 6.10 4.44
162 206   2    12     6 Grant-White     7 5.00 6.25 2.50 3.33 5.75 2.57 4.09 5.65 5.58
```

**Standardized factors, complete model specification** First, a model will be specified using the full lavaan syntax, giving names to all the parameters. This is the way it was done in Chapters 0 and 1. It will be seen presently that there is an easier way to get the job done.

```
> swine1 = '
            # Measurement model
+           visual  =~ lambda1*x1 + lambda2*x2 + lambda3*x3
+           verbal  =~ lambda4*x4 + lambda5*x5 + lambda6*x6
+           speed   =~ lambda7*x7 + lambda8*x8 + lambda9*x9
+           # Variances of error terms
+           x1 ~~ omega1*x1;  x2 ~~ omega2*x2;  x3 ~~ omega3*x3
+           x4 ~~ omega4*x4;  x5 ~~ omega5*x5;  x6 ~~ omega6*x6
+           x7 ~~ omega7*x7;  x8 ~~ omega8*x8;  x9 ~~ omega9*x9
+           # Variances of factors equal one
+           visual ~~ 1*visual; verbal  ~~ 1*verbal ; speed ~~ 1*speed
+           # Covariances of factors
+           visual ~~ phi12*verbal ; visual ~~  phi13*speed
+                                    verbal  ~~ phi23*speed
+         '
> smodel1 = lavaan(swine1, data=hs); summary(smodel1)
lavaan 0.6-7 ended normally after 19 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         21

  Number of observations                           145

Model Test User Model:

  Test statistic                                51.542
  Degrees of freedom                                24
  P-value (Chi-square)                           0.001

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
```

```
                   Estimate  Std.Err  z-value  P(>|z|)
  visual =~
    x1       (lmb1)    0.777    0.103    7.525    0.000
    x2       (lmb2)    0.572    0.101    5.642    0.000
    x3       (lmb3)    0.719    0.093    7.711    0.000
  verbal =~
    x4       (lmb4)    0.971    0.079   12.355    0.000
    x5       (lmb5)    0.961    0.083   11.630    0.000
    x6       (lmb6)    0.935    0.081   11.572    0.000
  speed =~
    x7       (lmb7)    0.679    0.087    7.819    0.000
    x8       (lmb8)    0.833    0.087    9.568    0.000
    x9       (lmb9)    0.719    0.086    8.357    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  visual ~~
    verbal   (ph12)    0.541    0.085    6.355    0.000
    speed    (ph13)    0.523    0.094    5.562    0.000
  verbal ~~
    speed    (ph23)    0.336    0.091    3.674    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .x1       (omg1)    0.715    0.126    5.675    0.000
   .x2       (omg2)    0.899    0.123    7.339    0.000
   .x3       (omg3)    0.557    0.103    5.409    0.000
   .x4       (omg4)    0.315    0.065    4.870    0.000
   .x5       (omg5)    0.419    0.072    5.812    0.000
   .x6       (omg6)    0.406    0.069    5.880    0.000
   .x7       (omg7)    0.600    0.091    6.584    0.000
   .x8       (omg8)    0.401    0.094    4.248    0.000
   .x9       (omg9)    0.535    0.089    6.010    0.000
    visual             1.000
    verbal             1.000
    speed              1.000
```

The output is pretty much self-explanatory to a reader who is familiar with the lavaan examples in Chapters 0 and 1. The estimated covariances (correlations) of the factors match Jöreskog's (1969, p. 192) values for Model (d). The factor loadings do not match, because Jöreskog standardizes the observed variables; we have not done that yet. The chi-squared test for model fit ($\chi^2(24) = 51.542$, $df = 21$, $p = 0.001$) indicates that the model is not fully compatible with the data[17]. A model based on the reference variable

[17] Jöreskog's value for the test of fit is 51.19. That's close, but not quite equal to the lavaan value. The reason is that in the formula for the likelihood ratio test test statistic (see Expression (1.18) on page 171) Jöreskog has a multiplier of $n - 1$ out in front, in place of $n$. This makes no difference asymptotically, of

rule performs better; we will get to that later.

**The cfa function** The same job can be accomplished with less work, using lavaan's `cfa` (confirmatory factor analysis) function with the default settings. Only the measurement part of the model needs to be given, and all the Greek letters are gone. There is a lot less typing. There are also fewer opportunities to make mistakes.

```
> swine2 = 'visual =~ x1 + x2 + x3
+           verbal =~ x4 + x5 + x6
+           speed  =~ x7 + x8 + x9
+          '
> smodel2 = cfa(swine2, data=hs); summary(smodel2)
lavaan 0.6-7 ended normally after 34 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         21

  Number of observations                           145

Model Test User Model:

  Test statistic                                51.542
  Degrees of freedom                                24
  P-value (Chi-square)                           0.001

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  visual =~
    x1                1.000
    x2                0.736    0.155    4.760    0.000
    x3                0.925    0.166    5.584    0.000
  verbal =~
    x4                1.000
    x5                0.990    0.087   11.418    0.000
    x6                0.963    0.085   11.377    0.000
```

course. To obtain Jöreskog's value from the lavaan output, $(n-1)/n\,G^2 = 144/145 * 51.542 = 51.18654$.

Jöreskog's likelihood approach is based on a Wishart distribution for a version of the sample covariance matrix with $n-1$ in the denominator. Some software, including SAS and Amos, follow Jöreskog's old LISREL software in this matter. Both lavaan and mplus, like this book, assume a multivariate normal likelihood for the original data, rather than starting with the covariance matrix.

```
  speed =~
    x7                1.000
    x8                1.226    0.187    6.569    0.000
    x9                1.058    0.165    6.429    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  visual ~~
    verbal            0.408    0.098    4.153    0.000
    speed             0.276    0.076    3.639    0.000
  verbal ~~
    speed             0.222    0.073    3.022    0.003

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .x1                0.715    0.126    5.675    0.000
   .x2                0.899    0.123    7.339    0.000
   .x3                0.557    0.103    5.409    0.000
   .x4                0.315    0.065    4.870    0.000
   .x5                0.419    0.072    5.812    0.000
   .x6                0.406    0.069    5.880    0.000
   .x7                0.600    0.091    6.584    0.000
   .x8                0.401    0.094    4.248    0.000
   .x9                0.535    0.089    6.010    0.000
    visual            0.604    0.160    3.762    0.000
    verbal            0.942    0.152    6.177    0.000
    speed             0.461    0.118    3.910    0.000
```

Let's take a close look to see what we have. The number of free parameters equals 21, as in the summary of `smodel1`. The chi-squared statistics for model fit are the same. This is promising.

Now compare the estimated factor loadings under `Latent Variables` in the summaries of `smodel1` and `smodel2`. The numbers are different, but don't worry about that yet. The abbreviations for the parameter names are missing for `smodel2`; this is really no great loss. The output is quite readable if you understand `=~` as standing for "is measured by." Under `Variances`, note that when an observable variable is preceded by a dot, it means this is the estimated variance not of the variable, but of its error term. Comparison with the `smodel1` summary, which has labels, helps to confirm this. Once you get used to lavaan output, the parameter labels are really not necessary. If you wish, you can compromise by supplying names for just some of the parameters. This can be a convenient way to set two parameters equal; just give them the same name.

In the summary of `smodel2`, the estimated factor loadings for $x_1$, $x_4$ and $x_7$ are all equal to one, and there are no standard errors or tests. By default, lavaan is fitting a surrogate model with a factor loading set to one for each factor; that's the model described as "Model One" in Section 3.3.1. The factor loadings of one could be mysterious for users who don't know about parameter identifiability, but lavaan is making a choice that's

designed to be helpful. It probably *is* helpful, most of the time.

Just to confirm the meaning of the parameter estimates, recall that under Model One, the factor loading for $x_2$ is $\lambda'_2 = \lambda_2/\lambda_1$. Under Model Two,

$$\begin{aligned} \frac{\lambda''_2}{\lambda''_1} &= \frac{\lambda_2\sqrt{\phi_{11}}}{\lambda_1\sqrt{\phi_{11}}} \\ &= \frac{\lambda_2}{\lambda_1} = \lambda'_2. \end{aligned}$$

By invariance, this equality must also be true of the MLEs. This means that $\widehat{\lambda}'_2 = 0.736$ (the factor loading for $x_2$ in the `smodel2` summary) can be recovered from the `smodel1` output, as follows. $\widehat{\lambda}''_2/\widehat{\lambda}''_1 =$

```
> 0.572/0.777
0.7361647
```

We are on the right track. It is clear that the default model fit in `smodel2` is the surrogate model in which a factor loading has been set to one for each factor.

**Equal diagonals** As given in (3.31), the main diagonal of the reproduced covariance matrix $\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$ is able to match the main diagonal of the sample covariance matrix.

```
> # Checking that the diagonal of Sigma(thetahat) = diagonal of Sigmahat
> x = hs[,7:15]; n = dim(x)[1]
> Sigmahat = (n-1)/n * var(x)
> SigOfThetahat = fitted(smodel2)$cov
> rbind(diag(Sigmahat),diag(SigOfThetahat))
           x1       x2       x3       x4       x5       x6     x7      x8       x9
[1,] 1.318647 1.226379 1.073365 1.257212 1.341665 1.280142 1.0618 1.09438 1.051048
[2,] 1.318647 1.226379 1.073365 1.257212 1.341665 1.280142 1.0618 1.09438 1.051048
```

It worked perfectly, as it does in all but the most peculiar models. Even when they fit badly overall, confirmatory factor analysis models almost always fit the diagonal of $\widehat{\boldsymbol{\Sigma}}$ perfectly.

**Standardized parameter estimates** One often encounters this expression in write-ups of confirmatory factor analysis and structural equation modelling. It's a bit misleading, because it's not the parameter estimates that are standardized. The statistics in question are parameter estimates for a model with the factors standardized — or both the factors and the observed variables standardized. The easiest way to get these numbers from lavaan is by adding the `standardized=TRUE` option to `summary`. Notice that the model does not to be re-fit.

```
> summary(smodel2, standardized=TRUE)
lavaan 0.6-7 ended normally after 34 iterations
```

```
  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         21

  Number of observations                           145

Model Test User Model:

  Test statistic                                51.542
  Degrees of freedom                                24
  P-value (Chi-square)                           0.001

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual =~
    x1                1.000                               0.777    0.677
    x2                0.736    0.155    4.760    0.000    0.572    0.517
    x3                0.925    0.166    5.584    0.000    0.719    0.694
  verbal =~
    x4                1.000                               0.971    0.866
    x5                0.990    0.087   11.418    0.000    0.961    0.829
    x6                0.963    0.085   11.377    0.000    0.935    0.826
  speed =~
    x7                1.000                               0.679    0.659
    x8                1.226    0.187    6.569    0.000    0.833    0.796
    x9                1.058    0.165    6.429    0.000    0.719    0.701

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual ~~
    verbal            0.408    0.098    4.153    0.000    0.541    0.541
    speed             0.276    0.076    3.639    0.000    0.523    0.523
  verbal ~~
    speed             0.222    0.073    3.022    0.003    0.336    0.336

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.715    0.126    5.675    0.000    0.715    0.542
   .x2                0.899    0.123    7.339    0.000    0.899    0.733
   .x3                0.557    0.103    5.409    0.000    0.557    0.519
```

```
.x4               0.315    0.065    4.870    0.000    0.315    0.251
.x5               0.419    0.072    5.812    0.000    0.419    0.312
.x6               0.406    0.069    5.880    0.000    0.406    0.317
.x7               0.600    0.091    6.584    0.000    0.600    0.566
.x8               0.401    0.094    4.248    0.000    0.401    0.367
.x9               0.535    0.089    6.010    0.000    0.535    0.509
 visual           0.604    0.160    3.762    0.000    1.000    1.000
 verbal           0.942    0.152    6.177    0.000    1.000    1.000
 speed            0.461    0.118    3.910    0.000    1.000    1.000
```

The `standardized=TRUE` option has added two columns to the `smodel2` summary output: `Std.lv` and `Std.all`. Naturally, `Std.lv` means that the latent variables (factors) have been standardized. These numbers perfectly match the `Estimate` column of the `smodel1` summary. The `Std.all` column gives estimates for a model where the observable variables as well as the latent variables are standardized. This is sometimes called the "completely standardized" model.

This time, the estimated factor loadings as well as the correlations between factors match Jöreskog's (1969, p. 192) "(d) Restricted Oblique Solution" [36]. This confirms that the `Std.all` values are what we think they are – estimates for a model in which both the factors and the observed variables have been standardized.

**Producing the numbers with matrix operations** It is instructive to see how the `Std.lv` and `Std.all` values could have been obtained[18] from the `smodel2` model fit. In the notation of Section 3.3.1, we are calculating double and triple-prime matrices from single-prime matrices.

First consider `Std.lv`. An application of the invariance principle to (3.22), (3.23) and (3.24) yields

$$\begin{aligned} \widehat{\boldsymbol{\Lambda}}_1'' &= dg(\widehat{\boldsymbol{\Phi}}')^{1/2} \\ \widehat{\boldsymbol{\Lambda}}_2'' &= \widehat{\boldsymbol{\Lambda}}_2' dg(\widehat{\boldsymbol{\Phi}}')^{1/2} \\ \widehat{\boldsymbol{\Phi}}'' &= dg(\widehat{\boldsymbol{\Phi}}')^{-1/2}\, \widehat{\boldsymbol{\Phi}}'\, dg(\widehat{\boldsymbol{\Phi}}')^{-1/2}. \end{aligned} \tag{3.33}$$

How can one obtain those single-prime matrices? The parameter estimates in matrix form are located in "slots" in the fitted lavaan model object. Slots are like properties of the object, or something. There can be slots within slots. One can refer to a slot of an object using the `@` sign, as in `object@slotname`. In the object `smodel2`, the slot called `Model` is an object with 59 slots. One of these is named `GLIST`; it is a list containing the estimated parameter matrices we want. In the following, single primes are represented by `_p`, and double primes are represented by `_pp`. After some experimenting,

```
> Lambda_p = (smodel2@Model)@GLIST$lambda; Lambda_p
          [,1]       [,2]     [,3]
```

[18] In the lavaan code, it may be done in a slightly different but equivalent way. I have not lookrd at the source code.

```
 [1,] 1.0000000 0.0000000 0.000000
 [2,] 0.7361563 0.0000000 0.000000
 [3,] 0.9247953 0.0000000 0.000000
 [4,] 0.0000000 1.0000000 0.000000
 [5,] 0.0000000 0.9897921 0.000000
 [6,] 0.0000000 0.9633398 0.000000
 [7,] 0.0000000 0.0000000 1.000000
 [8,] 0.0000000 0.0000000 1.225840
 [9,] 0.0000000 0.0000000 1.057888
```

Comparing with the numbers in the `smodel2` summary, this is definitely $\widehat{\mathbf{\Lambda}}'$. Of course, it doesn't have the observed variables in $\mathbf{d}_1$ and $\mathbf{d}_2$ separated. Extracting $\widehat{\mathbf{\Lambda}}_2'$ and then the other estimated parameter matrices,

```
> Lambda2_p = Lambda_p[-c(1,4,7),]; Lambda2_p
          [,1]      [,2]     [,3]
[1,] 0.7361563 0.0000000 0.000000
[2,] 0.9247953 0.0000000 0.000000
[3,] 0.0000000 0.9897921 0.000000
[4,] 0.0000000 0.9633398 0.000000
[5,] 0.0000000 0.0000000 1.225840
[6,] 0.0000000 0.0000000 1.057888
> Omega = (smodel2@Model)@GLIST$theta; Omega
           [,1]      [,2]      [,3]      [,4]      [,5]    [,6]      [,7]      [,8]     [,9]
 [1,] 0.7148977 0.0000000 0.0000000 0.0000000 0.0000000 0.00000 0.0000000 0.0000000 0.000000
 [2,] 0.0000000 0.8991918 0.0000000 0.0000000 0.0000000 0.00000 0.0000000 0.0000000 0.000000
 [3,] 0.0000000 0.0000000 0.5570105 0.0000000 0.0000000 0.00000 0.0000000 0.0000000 0.000000
 [4,] 0.0000000 0.0000000 0.0000000 0.3153055 0.0000000 0.00000 0.0000000 0.0000000 0.000000
 [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.4188895 0.00000 0.0000000 0.0000000 0.000000
 [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.40603 0.0000000 0.0000000 0.000000
 [7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000 0.6004945 0.0000000 0.000000
 [8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000 0.0000000 0.4011844 0.000000
 [9,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.00000 0.0000000 0.0000000 0.534789
> Phi_p = (smodel2@Model)@GLIST$psi; Phi_p
          [,1]      [,2]      [,3]
[1,] 0.6037495 0.4077212 0.2761904
[2,] 0.4077212 0.9419068 0.2215664
[3,] 0.2761904 0.2215664 0.4613051
```

Notice that except for $\mathbf{\Lambda}$, lavaan is using a different Greek letter notation for the parameter matrices. Nobody cares.

The matrix $dg(\widehat{\mathbf{\Phi}}')^{1/2}$ appears four times in (3.33). To carry out the calculations, it is convenient to give it a simple name. Call it `M`.

```
> M = sqrt(diag(diag(Phi_p))); M  # = Lambda1_pp, the factor loadings of the leading reference
          [,1]      [,2]      [,3]
[1,] 0.7770132 0.0000000 0.0000000
```

```
[2,] 0.0000000 0.9705188 0.0000000
[3,] 0.0000000 0.0000000 0.6791945
> Lambda2_pp = Lambda2_p %*% M;  Lambda2_pp #  the other factor loadings
          [,1]      [,2]      [,3]
[1,] 0.5720032 0.0000000 0.0000000
[2,] 0.7185782 0.0000000 0.0000000
[3,] 0.0000000 0.9606119 0.0000000
[4,] 0.0000000 0.9349394 0.0000000
[5,] 0.0000000 0.0000000 0.8325836
[6,] 0.0000000 0.0000000 0.7185117
> # Putting the full factor matrix Lambda_pp together,
> Lambda_pp = rbind(M[1,],Lambda2_pp[1:2,], M[2,],Lambda2_pp[3:4,], M[3,],Lambda2_pp[5:6,])
> Lambda_pp
           [,1]      [,2]      [,3]
 [1,] 0.7770132 0.0000000 0.0000000
 [2,] 0.5720032 0.0000000 0.0000000
 [3,] 0.7185782 0.0000000 0.0000000
 [4,] 0.0000000 0.9705188 0.0000000
 [5,] 0.0000000 0.9606119 0.0000000
 [6,] 0.0000000 0.9349394 0.0000000
 [7,] 0.0000000 0.0000000 0.6791945
 [8,] 0.0000000 0.0000000 0.8325836
 [9,] 0.0000000 0.0000000 0.7185117
```

These numbers match the estimated factor loadings in the `Std.lv` column of the `smodel2` model summary. For example, the loading of $x_8$ on the `speed` factor is 0.833 in the `Std.lv` column, and it is 0.8325836 in the matrix `Lambda_pp` above.

```
> Phi_pp = solve(M) %*% Phi_p %*% solve(M); Phi_pp
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.5406683 0.5233425
[2,] 0.5406683 1.0000000 0.3361288
[3,] 0.5233425 0.3361288 1.0000000
```

These estimated correlations match the contents of the `Std.lv` column under `Covariances`.

For the "completely standardized" estimates in the `Std.all` column — that is, for the estimates from a model with both the factors and the observed variables standardized, Expression (3.28) implies

$$\begin{aligned}\widehat{\mathbf{\Lambda}}''' &= \widehat{\mathbf{W}}^{-1/2}\widehat{\mathbf{\Lambda}}'' \\ \widehat{\mathbf{\Omega}}''' &= \widehat{\mathbf{W}}^{-1/2}\widehat{\mathbf{\Omega}}\widehat{\mathbf{W}}^{-1/2}.\end{aligned}$$

Calculating,

```
> # Standardized observed variables
> n = dim(hs)[1]; n
[1] 145
```

```
> Sigma = var(hs[,7:15]) * (n-1)/n # Actually Sigma-hat of course
> W = diag(diag(SigmaHat))
> Lambda_ppp = solve(sqrt(W)) %*% Lambda_pp; Lambda_ppp
           [,1]      [,2]      [,3]
 [1,] 0.6766500 0.0000000 0.0000000
 [2,] 0.5165187 0.0000000 0.0000000
 [3,] 0.6935860 0.0000000 0.0000000
 [4,] 0.0000000 0.8655649 0.0000000
 [5,] 0.0000000 0.8293273 0.0000000
 [6,] 0.0000000 0.8263318 0.0000000
 [7,] 0.0000000 0.0000000 0.6591327
 [8,] 0.0000000 0.0000000 0.7958731
 [9,] 0.0000000 0.0000000 0.7008460
> Omega_ppp = solve(sqrt(W)) %*% Omega %*% solve(sqrt(W)); Omega_ppp
           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]
 [1,] 0.5421448 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000
 [2,] 0.0000000 0.7332085 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000
 [3,] 0.0000000 0.0000000 0.5189386 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000
 [4,] 0.0000000 0.0000000 0.0000000 0.2507973 0.0000000 0.0000000 0.0000000 0.0000000 0.000000
 [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.3122162 0.0000000 0.0000000 0.0000000 0.000000
 [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.3171758 0.0000000 0.0000000 0.000000
 [7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.5655441 0.0000000 0.000000
 [8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.3665861 0.000000
 [9,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.508815
```

The numbers in these matrices match the `Std.all` output. For example, the estimated factor loading of $z_2$ on `visual` is 0.517 in the `Std.all` column, and it is 0.5165187 in `Lambda_ppp` above. The estimated variance of $e_6'''$ is 0.317 in the `Std.all` column, and 0.3171758 in `Omega_ppp` — so, the $x_2$ variable is estimated to be around 32% noise.

It goes without saying (and yet I find myself saying it anyway) that in a practical data analysis job, these matrix calculations would almost never be necessary. It's much easier to just use `standardized=TRUE`, and let lavaan do the work. The purpose of doing the matrix calculations here was to show where those `Std.lv` and `Std.all` numbers come from, and to provide a bridge between the theory and what the software is producing. It's also nice to know how to get at those estimated parameter matrices. I didn't know about slots before I did this.

**Fitting a model with standardized factors the easy way** The main disadvantage of the `standardized=TRUE` option is that one gets estimates, but not standard errors. In the summary output for `smodel2`, the standard errors in the output still apply to the surrogate model in which some loadings were set to one. So for example, if you wanted a confidence interval for a correlation between factors, you would still have a bit of work to do. It is possible to choose a model with standardized factors at the model fitting stage, by using the `std.lv` (standardized latent variables) option, as follows.

```
> smodel3 = cfa(swine2, data=hs, std.lv=TRUE); summary(smodel3, standardized=TRUE)
```

```
lavaan 0.6-7 ended normally after 19 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         21

  Number of observations                           145

Model Test User Model:

  Test statistic                                51.542
  Degrees of freedom                                24
  P-value (Chi-square)                           0.001

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual =~
    x1                0.777    0.103    7.525    0.000    0.777    0.677
    x2                0.572    0.101    5.642    0.000    0.572    0.517
    x3                0.719    0.093    7.711    0.000    0.719    0.694
  verbal =~
    x4                0.971    0.079   12.355    0.000    0.971    0.866
    x5                0.961    0.083   11.630    0.000    0.961    0.829
    x6                0.935    0.081   11.572    0.000    0.935    0.826
  speed =~
    x7                0.679    0.087    7.819    0.000    0.679    0.659
    x8                0.833    0.087    9.568    0.000    0.833    0.796
    x9                0.719    0.086    8.357    0.000    0.719    0.701

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual ~~
    verbal            0.541    0.085    6.355    0.000    0.541    0.541
    speed             0.523    0.094    5.562    0.000    0.523    0.523
  verbal ~~
    speed             0.336    0.091    3.674    0.000    0.336    0.336

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.715    0.126    5.675    0.000    0.715    0.542
```

```
   .x2               0.899    0.123    7.339    0.000    0.899    0.733
   .x3               0.557    0.103    5.409    0.000    0.557    0.519
   .x4               0.315    0.065    4.870    0.000    0.315    0.251
   .x5               0.419    0.072    5.812    0.000    0.419    0.312
   .x6               0.406    0.069    5.880    0.000    0.406    0.317
   .x7               0.600    0.091    6.584    0.000    0.600    0.566
   .x8               0.401    0.094    4.248    0.000    0.401    0.367
   .x9               0.535    0.089    6.010    0.000    0.535    0.509
    visual           1.000                               1.000    1.000
    verbal           1.000                               1.000    1.000
    speed            1.000                               1.000    1.000
```

The `Estimate` column now matches the `Std.lv` column; `std.lv=TRUE` had its intended effect. All is well.

**Analyzing the correlation matrix** There is also a `std.ov` option for the `cfa` function, and one would think the observed variables could be standardized by specifying `std.ov=TRUE`. However, as of this writing[19] it does not quite work as expected. When the observed variables are standardized, they are divided by a sample standard deviation with $n-1$ in the denominator, rather than $n$. This makes no difference asymptotically, so the estimates, tests and confidence intervals are just as good either way. However, if `std.lv` and `std.ov` are both `TRUE`, the numbers in the `Estimate` column don't quite don't quite match the `Std.all` column. It's a bit unsettling.

One option is to standardize the observed variables yourself, making sure you divide by $n$ to get the true MLEs. This works, but it's awkward. It's easier to use the sample correlation matrix as input.

The lavaan software allows a sample covariance matrix and a sample size as input, in place of the raw data[20]. When you give lavaan a correlation matrix, it treats it as a sample covariance matrix. There are two consequences, both a bit subtle. The first is that lavaan assumes the sample variances just happen to be all equal to one (an event of zero probability, by the way), and it treats the error variances (the $\omega'''_{j,j}$) as free parameters to be estimated, rather than using the fact that they are functions of the other parameters. This actually works out very well. The standard errors are correct, and it's a lot easier to obtain confidence intervals for uniquenesses and commonalities than it would be otherwise.

The second consequence of treating the correlation matrix as a covariance matrix is the issue of whether the sample variances and covariances have $n$ in the denominator, or $n-1$. Actually, this question should not apply to sample correlations. Thinking of a sample correlation as a sample covariance divided by a product of sample standard deviations, the denominators, whether they are $n$ or $n-1$, are already cancelled. To lavaan, though, it's just a sample covariance matrix. Many sample covariance matrices (for example, the ones produced by R's `var` function) have $n-1$ in the denominators,

[19] September 2021, lavaan version 0.6-7.

[20] This is really handy for re-analyzing published data, because books and journal articles often display covariance matrices or correlation matrices even when they do not provide access to the raw data.

so that the estimates are unbiased, but not quite true MLEs. The `cfa` function has an option to deal with this. By default, `sample.cov.rescale` is set to `TRUE`, meaning please correct the input sample covariance matrix, multiplying all entries by $(n-1)/n$. If the input matrix is a sample correlation matrix, you want `sample.cov.rescale=FALSE`. Here's how it goes with the Holzinger-Swineford data. I think factor analysis is nicer with standardized observed variables, so to me, this is a good example of how to do a confirmatory factor analysis with lavaan[21].

```
> # Analyze the correlation matrix
> x = hs[,7:15] # Columns 7 through 15 of the data frame: Just the x variables.
> xcorr = cor(x); round(xcorr,3)
      x1    x2    x3    x4    x5    x6    x7    x8    x9
x1 1.000 0.326 0.449 0.342 0.309 0.317 0.104 0.308 0.487
x2 0.326 1.000 0.417 0.228 0.159 0.195 0.066 0.168 0.248
x3 0.449 0.417 1.000 0.328 0.287 0.347 0.075 0.239 0.373
x4 0.342 0.228 0.328 1.000 0.719 0.714 0.209 0.104 0.314
x5 0.309 0.159 0.287 0.719 1.000 0.685 0.254 0.198 0.356
x6 0.317 0.195 0.347 0.714 0.685 1.000 0.179 0.121 0.272
x7 0.104 0.066 0.075 0.209 0.254 0.179 1.000 0.587 0.418
x8 0.308 0.168 0.239 0.104 0.198 0.121 0.587 1.000 0.528
x9 0.487 0.248 0.373 0.314 0.356 0.272 0.418 0.528 1.000
> smodel4 = cfa(swine2, sample.cov=xcorr, sample.nobs=145,
+ std.lv=TRUE, sample.cov.rescale=FALSE)
> summary(smodel4, standardized=TRUE)
lavaan 0.6-7 ended normally after 20 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         21

  Number of observations                           145

Model Test User Model:

  Test statistic                                51.542
  Degrees of freedom                                24
  P-value (Chi-square)                           0.001

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
```

[21] The reader may be thinking "Well, it's about time!"

```
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual =~
    x1                0.677    0.090    7.525    0.000    0.677    0.677
    x2                0.517    0.092    5.642    0.000    0.517    0.517
    x3                0.694    0.090    7.711    0.000    0.694    0.694
  verbal =~
    x4                0.866    0.070   12.355    0.000    0.866    0.866
    x5                0.829    0.071   11.630    0.000    0.829    0.829
    x6                0.826    0.071   11.572    0.000    0.826    0.826
  speed =~
    x7                0.659    0.084    7.819    0.000    0.659    0.659
    x8                0.796    0.083    9.568    0.000    0.796    0.796
    x9                0.701    0.084    8.357    0.000    0.701    0.701

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual ~~
    verbal            0.541    0.085    6.355    0.000    0.541    0.541
    speed             0.523    0.094    5.562    0.000    0.523    0.523
  verbal ~~
    speed             0.336    0.091    3.674    0.000    0.336    0.336

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.542    0.096    5.675    0.000    0.542    0.542
   .x2                0.733    0.100    7.339    0.000    0.733    0.733
   .x3                0.519    0.096    5.409    0.000    0.519    0.519
   .x4                0.251    0.051    4.870    0.000    0.251    0.251
   .x5                0.312    0.054    5.812    0.000    0.312    0.312
   .x6                0.317    0.054    5.880    0.000    0.317    0.317
   .x7                0.566    0.086    6.584    0.000    0.566    0.566
   .x8                0.367    0.086    4.248    0.000    0.367    0.367
   .x9                0.509    0.085    6.010    0.000    0.509    0.509
    visual            1.000                               1.000    1.000
    verbal            1.000                               1.000    1.000
    speed             1.000                               1.000    1.000
# That's it!
```

There we go. The `Estimate` and `Std.lv` both match `Std.all`. These numbers are very interpretable. For example, the estimate of 0.733 for $\omega_{2,2}$ (triple prime deleted) means we estimate that $x_2$ is around 73.3% noise. Because maximum likelihood estimates are asymptotically normal, an approximate 95% confidence interval to go with this estimate is just the estimate plus or minus 1.96 times the standard error.

```
> c(0.733-1.96*0.1, 0.733+1.96*0.1)
[1] 0.537 0.929
```

This confidence interval is produced automatically by `parameterEstimates(smodel4)`.

Also, notice that in obedience to Theorem 3.1, the chi-squared statistic for lack of model fit is still 51.542. It is unaffected by standardizing the observed variables.

**A confidence interval for uniqueness, the hard way** Suppose you do not standardize the observed variables, and you want a point estimate and confidence interval for the uniqueness of $x_2$ — *under the original model*. Assume the lavaan model `swine1` on page 333, the very explicit model with standardized factors. Under this model (with double primes, as in Section 3.3.1),

$$\begin{aligned} Var(x_2) &= Var(\lambda_2'' F_1'' + e_2) \\ &= \lambda_2''^2 Var(F_1'') + \omega_2 \\ &= \lambda_2''^2 \phi_{1,1}'' + \omega_2. \end{aligned}$$

Bearing in mind that $\lambda_2'' = \lambda_2 \phi_{1,1}^{1/2}$ and $\phi_{1,1}'' = 1$, the proportion of unexplained variance under the surrogate model is

$$\begin{aligned} \frac{\omega_2}{\lambda_2''^2 \phi_{1,1}'' + \omega_2} &= \frac{\omega_2}{\left(\lambda_2 \phi_{1,1}^{\frac{1}{2}}\right)^2 \times 1 \ + \omega_2} \\ &= \frac{\omega_2}{\lambda_2^2 \phi_{1,1} + \omega_2}. \end{aligned} \tag{3.34}$$

For the centered original model, $Var(x_2) = \lambda_2^2 \phi_{1,1} + \omega_2$, so that the proportion of unexplained variance is exactly Expression (3.34). Remarkably, this identifiable function of the original model parameters is the same under the surrogate model with standardized factors. It's not the kind of thing you can depend on in general.

In any case, we want a point estimate and confidence interval for $\frac{\omega_2}{\lambda_2''^2 + \omega_2}$. The point estimate can be easily obtained from numbers in the output of `summary(smodel1)`. On a test or quiz, you could do it with a calculator.

```
> 0.899/(0.572^2 + 0.899)
[1] 0.7331689
```

That's exactly the $\widehat{\omega}_2$ of 0.733 from the model with both factors and observed variables standardized. We also want a confidence interval, something that cannot be calculated from the `summary(smodel1)` output.

As a nice smooth function of asymptotically normal MLEs, $\frac{\widehat{\omega}_2}{\widehat{\lambda}_2''^2 + \widehat{\omega}_2}$ is asymptotically normal. All we need is a standard error — an estimated standard deviation. We can get it easily using lavaan's `:=` syntax for estimating non-linear functions of model parameters. Include this line at the end of the `swine1` model string:

```
x2uniqueness := omega2 / (lambda2^2 + omega2)
```

The lovely `:=` feature is only available if you explicitly provide names (labels) for the parameters. It is incompatible with the shorthand syntax of model `smodel2`.

One is tempted to just copy-paste the entire `swine1` string, and add the line at the end and call the result `swine1b` or something. This is not ideal, because it's not a good idea to have more than one slightly different version of the same code floating around. What if you found an error in `swine1`, or decided to change it for some other reason? Would you remember to make the same change(s) in `swine1b`? Here's a better option.

```
> swine1b = paste(swine1, "x2uniqueness := omega2 / (lambda2^2 + omega2)")
```

Take a look at the result.

```
> cat(swine1b)
          # Measurement model
          visual  =~ lambda1*x1 + lambda2*x2 + lambda3*x3
          verbal =~ lambda4*x4 + lambda5*x5 + lambda6*x6
          speed   =~ lambda7*x7 + lambda8*x8 + lambda9*x9
          # Variances of error terms
          x1 ~~ omega1*x1;  x2 ~~ omega2*x2;  x3 ~~ omega3*x3
          x4 ~~ omega4*x4;  x5 ~~ omega5*x5;  x6 ~~ omega6*x6
          x7 ~~ omega7*x7;  x8 ~~ omega8*x8;  x9 ~~ omega9*x9
          # Variances of factors equal one
          visual ~~ 1*visual; verbal ~~ 1*verbal; speed ~~ 1*speed
          # Covariances of factors
          visual ~~ phi12*verbal; visual ~~  phi13*speed
                                  verbal ~~ phi23*speed
          x2uniqueness := omega2 / (lambda2^2 + omega2)
```

The non-linear function is added neatly to the end. If `swine1` changes, `swine1b` will also be changed when the code is re-run. Now fit the new model and look at the results.

```
> smodel1b = lavaan(swine1b, data=hs); summary(smodel1b)
```

The estimate and standard error for `x2uniqueness` appears at the end of the `summary` output. everything else is the same as the output of `summary(smodel1)`. Showing just the last part,

```
Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    x2uniqueness      0.733    0.081    9.078    0.000
```

The estimated uniqueness (73% noise) is exactly the same as the $\widehat{\omega}_2'''$ obtained from `smodel4`, the model with both factors and observed variables standardized. The standard errors are a bit different, 0.081 for `x2uniqueness`, versus 0.10 for $\widehat{\omega}_2'''$ in `smodel4`. The reason is that by default, lavaan uses the multivariate delta method (See Appendix A, page 564) to estimate the standard deviations of non-linear functions of the parameter estimates. These numbers are close to the ones that come from re-parameterization, in the sense that the difference goes to zero in probability as the sample size tends to infinity. They need not be the same for finite sample sizes, but they are equally valid.

Another option is to use the `se = "bootstrap"` option in the `cfa` or `lavaan` function. This yields standard errors based on the bootstrap, which is distribution-free. Because the bootstrap is a randomization technique, the standard errors will be slightly different every time you run your code, unless you set the seed of the random number generator with the `set.seed` function.

**A model that fits** Let's not get too carried away here. We got the lavaan software to do what we want, but the model still does not fit ($\chi^2 = 51.542$, $df = 24$, $0 = 0.001$). This means that the estimates, and especially the tests and confidence intervals, are open to question. Jöreskog's analysis in [36] includes several models that fit the data, including model (c), described as the "Reference variables solution." This is exactly the model of the reference variable rule, except that it's a special case with the errors independent[22].

Starting over for completeness,

```
> rm(list=ls())
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
> hs = subset(HolzingerSwineford1939,school==’Grant-White’)
> x = hs[,7:15]; xcorr = cor(x)
```

Now specify the reference variable model. The reference variables ($x_1$, $x_4$ and $x_7$) are out front, while the other observed variables, which are influenced by all factors, are grouped together, identically in each line of the model string.

```
> swine3 = ’
+           visual =~ x1 + x2+x3+x5+x6+x8+x9
+           verbal =~ x4 + x2+x3+x5+x6+x8+x9
+           speed  =~ x7 + x2+x3+x5+x6+x8+x9
+          ’
```

Now fit the model, analyzing the correlation matrix because it is the easiest way to standardize the observed variables.

```
> smodel5 = cfa(swine3, sample.cov=xcorr, sample.nobs=145,
+               std.lv=TRUE, sample.cov.rescale=FALSE)
> summary(smodel5) # standardized=TRUE is not necessary.
lavaan 0.6-7 ended normally after 32 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         33
```

[22] Actually, I discovered the reference variable rule by attempting to generalize Jöreskog's model (c). Others may have known about this rule, but I did not.

```
  Number of observations                          145

Model Test User Model:

  Test statistic                                9.846
  Degrees of freedom                               12
  P-value (Chi-square)                          0.629

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  visual =~
    x1                0.708    0.087    8.144    0.000
    x2                0.538    0.123    4.362    0.000
    x3                0.674    0.125    5.392    0.000
    x5               -0.033    0.093   -0.350    0.726
    x6                0.013    0.092    0.137    0.891
    x8                0.415    0.115    3.612    0.000
    x9                0.557    0.113    4.916    0.000
  verbal =~
    x4                0.871    0.070   12.434    0.000
    x2               -0.031    0.118   -0.265    0.791
    x3                0.042    0.119    0.354    0.724
    x5                0.808    0.090    8.939    0.000
    x6                0.819    0.090    9.055    0.000
    x8               -0.298    0.111   -2.673    0.008
    x9               -0.061    0.110   -0.552    0.581
  speed =~
    x7                0.782    0.096    8.161    0.000
    x2               -0.075    0.107   -0.700    0.484
    x3               -0.086    0.108   -0.790    0.429
    x5                0.128    0.074    1.729    0.084
    x6               -0.007    0.075   -0.093    0.926
    x8                0.731    0.109    6.690    0.000
    x9                0.413    0.096    4.321    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  visual ~~
    verbal            0.543    0.112    4.850    0.000
```

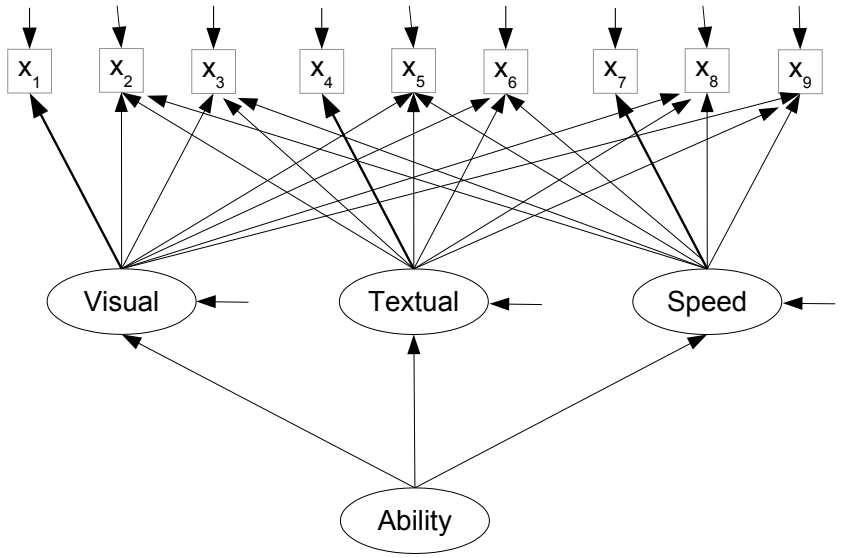
```
    speed             0.240    0.148    1.626    0.104
  verbal ~~
    speed             0.284    0.116    2.439    0.015

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .x1                0.499    0.091    5.498    0.000
   .x2                0.740    0.102    7.245    0.000
   .x3                0.535    0.095    5.661    0.000
   .x5                0.302    0.054    5.589    0.000
   .x6                0.322    0.054    5.924    0.000
   .x8                0.317    0.095    3.323    0.001
   .x9                0.456    0.071    6.421    0.000
   .x4                0.241    0.052    4.626    0.000
   .x7                0.388    0.113    3.428    0.001
    visual            1.000
    verbal            1.000
    speed             1.000
```

This model fits ($\chi^2 = 9.846$, $df = 12$, $p = 0.629$). The estimated factor loadings (and the associated tests) suggest that the model of Figure 3.15 did not fit because $x_8$ (Counting Dots) and $x_9$ (Straight-curved Capitals) are positively influenced by the speed factor. There is also evidence that $x_8$ may be *negatively* influenced by the verbal factor $F_2$.

It is interesting that under the model of Figure 3.15, the estimated correlation between the visual and speed factors is substantial ($\widehat{\phi}_{1,3} = 0.523$) and undeniably significant ($z = 5.562$, $p \approx 0$). See for example the output of `summary(smodel4, standardized=TRUE)`. This is an important conclusion, because it might reflect something fundamental about cognition and the human nervous system. However, for the model that fits the data, there is not enough evidence to conclude a non-zero correlation ($\widehat{\phi}_{1,3} = 0.24$, $z = 1.626$, $p = 0.104$): see the output of `summary(smodel5)` starting on page 349. When a model does not fit the data, conclusions from the significance tests are highly suspect. This issue is discussed in Chapter 7.

**A second-order model** It is fairly reasonable to hypothesize that there is a general factor underlying the visual, verbal and speed factors; it might be called mental ability. Figure 3.16 shows the path diagram. The top part is Jöreskog's [36] Reference Variables model c (also `smodel4`), identified by the reference variable rule. In the lower part, the curved arrows representing correlations between factors have been replaced by the hypothesized second-order ability factor, with arrows pointing from the ability factor to the first-order visual, verbal and speed factors. There are also arrows that seem to come from nowhere, pointing at the first-order factors. These represent error terms. One might think that their variances would introduce three additional parameters, but because the factors are standardized (including ability), the variances are functions of the second-order factor loadings.

Figure 3.16: A second-order model for the Holzinger and Swineford Data




There are three of these second-order factor loadings. They replace the three correlations between first-order factors. Expression (3.13) on page 293 in the proof of the three-variable rule shows that there is a one-to-one connection between the second-order factor loadings and the correlations between first-order factors, provided that the sign of at least second-order loadings is known. Here, there is no problem; theoretically, they are all positive.

To incorporate the second-order ability factor into the model, it's enough to add a line that says ability if measured by visual, verbal and speed.

```
> swine4 = paste(swine3, "ability =~ visual + verbal + speed"); cat(swine4)

          visual =~ x1 + x2+x3+x5+x6+x8+x9
          verbal =~ x4 + x2+x3+x5+x6+x8+x9
          speed  =~ x7 + x2+x3+x5+x6+x8+x9
          ability =~ visual + verbal + speed
```

For the sake of interpretability, I wanted to stay with a "completely standardized" model, in which both the observed and latent variables are standardized.

```
> smodel6 = cfa(swine4, sample.cov=xcorr, sample.nobs=145,
+               std.lv=TRUE, sample.cov.rescale=FALSE)
```

Before looking at any output, let's consider what to expect. First, since the parameters of `smodel5` and `smodel6` are one-to-one, the fit should be the same and and we should get the same chi-squared value of 9.846 with 12 degrees of freedom. Second, all the estimated

first-order factor loadings (and consequently, the error variances) should be the same in the two fitted models. Third, the invariance principle of maximum likelihood estimation[23] dictates a very specific connection between the estimated second-order factor loadings and the estimated correlations between first-order factors. To make this explicit, denote the ability factor by $F_0$, and write the second-order model equations as follows.

$$\begin{aligned} F_1 &= \gamma_1 F_0 + \epsilon_1 \\ F_2 &= \gamma_2 F_0 + \epsilon_2 \\ F_3 &= \gamma_3 F_0 + \epsilon_3 \end{aligned} \tag{3.35}$$

Then, basically transcribing material from 3.12 on page 292, we must have

$$\widehat{\phi}_{1,2} = \widehat{\gamma}_1\widehat{\gamma}_2 \qquad \widehat{\phi}_{1,3} = \widehat{\gamma}_1\widehat{\gamma}_3 \qquad \widehat{\phi}_{2,3} = \widehat{\gamma}_2\widehat{\gamma}_3, \tag{3.36}$$

where the $\widehat{\phi}_{i,j}$ are from the `Covariances` part of the output from `summary(smodel5)`; the output begins on page 349.

Now we know what to expect from `summary(smodel6)`.

```
> summary(smodel6, standardized=TRUE)
lavaan 0.6-7 ended normally after 43 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         33

  Number of observations                           145

Model Test User Model:

  Test statistic                                 9.846
  Degrees of freedom                                12
  P-value (Chi-square)                           0.629

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual =~
    x1                0.520    0.170    3.063    0.002    0.708    0.708
    x2                0.396    0.127    3.116    0.002    0.538    0.538
```

[23] Roughly, the MLE of a function is that function of the MLE.

```
    x3                0.496    0.141    3.508    0.000    0.674    0.674
    x5               -0.024    0.068   -0.355    0.723   -0.033   -0.033
    x6                0.009    0.068    0.137    0.891    0.013    0.013
    x8                0.305    0.127    2.397    0.017    0.415    0.415
    x9                0.409    0.139    2.937    0.003    0.557    0.557
  verbal =~
    x4                0.522    0.286    1.827    0.068    0.871    0.871
    x2               -0.019    0.071   -0.264    0.792   -0.031   -0.031
    x3                0.025    0.073    0.343    0.731    0.042    0.042
    x5                0.484    0.264    1.832    0.067    0.808    0.808
    x6                0.491    0.265    1.851    0.064    0.819    0.819
    x8               -0.178    0.097   -1.847    0.065   -0.298   -0.298
    x9               -0.036    0.064   -0.568    0.570   -0.061   -0.061
  speed =~
    x7                0.731    0.105    6.971    0.000    0.782    0.782
    x2               -0.070    0.099   -0.707    0.480   -0.075   -0.075
    x3               -0.080    0.100   -0.803    0.422   -0.086   -0.086
    x5                0.119    0.070    1.706    0.088    0.128    0.128
    x6               -0.007    0.070   -0.093    0.926   -0.007   -0.007
    x8                0.683    0.105    6.495    0.000    0.731    0.731
    x9                0.386    0.096    4.031    0.000    0.413    0.413
  ability =~
    visual            0.923    0.575    1.605    0.108    0.678    0.678
    verbal            1.336    1.133    1.179    0.238    0.801    0.801
    speed             0.379    0.182    2.085    0.037    0.355    0.355

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.499    0.091    5.498    0.000    0.499    0.499
   .x2                0.740    0.102    7.245    0.000    0.740    0.740
   .x3                0.535    0.095    5.661    0.000    0.535    0.535
   .x5                0.302    0.054    5.589    0.000    0.302    0.302
   .x6                0.322    0.054    5.924    0.000    0.322    0.322
   .x8                0.317    0.095    3.323    0.001    0.317    0.317
   .x9                0.456    0.071    6.421    0.000    0.456    0.456
   .x4                0.241    0.052    4.626    0.000    0.241    0.241
   .x7                0.388    0.113    3.428    0.001    0.388    0.388
   .visual            1.000                              0.540    0.540
   .verbal            1.000                              0.359    0.359
   .speed             1.000                              0.874    0.874
    ability           1.000                              1.000    1.000
```

Comparing this to the `smodel5` output that begins on page 349, we do get the same chi-squared fit test value of 9.846 with 12 degrees of freedom, so that is okay. However, the estimated first-order factor loadings are quite different. For example, the estimated loading that links the visual factor to $x_1$ is 0.520 for `smodel6`, compared to 0.708 for `smodel5`. It's way off.

After a while, I finally saw a hint pointing to the source of the problem. At the end of the `smodel6` output directly above, there are dots in front of `visual`, `verbal` and `speed`. This indicates that we are not looking at estimated variances of variables, but at estimated variances of *error terms*. It would appear that the variances of the first-order factors were not set to one after all. Instead, the variances of the error terms (that is, $\epsilon_1$, $\epsilon_2$ and $\epsilon_3$ in Expression 3.35) were set to one. I verified this by doing some calculations on numbers from the output. The result is a surrogate model that, while it's technically correct and has identifiable parameters, is just strange and does not correspond to anything we want.

On the other hand, the `Std.lv` and `Std.all` columns do contain the desired estimates. Unlike the `std.lv=TRUE` option in the `cfa` function, the `standardized=TRUE` option in `summary` is working as expected. The estimated factor loadings match `summary(smodel5)` perfectly.

To check (3.36), we first obtain $\widehat{\phi}_{1,2} = 0.543$, $\widehat{\phi}_{1,3} = 0.240$ and $\widehat{\phi}_{2,3} = 0.284$ from the `Covariances` part of `summary(smodel5)`. Then, obtaining $\widehat{\gamma}_1 = 0.678$ (not 0.923), $\widehat{\gamma}_2 = 0.801$ and $\widehat{\gamma}_3 = 0.355$ from `summary(smodel6)`,

```
> 0.678*0.801 # Should equal phihat12 = 0.543
[1] 0.543078
> 0.678*0.355 # Should equal phihat13 = 0.240
[1] 0.24069
> 0.801*0.355 # Should equal phihat13 = 0.284
[1] 0.284355
```

So, the `Std.all` column clearly has the estimates from a model with both the observed variables and the latent variables (not the error terms of the latent variables) standardized. It means, for example, that the estimated correlation between the ability factor and the visual factor equals 0.678 — the same as the factor loading. This is also the estimated correlation for the original model.

There is a slightly easier way to get these numbers, and that is to use the default surrogate model with a factor loading set to one for each factor, including second-order factors. The model string `swine4` is fine as it is.

```
> cat(swine4)
          visual =~ x1 + x2+x3+x5+x6+x8+x9
          verbal =~ x4 + x2+x3+x5+x6+x8+x9
          speed  =~ x7 + x2+x3+x5+x6+x8+x9
          ability =~ visual + verbal + speed
```

The `cfa` function call is simpler.

```
> smodel7 = cfa(swine4, data=hs)
> summary(smodel7, standardized=TRUE)
lavaan 0.6-7 ended normally after 48 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
```

```
  Number of free parameters                         33

  Number of observations                           145

Model Test User Model:

  Test statistic                                 9.846
  Degrees of freedom                                12
  P-value (Chi-square)                           0.629

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual =~
    x1                1.000                               0.813    0.708
    x2                0.733    0.191    3.841    0.000    0.596    0.538
    x3                0.859    0.196    4.378    0.000    0.699    0.674
    x5               -0.046    0.133   -0.350    0.727   -0.038   -0.033
    x6                0.018    0.128    0.137    0.891    0.014    0.013
    x8                0.534    0.162    3.302    0.001    0.434    0.415
    x9                0.702    0.167    4.197    0.000    0.571    0.557
  verbal =~
    x4                1.000                               0.977    0.871
    x2               -0.035    0.134   -0.265    0.791   -0.035   -0.031
    x3                0.045    0.126    0.354    0.724    0.044    0.042
    x5                0.958    0.112    8.534    0.000    0.936    0.808
    x6                0.948    0.109    8.684    0.000    0.926    0.819
    x8               -0.319    0.120   -2.666    0.008   -0.312   -0.298
    x9               -0.064    0.115   -0.552    0.581   -0.062   -0.061
  speed =~
    x7                1.000                               0.806    0.782
    x2               -0.103    0.147   -0.698    0.485   -0.083   -0.075
    x3               -0.110    0.140   -0.786    0.432   -0.089   -0.086
    x5                0.184    0.108    1.696    0.090    0.148    0.128
    x6               -0.010    0.105   -0.093    0.926   -0.008   -0.007
    x8                0.949    0.200    4.747    0.000    0.765    0.731
    x9                0.525    0.135    3.882    0.000    0.423    0.413
  ability =~
    visual            1.000                               0.678    0.678
    verbal            1.418    0.862    1.644    0.100    0.801    0.801
    speed             0.518    0.238    2.173    0.030    0.355    0.355
```

```
Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .x1                0.657    0.120    5.498    0.000    0.657    0.499
   .x2                0.907    0.125    7.245    0.000    0.907    0.740
   .x3                0.575    0.101    5.661    0.000    0.575    0.535
   .x5                0.405    0.073    5.589    0.000    0.405    0.302
   .x6                0.412    0.069    5.924    0.000    0.412    0.322
   .x8                0.347    0.104    3.323    0.001    0.347    0.317
   .x9                0.480    0.075    6.421    0.000    0.480    0.456
   .x4                0.303    0.065    4.626    0.000    0.303    0.241
   .x7                0.412    0.120    3.428    0.001    0.412    0.388
   .visual            0.357    0.233    1.532    0.126    0.540    0.540
   .verbal            0.343    0.375    0.914    0.361    0.359    0.359
   .speed             0.568    0.163    3.486    0.000    0.874    0.874
    ability           0.304    0.208    1.460    0.144    1.000    1.000
```

Notice how all the leading factor loadings are set to one, including for the second-order factor. The `Std.all` column has the same numbers obtained from `smodel6`. If we want the estimates for a completely standardized model and don't care about standard errors, this is all we need. If necessary, one could define custom non-linear functions of the parameters using the `:=` notation, as on page 348, and get standard errors based on the delta method.

**Constraining the error variances** It is possible (but not very convenient) to actually fit a model with the variances of the first-order factors set to one. This is accomplished by constraining the variances of the error terms that feed into the first-order factors in Figure 3.16. The model equations for the latent variable part are given in Expression (3.35); they are repeated below for convenience.

$$\begin{aligned} F_1 &= \gamma_1 F_0 + \epsilon_1 \\ F_2 &= \gamma_2 F_0 + \epsilon_2 \\ F_3 &= \gamma_3 F_0 + \epsilon_3 \end{aligned}$$

Denote $Var(\epsilon_j)$ by $\psi_j$. With the variance of $F_0$ (ability) equal to one, we have $Var(F_j) = \gamma_j^2 + \psi_j$, so that $Var(F_j)$ will equal one provided $\psi_j = 1 - \gamma_j^2$ for $j = 1, 2, 3$. Here is the lavaan model string. It will be considered one piece at a time.

```
> swine5 =  '
+           # Measurement model
+           visual =~ NA*x1 + x2+x3+x5+x6+x8+x9
+           verbal =~ NA*x4 + x2+x3+x5+x6+x8+x9
+           speed  =~ NA*x7 + x2+x3+x5+x6+x8+x9
+           ability =~ NA*visual + gamma1*visual +
+                      gamma2*verbal + gamma3*speed
+           # Variances
```

```
+           ability ~~ 1*ability
+           visual ~~ psi1*visual; verbal ~~ psi2*verbal; speed ~~ psi3*speed
+           # Constraints to make variances of 1st order factors = 1
+           psi1 == 1 - gamma1^2
+           psi2 == 1 - gamma2^2
+           psi3 == 1 - gamma3^2
+           '
```

The first minor hurdle to overcome is that the `std.lv=TRUE` option would standardize the $\epsilon_j$ rather than the $F_j$, which is not what we want. However, if `std.lv=TRUE` is *not* specified, then the `cfa` function will set the leading factor loadings to one for each factor, whether or not parameter names are provided. It would be possible to specify the model more completely and use the `lavaan` function as in the `swine1` model string on page 333, but that's a lot of typing. Here's a better way. Look at the first three lines of the measurement model.

```
visual =~ NA*x1 + x2+x3+x5+x6+x8+x9
verbal =~ NA*x4 + x2+x3+x5+x6+x8+x9
speed  =~ NA*x7 + x2+x3+x5+x6+x8+x9
```

Pre-multiplying the reference variables by `NA` has the effect of *freeing* the factor loading — making it a free parameter to be estimated. The next statement shows that this facility can co-exist with providing a name for the factor loading, by naming the variable twice. This is similar to how starting values are specified — see page 80. It's not enough to name the parameter, when you are using `cfa`.

```
ability =~ NA*visual + gamma1*visual +
           gamma2*verbal + gamma3*speed
```

After fixing the variance of ability equal to one, we give names to the variances of $\epsilon_1$, $\epsilon_2$ and $\epsilon_3$. The rule is that if you want to use a parameter in a constraint, you must name it.

```
ability ~~ 1*ability
visual ~~ psi1*visual; verbal ~~ psi2*verbal; speed ~~ psi3*speed
```

Last come the constraints, set with double equals signs.

```
psi1 == 1 - gamma1^2
psi2 == 1 - gamma2^2
psi3 == 1 - gamma3^2
```

Analyzing the correlation matrix in order to obtain standardized observed variables while avoiding the `std.ov` option[24],

[24] You are forgiven if you forgot that `std.ov` divides by a sample standard deviation with $n-1$ in the denominator.

```
> smodel8 = cfa(swine5, sample.cov=xcorr, sample.nobs=145, sample.cov.rescale=FALSE)
> summary(smodel8, standardized=TRUE)
lavaan 0.6-7 ended normally after 190 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         36

  Number of observations                           145

Model Test User Model:

  Test statistic                                 9.846
  Degrees of freedom                                12
  P-value (Chi-square)                           0.629

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  visual =~
    x1                0.708    0.087    8.144    0.000    0.708    0.708
    x2                0.538    0.123    4.362    0.000    0.538    0.538
    x3                0.674    0.125    5.392    0.000    0.674    0.674
    x5               -0.033    0.093   -0.350    0.726   -0.033   -0.033
    x6                0.013    0.092    0.137    0.891    0.013    0.013
    x8                0.415    0.115    3.612    0.000    0.415    0.415
    x9                0.557    0.113    4.916    0.000    0.557    0.557
  verbal =~
    x4                0.871    0.070   12.434    0.000    0.871    0.871
    x2               -0.031    0.118   -0.264    0.791   -0.031   -0.031
    x3                0.042    0.119    0.354    0.724    0.042    0.042
    x5                0.808    0.090    8.939    0.000    0.808    0.808
    x6                0.819    0.090    9.055    0.000    0.819    0.819
    x8               -0.298    0.111   -2.673    0.008   -0.298   -0.298
    x9               -0.061    0.110   -0.552    0.581   -0.061   -0.061
  speed =~
    x7                0.782    0.096    8.161    0.000    0.782    0.782
    x2               -0.075    0.107   -0.700    0.484   -0.075   -0.075
    x3               -0.086    0.108   -0.790    0.429   -0.086   -0.086
    x5                0.128    0.074    1.729    0.084    0.128    0.128
    x6               -0.007    0.075   -0.093    0.926   -0.007   -0.007
```

```
    x8                0.731    0.109    6.690    0.000    0.731    0.731
    x9                0.413    0.096    4.321    0.000    0.413    0.413
  ability =~
    visual  (gmm1)    0.678    0.228    2.971    0.003    0.678    0.678
    verbal  (gmm2)    0.801    0.244    3.284    0.001    0.801    0.801
    speed   (gmm3)    0.355    0.149    2.385    0.017    0.355    0.355

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    ability           1.000                               1.000    1.000
   .visual  (psi1)    0.540    0.309    1.746    0.081    0.540    0.540
   .verbal  (psi2)    0.359    0.390    0.920    0.358    0.359    0.359
   .speed   (psi3)    0.874    0.105    8.295    0.000    0.874    0.874
   .x1                0.499    0.091    5.498    0.000    0.499    0.499
   .x2                0.740    0.102    7.245    0.000    0.740    0.740
   .x3                0.535    0.095    5.661    0.000    0.535    0.535
   .x5                0.302    0.054    5.589    0.000    0.302    0.302
   .x6                0.322    0.054    5.924    0.000    0.322    0.322
   .x8                0.317    0.095    3.323    0.001    0.317    0.317
   .x9                0.456    0.071    6.421    0.000    0.456    0.456
   .x4                0.241    0.052    4.626    0.000    0.241    0.241
   .x7                0.388    0.113    3.428    0.001    0.388    0.388

Constraints:
                                             |Slack|
    psi1 - (1-gamma1^2)                        0.000
    psi2 - (1-gamma2^2)                        0.000
    psi3 - (1-gamma3^2)                        0.000
```

The `Estimate` column is identical to the `Std.all` column, so it worked. This example shows that explicitly constraining error variances is an effective way to standardize endogenous latent variables. However, it can be tedious for large, multistage models. By using `parameterEstimates(smodel8)`, one could automatically obtain 95% confidence intervals for all the parameters, including the ones ($\psi_1$, $\psi_2$ and $\psi_3$) that have been made functionally dependent on other parameters.

**Testing equal factor loadings** Constraints are a handy way to specify null hypotheses for likelihood ratio tests. They may be placed in the model string, but it's preferable to give them in the `cfa` or `lavaan` statement. That way, the same model string can be used to specify the full model and the restricted model.

Suppose we wish to test equality of the second order factor loadings; the null hypothesis is $H_0 : \gamma_1 = \gamma_2 = \gamma_3$. The model under this null hypothesis is expressed as

```
> smodel8H0 = cfa(swine5, sample.cov=xcorr,
+                 sample.nobs=145, sample.cov.rescale=FALSE,
+                 constraints = 'gamma1 == gamma2
+                                gamma2 == gamma3')
```

It's often advisable to look at a summary of the restricted model, just to be sure that nothing obvious has gone wrong. That step is not shown here. Then, the `anova` function generates a likelihood ratio test. If $p < 0.5$, the null hypothesis given in the restricted model is rejected at the 0.05 significance level.

```
> anova(smodel8H0,smodel8)
Chi-Squared Difference Test

          Df    AIC    BIC   Chisq Chisq diff Df diff Pr(>Chisq)
smodel8   12 3273.5 3371.7  9.8461
smodel8H0 14 3273.5 3365.8 13.8616     4.0155       2     0.1343
```

So, there's insufficient evidence to conclude that the second-order factor loadings are different. Or, one could say that the results are consistent with equal second-order factor loadings[25].

To do this test with `smodel7` (in which leading factor loadings equal one), realize that under `smodel8`, the $\gamma_j$ are exactly the correlations of $F_0$ with $F_j$. They are also the correlations of $F_0$ with $F_j$, under the original model, and under the model of `smodel7`. So for the model of `smodel7`, we seek to test the null hypothesis of equal correlations. This implies some constraints on the parameters that are not at all intuitive. The result is either a good homework problem or a place where I need to show my work[26].

For `smodel7`, the equations of the latent part of the model are

$$\begin{aligned} F_1 &= F_0 + \epsilon_1 \\ F_2 &= \gamma_2 F_0 + \epsilon_2 \\ F_3 &= \gamma_3 F_0 + \epsilon_3, \end{aligned}$$

with $Var(F_0) = \phi$, $Var(\epsilon_j) = \psi_j$ for $j = 1, 2, 3$, and of course $F_0$ independent of the $\epsilon_j$. The variances of the first-order factors are

$$Var(F_1) = \phi + \psi_1, \qquad Var(F_2) = \gamma_2^2\phi + \psi_2, \qquad Var(F_3) = \gamma_3^2\phi + \psi_3,$$

and

$$\begin{aligned} Cov(F_0, F_1) &= Cov(F_0, F_0 + \epsilon_1) &&= \phi \\ Cov(F_0, F_2) &= Cov(F_0, \gamma_2 F_0 + \epsilon_2) &&= \gamma_2\phi \\ Cov(F_0, F_3) &= Cov(F_0, \gamma_3 F_0 + \epsilon_3) &&= \gamma_3\phi, \end{aligned}$$

[25] As usual in applied statistics, we are not actively accepting the null hypothesis. For example, if we say that the results are consistent with equal second-order factor loadings, what we really mean is that they are not inconsistent with equal factor loadings. That is, the null hypothesis was not rejected.

[26] Maybe it's both. Some students are surprised when they discover that the answers to many homework problems are directly in the textbook. It's a sneaky way to encourage students to read the text.

so that

$$\begin{array}{rcccl} Corr(F_0,F_1) & = & \frac{\phi}{\sqrt{\phi(\phi+\psi_1)}} & = & \frac{\sqrt{\phi}}{\sqrt{\phi+\psi_1}} \\ Corr(F_0,F_2) & = & \frac{\gamma_2\phi}{\sqrt{\phi(\gamma_2^2\phi+\psi_2)}} & = & \frac{\gamma_2\sqrt{\phi}}{\sqrt{\gamma_2^2\phi+\psi_2}} \\ Corr(F_0,F_3) & = & \frac{\gamma_3\phi}{\sqrt{\phi(\gamma_3^2\phi+\psi_3)}} & = & \frac{\gamma_3\sqrt{\phi}}{\sqrt{\gamma_3^2\phi+\psi_3}}. \end{array}$$

Since $Corr(F_0,F_1) > 0$, the null hypothesis of equal correlations implies $\gamma_2 > 0$ and $\gamma_3 > 0$. Using this,

$$\begin{array}{rcl} Corr(F_0,F_1) = Corr(F_0,F_2) & \iff & \frac{\sqrt{\phi}}{\sqrt{\phi+\psi_1}} = \frac{\gamma_2\sqrt{\phi}}{\sqrt{\gamma_2^2\phi+\psi_2}} \\ & \iff & \frac{1}{\sqrt{\phi+\psi_1}} = \frac{\gamma_2}{\sqrt{\gamma_2^2\phi+\psi_2}} \\ & \iff & \gamma_2\sqrt{\phi+\psi_1} = \sqrt{\gamma_2^2\phi+\psi_2} \\ & \iff & \gamma_2^2(\phi+\psi_1) = \gamma_2^2\phi+\psi_2 \\ & \iff & \gamma_2^2\phi+\gamma_2^2\psi_1 = \gamma_2^2\phi+\psi_2 \\ & \iff & \psi_2 = \gamma_2^2\psi_1. \end{array} \tag{3.37}$$

Similarly,

$$\begin{array}{rcl} Corr(F_0,F_1) = Corr(F_0,F_3) & \iff & \frac{\sqrt{\phi}}{\sqrt{\phi+\psi_1}} = \frac{\gamma_3\sqrt{\phi}}{\sqrt{\gamma_3^2\phi+\psi_3}} \\ & \iff & \frac{1}{\sqrt{\phi+\psi_1}} = \frac{\gamma_3}{\sqrt{\gamma_3^2\phi+\psi_3}} \\ & \iff & \gamma_3\sqrt{\phi+\psi_1} = \sqrt{\gamma_3^2\phi+\psi_3} \\ & \iff & \gamma_3^2(\phi+\psi_1) = \gamma_3^2\phi+\psi_3 \\ & \iff & \gamma_3^2\phi+\gamma_3^2\psi_1 = \gamma_3^2\phi+\psi_3 \\ & \iff & \psi_3 = \gamma_3^2\psi_1. \end{array} \tag{3.38}$$

Finally,

$$\begin{array}{rcl} Corr(F_0,F_2) = Corr(F_0,F_3) & \iff & \frac{\gamma_2\sqrt{\phi}}{\sqrt{\gamma_2^2\phi+\psi_2}} = \frac{\gamma_3\sqrt{\phi}}{\sqrt{\gamma_3^2\phi+\psi_3}} \\ & \iff & \frac{\gamma_2}{\sqrt{\gamma_2^2\phi+\psi_2}} = \frac{\gamma_3}{\sqrt{\gamma_3^2\phi+\psi_3}} \\ & \iff & \gamma_2\sqrt{\gamma_3^2\phi+\psi_3} = \gamma_3\sqrt{\gamma_2^2\phi+\psi_2} \\ & \iff & \gamma_2^2(\gamma_3^2\phi+\psi_3) = \gamma_3^2(\gamma_2^2\phi+\psi_2) \\ & \iff & \gamma_2^2\gamma_3^2\phi+\gamma_2^2\psi_3 = \gamma_2^2\gamma_3^2\phi+\gamma_3^2\psi_2 \\ & \iff & \gamma_2^2\psi_3 = \gamma_3^2\psi_2 \end{array} \tag{3.39}$$

Only two of these constraints are necessary; any two imply the remaining one. One thing that's clear from all this so far is that even though the calculations are elementary, this is a lot of work to set up just one null hypothesis. When the interest is in correlations, a model with standardized variables is preferable. Since most of the work has been done, let's proceed.

In order to impose constraints on parameters in lavaan, the parameters involved must be named in the model string. It's convenient to assemble a new model string by adding to `swine3`.

```
> cat(swine3)
          visual =~ x1 + x2+x3+x5+x6+x8+x9
          verbal =~ x4 + x2+x3+x5+x6+x8+x9
          speed  =~ x7 + x2+x3+x5+x6+x8+x9
> part2  = '# Second order measurement model
+           ability =~ visual + gamma2*verbal + gamma3*speed
+           # Variances of error terms (epsilons)
+           visual ~~ psi1*visual; verbal ~~ psi2*verbal; speed ~~ psi3*speed '
> swine6 = paste(swine3,part2)
> cat(swine6)
          visual =~ x1 + x2+x3+x5+x6+x8+x9
          verbal =~ x4 + x2+x3+x5+x6+x8+x9
          speed  =~ x7 + x2+x3+x5+x6+x8+x9
          # Second order measurement model
          ability =~ visual + gamma2*verbal + gamma3*speed
          # Variances of error terms (epsilons)
          visual ~~ psi1*visual; verbal ~~ psi2*verbal; speed ~~ psi3*speed
```

To test this code, I verified that it produced the same fit and parameter estimates as `smodel7` (starting on page 355), except with a few extra labels. Then I tried to fit the model with the constraints (3.37) and (3.38).

```
> # Constraints are equivalent to equal correlations of F0 with F_j. This is H0.
> smodel7H0 = cfa(swine6, data = hs, constraints = 'psi2 == gamma2^2 * psi1
+                                                   psi3 == gamma2^3 * psi1 ' )
```

It took a long time to run, which is almost always a bad sign. Then,

```
Warning messages:
1: In lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable,  :
  lavaan WARNING: the optimizer warns that a solution has NOT been found!
2: In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats,  :
  lavaan WARNING:
    The variance-covariance matrix of the estimated parameters (vcov)
    does not appear to be positive definite! The smallest eigenvalue
    (= -2.656966e-21) is smaller than zero. This may be a symptom that
    the model is not identified.
3: In lav_object_post_check(object) :
  lavaan WARNING: some estimated lv variances are negative
```

The parameters are identifiable in most of the parameter space, and the regions where they are not identifiable do not correspond to the constraints. So, we can discount suggestion that possibly "the model is not identified"— though typos can accidentally specify a model that is not what one intends, and whose parameters are not identifiable. The warning about negative variance estimates is helpful. Let's look at a summary.

```
> summary(smodel7H0, standardized=TRUE)
lavaan 0.6-7 ended normally after 1336 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         33

  Number of observations                           145

Model Test User Model:

  Test statistic                               671.784
  Degrees of freedom                                14
  P-value (Chi-square)                           0.000

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate    Std.Err  z-value       P(>|z|)   Std.lv  Std.all
  visual =~
    x1                1.000                                        NA       NA
    x2               -0.002    0.001       -3.092    0.002       NA       NA
    x3               -0.004    0.000       -9.062    0.000       NA       NA
    x5               -0.005    0.000      -10.974    0.000       NA       NA
    x6               -0.001    0.001       -2.249    0.025       NA       NA
```

```
    x8                -0.001    0.000        -1.795    0.073       NA       NA
    x9                -0.004    0.000       -10.413    0.000       NA       NA
  verbal =~
    x4                 1.000                                       NA       NA
    x2                 2.116    0.000  32578611.185    0.000       NA       NA
    x3                 3.977    0.000  39039739.233    0.000       NA       NA
    x5                 8.504    0.000  15717415.576    0.000       NA       NA
    x6                 5.219    0.000  94329059.781    0.000       NA       NA
    x8                -0.891    0.000  -5466418.974    0.000       NA       NA
    x9                 1.521    0.000  14915452.630    0.000       NA       NA
  speed =~
    x7                 1.000                                    3.528    0.815
    x2                 0.116    0.000     18821.703    0.000    0.410    0.335
    x3                 0.217    0.000     34832.277    0.000    0.765    0.620
    x5                 0.394    0.000     11697.369    0.000    1.390    0.934
    x6                 0.317    0.000     63860.916    0.000    1.117    0.747
    x8                 0.101    0.000      7754.555    0.000    0.357    0.337
    x9                 0.218    0.000     19810.814    0.000    0.768    0.649
  ability =~
    visual             1.000                                       NA       NA
    verbal   (gmm2)    0.004    0.000    190959.565    0.000       NA       NA
    speed    (gmm3) -14577.264                                   -1.000   -1.000

Variances:
                   Estimate    Std.Err  z-value       P(>|z|)   Std.lv  Std.all
   .visual  (psi1)   -129.680    0.000 -20649480.153    0.000       NA       NA
   .verbal  (psi2)     -0.002    0.001        -1.772    0.076       NA       NA
   .speed   (psi3)     -0.000                                   -0.000   -0.000
   .x1                131.055    0.000  20972882.910    0.000  131.055   95.304
   .x2                  1.334    0.000   4375786.419    0.000    1.334    0.894
   .x3                  0.973    0.000    847631.935    0.000    0.973    0.640
   .x5                  0.441    0.000     33863.868    0.000    0.441    0.199
   .x6                  1.047    0.000    893652.826    0.000    1.047    0.468
   .x8                  0.992    0.000    929543.299    0.000    0.992    0.888
   .x9                  0.820    0.000    498369.776    0.000    0.820    0.584
   .x4                  1.533    0.000  13735738.550    0.000    1.533    1.001
   .x7                  6.310    0.000   5932823.378    0.000    6.310    0.336
    ability             0.000                                    1.000    1.000

Constraints:
                                               |Slack|
    psi2 - (gamma2^2*psi1)                       0.000
    psi3 - (gamma2^3*psi1)                       0.000
# parTable(smodel7H0) # Start obeyed the constraints.
```

It can be beneficial too look at something this ugly. There are several indications that the numerical search for the MLE went off the rails. The number of iterations was 1336,

which is too many; `smodel7` took 48 iterations, and `smodel8` took 190. Also, many of the supposed parameter estimates are really huge. Then there's the $\widehat{\psi}_3$ of -20649480.153. This is an estimated *variance*. The output shows all the signs of a numerical search that accidentally left the parameter space, found a direction that was slightly less bad than where it landed, and then wandered off into nowhere until lavaan (or actually, the underlying `nlminb` function) pulled the plug because of too many iterations.

The standard cure for this disease is better starting values. As we saw in the BMI Health Study (Section 0.10.4, starting on page 89), providing a large number of starting values can be a lot of work. There is an alternative that is promising in this case, but which I have not tried. It's to use the `imposeStart` function from the `semTools` package. Using `imposeStart`, you are able to start a numerical search where another similar model successfully finished. Just provide names for the parameters involved. Here, I would start with the most of the estimates from `smodel7`; it would be necessary to provide labels for the parameters whose estimates were to be used as starting values.

I did not do this, because it turned out that I did not need to. Hoping for the best, I imposed the constraints (3.37) and (3.39) in place of (3.37) and (3.38). Even though these two ways of expressing the null hypothesis are mathematically equivalent, numerical software does not do all the math. I was guessing that the numerical details of imposing the constraints would be sufficiently different so that the search would not get lost at the same point as before. Presumably because I have been a good person my entire life, it worked.

```
> # Try again, with different expression of the same constraints
> smodel7H0 = cfa(swine6, data = hs,
+                 constraints = 'psi2 == gamma2^2 * psi1
+                                gamma2^2 * psi3 == gamma3^2 * psi2 ')
>
> # summary(smodel7H0) # Commented out
> anova(smodel7H0,smodel7)
Chi-Squared Difference Test

          Df    AIC    BIC   Chisq Chisq diff Df diff Pr(>Chisq)
smodel7   12 3494.1 3592.3  9.8461
smodel7H0 14 3494.1 3586.4 13.8616     4.0155       2     0.1343
```

The results are exactly the same as for `anova(smodel8H0,smodel8)` on page 361.

Two lessons may be learned from this last excursion. The first lesson is that it's too much work. If you are interested in an identifiable function of the original model parameters, try to use a surrogate model in which that identifiable function is a model parameter. Expressions (3.17) and (3.19) may be helpful, as may the discussion in Section 3.7. If there is no such surrogate model, okay. But don't make things more complicated than they need to be.

The second lesson is narrow and technical, but unexpected. Suppose a restricted model is being tested against an unrestricted model, and that the unrestricted model fits, while the restricted model does not. Re-expressing the constraints in a mathematically equivalent (and not necessarily simpler) way may be helpful.

## 3.9 The Importance of Planning and Design

In the typical factor analytic study, the investigator hands out a bunch of questionnaires[27], or invites people to complete the questionnaires online. Either way, the observable variables in the study are derived from the responses of the participants. Values of the variables are all generated at more or less the same time and under more or less the same circumstances. This unleashes a flood of very predictable common influences. The respondents' mood, recent experiences, view of the investigator, self-presentation strategies, and guesses about the real purpose of the study — all of these latent variables and many more may be assumed to affect the observable variables.

Chances are very good that such variables are not the focus of the study, and are not among the hypothesized factors. That means they are incorporated into the error terms, and because the same extraneous variables will impact more than one observable variable, the result is non-zero covariances between error terms. These common influences are numerous and we don't know exactly what they are, so the most reasonable model will include all possible covariances between error terms.

Such a model may be reasonable, but it is not useable. There are $k$ observable variables, and an error term for each one. The covariance matrix of the observable variables, $\boldsymbol{\Sigma}$, is $k \times k$, and the covariance matrix of the errors, $\boldsymbol{\Omega}$, is also $k \times k$. There are already as many unknown parameters as covariance structure equations, so the presence of even a single common factor will violate the parameter count rule. Parameter identifiability is out of reach, and so is consistent estimation. Trying to fit the model by maximum likelihood is guaranteed to fail.

Of course, model with all possible covariances between the errors is not what the factor analyst will try to fit. Instead, it is *very* common to assume a model in which all the error terms are independent. The parameters of such a model may be identifiable, but the model is *mis-specified*. That is, it's not correct. Correlations between observable variables will be taken as evidence for the operation of common factors, when in reality they are due to correlations between errors.

How bad will it be? It's really impossible to say. Certainly, parameter estimates will be at least a little off, even for very large sample sizes. Maybe, the effects of the extraneous unmeasured variables will be small compared to the effects of the common factors, and the picture that emerges will be a fair reflection of reality in all essential respects. Or maybe, the correlations between observed variables will be largely determined by the correlations between error terms, making any conclusions from the analysis scientifically worthless. It is impossible to tell, precisely because, apart from the background noise of sampling error, what identifiable means is *knowable*.

When the model with independent errors is applied to data, it may fit and it may not. If it does not fit, it could be that the correlations between errors have created a sample covariance matrix that is inconsistent with the common factor part of the model. At least there is a clue that something is wrong. If the model does fit, it may be that everything is fairly close to being okay, but not necessarily. This is the case with the Holzinger and

[27] Or educational tests.

Swineford data of Section 3.8.

In any observational study, there will inevitably be omitted variables that can distort the results. See Section 0.4 in Chapter 0 for a discussion of how this can affect ordinary regression. However, the focus here is on a set of processes that specifically corrupt the measurement process, and can be controlled.

First, the problem is worst when human subjects are involved, and are aware that their behaviour is being assessed. In contrast, imagine a physical anthropology study in which 14 measurements are to be conducted on a sample of 273 fossilized bones and bone fragments. Measurement error is certainly going to occur, but it's easy enough to minimize correlations among the errors. Just randomize the order in which the measurements are taken, over both bones and features. So for example, the person collecting the data will first measure characteristic 6 on bone 47, then characteristic 11 on bone 122, and so on. It's a bit of extra work, but it would make a model with independent errors quite reasonable.

**Research design** The key to the last example was collecting the data a bit differently. This is an aspect of research design. That's true in more difficult cases as well. This chapter has introduced a good number of rules for establishing parameter identifiability, but there are two big ones — the double measurement rule and the reference variable rule[28]. In both cases, the rules allow for subsets of variables whose error terms might be correlated, but require zero correlation of the error terms for different subsets. For examples, see the BMI health study of Section 0.10.4 and the Brand Awareness study of Section 1.6. Section 3.4 on the reference variable rule also has examples, as well as extended discussion of correlated measurement error and how the design allows for it.

The point here is that there are good alternatives to just handing out a bunch of questionnaires and hoping for the best, but they require advance planning. In other applications of statistics, especially in experiments with random assignment, it is commonplace to think of the research question, the statistical analysis and the details of data collection all at the same time. Factor analysis should be no different. Of course, this principle also holds when a factor analysis model is part of a larger structural equation model.

[28]Why are these two the "big ones?" Because they grant entry to the system, establishing the parameters of a model or sub-model as identifiable. Then other rules may be used to expand the model or put sub-models together.

# Chapter 4

# Path Analysis

## 4.1 Introduction

Path analysis refers to a family of regression-like methods with multiple equations, in which a response variable in one equation may be an explanatory variable in another equation. The term was originated by the American geneticist Sewell Wright, who did his primary work on the topic in the 1920s and 1930s. Wright was the author of numerous influential papers on path analysis. For a statistician, his most noteworthy may be [73], a 1934 paper in the *Annals of mathemetical statistics*. Historically, the field of structural equation modeling arose as a fusion of path analysis and confirmatory factor analysis.

**The two-stage model** Let us begin with the general two-stage structural equation model described in Chapter 1, Section 1.2. Copying from (1.1), the model equations are

$$\begin{aligned} \mathbf{y}_i &= \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\ \mathbf{F}_i &= \begin{pmatrix} \mathbf{x}_i \\ \hline \mathbf{y}_i \end{pmatrix} \\ \mathbf{d}_i &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \end{aligned}$$

where . . . , well a lot of things, given in the model specification that begins on page 137. In this model, $\mathbf{x}_i$ and $\mathbf{y}_i$ are vectors of latent variables that are collected into $\mathbf{F}_i$ in line two. Then line three is a confirmatory factor analysis model, as described in Chapter 3. The presence of intercepts tells us that this is an *original* model, one that has not been centered yet. The first line corresponds to the *latent variable model*, while the factor analysis part is described as the *measurement model*.

The two-stage model is specifically designed to facilitate two-stage proofs of identifiability. Suppose that all the variables have been been centered and the parameters of the measurement model are identifiable, so that $\boldsymbol{\Phi} = cov(\mathbf{F}_i)$ is a function of $\boldsymbol{\Sigma} = cov(\mathbf{d}_i)$. If it can be shown that $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Psi} = cov(\boldsymbol{\epsilon}_i)$ are functions of $\boldsymbol{\Phi}$, then they too are functions of $\boldsymbol{\Sigma}$, and all the parameters are identifiable.

**Surface path analysis** This chapter will focus on the latent variable part of the model, but with a twist. We will pretend that the variables in $\mathbf{x}_i$ and $\mathbf{y}_i$ are observable, yielding what could be called a *surface path analysis model*. The main reason for doing this is ease of exposition. By making all the variables observable, it will be possible to give examples without either providing details of the (irrelevant) measurement part of the model, or leaving those details conspicuously absent.

Except for expected values and intercepts, everything we learn about surface path models will apply directly to the latent variable part of the general structural equation model. This is notably true of identifiability rules, because the process of identifying $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Psi}$ from $\boldsymbol{\Sigma} = cov\begin{pmatrix}\mathbf{x}_i \\ \hline \mathbf{y}_i\end{pmatrix}$ is the same as the process of recovering them from $\boldsymbol{\Phi}$.

It is also true that surface path models are widely used in practice. Of course almost nothing can be measured without error, and as in ordinary regression, ignoring measurement error can have very bad consequences. See Chapter 0 and [14]. Applications of surface path models to real data are no better and no worse than most applications of regression and regression-like methods to observational data.

**The surface path analysis model** In this chapter, we shall adopt the following centered surrogate model. Independently for $i = 1, \ldots, n$, let

$$\mathbf{y}_i = \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i, \tag{4.1}$$

where

- $\mathbf{y}_i$ is a $q \times 1$ observable random vector.
- $\boldsymbol{\beta}$ is a $q \times q$ matrix of constants with zeros on the main diagonal.
- $\boldsymbol{\Gamma}$ is a $q \times p$ matrix of constants.
- $\mathbf{x}_i$ is a $p \times 1$ observable random vector with expected value zero and positive definite covariance matrix $\boldsymbol{\Phi}$.[1]
- $\boldsymbol{\epsilon}_i$ is a $q \times 1$ random vector with expected value zero and positive definite covariance matrix $\boldsymbol{\Psi}$. It is an error term, so it is not observable.
- $\mathbf{x}_i$ and $\boldsymbol{\epsilon}_i$ are independent.

As mentioned in Chapter 1, the $\mathbf{x}_i$ are called *exogenous* variables and the $\mathbf{y}_i$ are called *endogenous* variables. In a path diagram, *end*ogenous variables are found at the *end*s of straight arrows, while the exogenous ($x$) variables do not have any arrows pointing toward them. Error terms are generically exogenous, but they are in a separate category.

Recalling that (4.1) is a model of influence, note that endogenous variables can influence other endogenous variables through the coefficients in the parameter matrix $\boldsymbol{\beta}$. The

[1] In the general two-stage model, $\boldsymbol{\Phi}$ is the covariance matrix of $\mathbf{F}_i$ and $cov(\mathbf{x}_i)$ is denoted by $\boldsymbol{\Phi}_x$. So, the notation in this chapter is not quite consistent with the rest of the book, but it is a bit simpler.

stipulation that $\boldsymbol{\beta}$ has zeros on the main diagonal means that the endogenous variables cannot influence themselves directly, though they may do so indirectly through other variables.

**Covariance matrix** As usual, identifiability and estimation will be based on the covariance matrix of the observable variables. Slightly adapting (1.4),

$$\boldsymbol{\Sigma} = cov\begin{pmatrix}\mathbf{x}_i \\ \hline \mathbf{y}_i\end{pmatrix} = \left(\begin{array}{c|c} \boldsymbol{\Phi} & \boldsymbol{\Phi}\boldsymbol{\Gamma}^\top(\mathbf{I}-\boldsymbol{\beta}^\top)^{-1} \\ \hline (\mathbf{I}-\boldsymbol{\beta})^{-1}\left(\boldsymbol{\Gamma}\boldsymbol{\Phi}\boldsymbol{\Gamma}^\top+\boldsymbol{\Psi}\right) & (\mathbf{I}-\boldsymbol{\beta}^\top)^{-1}\end{array}\right), \tag{4.2}$$

where the existence of the inverses is guaranteed by Theorem 1.1.

**Example 4.1.1** *Birth weight of guinea pigs*

This is a simplified version of an example that Wright [72] gives in a 1921 paper[2]. The example concerns birth weight of guinea pigs. There are three observable variables: birth weight, number of days since mother's last litter, and litter size. Number of days since last litter is a stand-in for gestation period, or how long the guinea pig babies are in the mother. In our conceptual framework, interval since last litter would be a reference variable for the latent variable gestation period, but we'll stick to observable variables here.

The longer the gestation period, the bigger the baby guinea pig should be, because it has more time to grow. Other considerations come into play. As Wright puts it, "a large number in a litter has a fairly direct tendency to shorten the gestation period, but this is probably balanced in part by its tendency to reduce the rate of growth of the foetuses, slow growth permitting a longer gestation period. Large litters tend to reduce gestation period and rate of growth before and after birth. " (p. 561)

So in some way, litter size influences gestation period and gestation period influences birth weight. Litter size also has a direct influence on birth weight. This is depicted in Figure 4.1. In scalar form, the model equations are

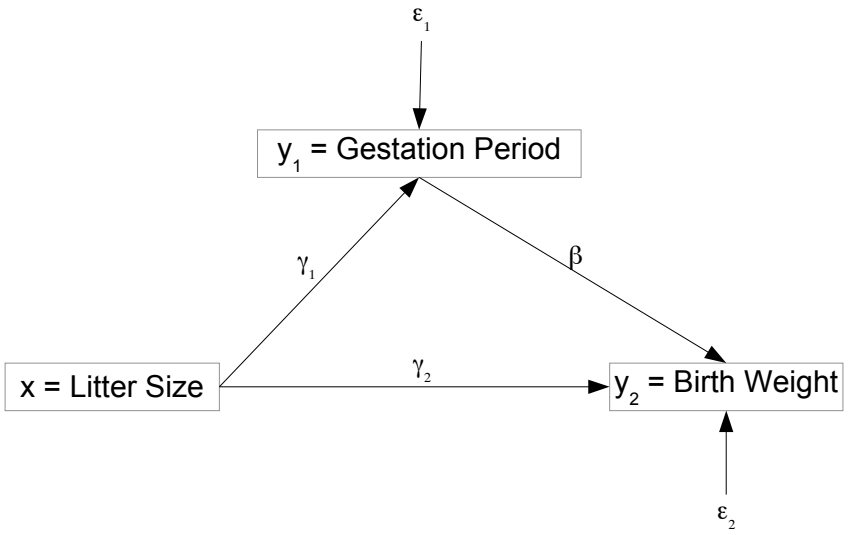
$$\begin{aligned} y_1 &= \gamma_1 x + \epsilon_1 \\ y_2 &= \beta y_1 + \gamma_2 x + \epsilon_2. \end{aligned} \tag{4.3}$$

To clarify the notation, it may be helpful to express the equations in the matrix form of Equation (4.1).

$$\begin{array}{ccccccccc} \mathbf{y} & = & \boldsymbol{\beta} & \mathbf{y} & + & \boldsymbol{\Gamma} & \mathbf{x} & + & \boldsymbol{\epsilon} \\ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} & = & \begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix} & \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} & + & \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} & \begin{pmatrix} x \end{pmatrix} & + & \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix} \end{array} \tag{4.4}$$

[2]The title of the paper is "Correlation and causation." When I first saw it, I thought I had found the original source of the warning that correlation does not necessarily imply causation. Wright was far beyond that. His point was that mere correlations ignore prior knowledge about the likely causal connections among the variables. He proposed path analysis as a way of deriving a causal structure that is logically consistent with a set of correlations. He described it as "a method of analysis by which the knowledge that we have in regard to causal relations may be combined with the knowledge of the degree of relationship furnished by the coefficients of correlation." (p. 559)

Figure 4.1: Guinea Pig Birth Weight




with

$$\boldsymbol{\Psi} = cov(\boldsymbol{\epsilon}) = \begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix}.$$

**Identifiability** As usual, parameter identifiability is to be established by solving the covariance structure equations for the parameters, or showing that such a solution is possible. To obtain $\boldsymbol{\Sigma}$ from the scalar version of the model equations, it is helpful to express the endogenous variables only in terms of exogenous variables and error terms. In the general matrix formulation, this process is the source of $(\mathbf{I} - \boldsymbol{\beta})^{-1}$ in (4.2). For the guinea pig example, all that's necessary is to substitute the first equation of (4.3) into the second. Then, elementary calculations yield

$$\boldsymbol{\Sigma} = cov \begin{pmatrix} x \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \phi & \gamma_1\phi & (\beta\gamma_1 + \gamma_2)\phi \\ & \gamma_1^2\phi + \psi_1 & \gamma_1(\beta\gamma_1 + \gamma_2)\phi + \beta\psi_1 \\ & & (\beta\gamma_1 + \gamma_2)^2\phi + \beta^2\psi_1 + \psi_2 \end{pmatrix}. \tag{4.5}$$

The parameters $\phi$, $\gamma_1$ and $\psi_1$ are easily identified. After that it does not look very obvious, until one notices that the expressions for $\sigma_{1,3}$ and $\sigma_{2,3}$ yield two linear equations in the two unknowns $\beta$ and $\gamma_2$. Even without going into the details, we can still be assured that a unique solution exists, at least in most of the parameter space. Finally, $\psi_2$ may be obtained from $\sigma_{3,3}$ by subtraction. Thus, at least in a rough way, it is established that the model parameters are identifiable.

**Sage** The calculations leading to (4.5) were elementary, but even for this simple example they were a bit tedious. Also, for a detailed picture of where in the parameter space the

parameters are identifiable, it's necessary to solve the two linear equations for $\beta$ and $\gamma_2$. This is also a chore, and most people would not bother. Sage can help. In the following, quite a bit of the code follows Section 1.4.4.

The `sem` package has functions that make it easy to set up the matrices $\mathbf{\Phi}$, $\boldsymbol{\beta}$, $\mathbf{\Gamma}$ and $\mathbf{\Psi}$ as given in (4.4).

```
# Guinea pig example
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
# Set up Phi, Gamma, Beta, Psi
PHI = ZeroMatrix(1,1); PHI[0,0] = var('phi'); show(PHI)
GAMMA = ZeroMatrix(2,1)
GAMMA[0,0] = var('gamma1'); GAMMA[1,0] = var('gamma2'); show(GAMMA)
BETA = ZeroMatrix(2,2); BETA[1,0] = var('beta'); show(BETA)
# The default symbol for DiagonalMatrix is psi
PSI = DiagonalMatrix(2); show(PSI)
```

evaluate

$$\begin{pmatrix} \phi \end{pmatrix}$$
$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}$$
$$\begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix}$$
$$\begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix}$$

The `PathCov` function takes the $\Phi$, $\beta$, $\Gamma$ and $\Psi$ matrices as input, and calculates (4.2) to return the covariance matrix $\mathbf{\Sigma}$. Type `PathCov?` in the Sage environment for details.

```
# Calculate the covariance matrix.
Sigma = PathCov(Phi=PHI,Beta=BETA,Gamma=GAMMA,Psi=PSI)
show(Sigma)
```

evaluate

$$\begin{pmatrix} \phi & \gamma_1\phi & (\beta\gamma_1+\gamma_2)\phi \\ \gamma_1\phi & \gamma_1^2\phi+\psi_1 & \beta\gamma_1^2\phi+\gamma_1\gamma_2\phi+\beta\psi_1 \\ (\beta\gamma_1+\gamma_2)\phi & \beta\gamma_1^2\phi+\gamma_1\gamma_2\phi+\beta\psi_1 & \beta^2\gamma_1^2\phi+2\,\beta\gamma_1\gamma_2\phi+\gamma_2^2\phi+\beta^2\psi_1+\psi_2 \end{pmatrix}$$

There are six covariance structure equations in six unknowns. With the same number of equations and unknowns, Sage's  function is a powerful tool. First, it's necessary to set up the covariance structure equations. The `SetupEqns` does this, taking a symbolic covariance matrix as input. See `SetupEqns?` for details and options.

```
# Show identifiability by direct solution
param = [phi, beta, gamma1, gamma2, psi1, psi2] # List of model parameters
eqns = SetupEqns(Sigma)
for item in eqns: show(item)
```

evaluate

$\phi = \sigma_{11}$
$\gamma_1\phi = \sigma_{12}$
$(\beta\gamma_1 + \gamma_2)\phi = \sigma_{13}$
$\gamma_1^2\phi + \psi_1 = \sigma_{22}$
$\beta\gamma_1^2\phi + \gamma_1\gamma_2\phi + \beta\psi_1 = \sigma_{23}$
$\beta^2\gamma_1^2\phi + 2\,\beta\gamma_1\gamma_2\phi + \gamma_2^2\phi + \beta^2\psi_1 + \psi_2 = \sigma_{33}$

The `solve` function returns a *list* of solutions. How many are there?

```
solut = solve(eqns,param); len(solut)
```

evaluate

1

There is one solution; it is is number zero in the list (of lists).

```
solut = solut[0] # First and only item in the list
for item in solut: show(item)
```

evaluate

$\phi = \sigma_{11}$
$\beta = \frac{\sigma_{12}\sigma_{13} - \sigma_{11}\sigma_{23}}{\sigma_{12}^2 - \sigma_{11}\sigma_{22}}$
$\gamma_1 = \frac{\sigma_{12}}{\sigma_{11}}$
$\gamma_2 = -\frac{\sigma_{13}\sigma_{22} - \sigma_{12}\sigma_{23}}{\sigma_{12}^2 - \sigma_{11}\sigma_{22}}$
$\psi_1 = -\frac{\sigma_{12}^2 - \sigma_{11}\sigma_{22}}{\sigma_{11}}$
$\psi_2 = \frac{\sigma_{13}^2\sigma_{22} - 2\,\sigma_{12}\sigma_{13}\sigma_{23} + \sigma_{12}^2\sigma_{33} + \left(\sigma_{23}^2 - \sigma_{22}\sigma_{33}\right)\sigma_{11}}{\sigma_{12}^2 - \sigma_{11}\sigma_{22}}$

Now it's clear that the solution exists and we have identifiability except where $\sigma_{12}^2 - \sigma_{11}\sigma_{22} = 0$. What is that in terms of the model parameters?

```
# For identifiability, this determinant must not be zero.
Sigma[0,0]*Sigma[1,1] - Sigma[0,1]*Sigma[1,0]
```

evaluate

$$-\gamma_1^2\phi^2 + (\gamma_1^2\phi + \psi_1)\phi$$

```
# Oh come on
expand(_)
```

evaluate

$$\phi\psi_1$$

This is very good. Both $\phi$ and $\psi_1$ are variances, and they are never zero. That means the unique solution exists for all valid parameter values, and we have identifiability everywhere in the parameter space.

In this particular case, since the number of covariance structure equations equals the number of unknowns, the model is called *saturated*, or *just identified*. It fits the sample covariance matrix perfectly, and its fit is untestable by the likelihood ratio chi-squared test. By the invariance principle, the maximum likelihood estimates may be calculated exactly by putting hats on the Greek letters in the solution for the model parameters.

**Standardized observed variables** In classical path analysis as developed by Wright, the observed variables are standardized, and we work with correlations rather than variances and covariances. For the guinea pig example, this means that $\phi = Var(x) = 1$, and also that $\psi_1$ and $\psi_2$ are no longer free parameters, since $Var(y_1) = Var(y_2) = 1$ dictates that $\psi_1 = 1 - \gamma_1^2\phi$ and $\psi_2 = 1 - (\beta\gamma_1 + \gamma_2)^2\phi - \beta^2\psi_1$. It's convenient to do the calculations with Sage.

```
# Get correlation matrix for surrogate model with standardized variables.
Rho = Sigma(psi2 = 1-(beta*gamma1+gamma2)^2*phi-beta^2*psi1)
Rho = factor(Rho(phi=1, psi1=1-gamma1^2))
show(Rho)
```

evaluate

$$\begin{pmatrix} 1 & \gamma_1 & \beta\gamma_1 + \gamma_2 \\ \gamma_1 & 1 & \gamma_1\gamma_2 + \beta \\ \beta\gamma_1 + \gamma_2 & \gamma_1\gamma_2 + \beta & 1 \end{pmatrix}$$

Even in this toy example, it was necessary to do the substitutions carefully in stages and in the correct order in order to eliminate both $\psi_1$ and $\psi_2$. Especially for bigger models, it it preferable to use the `PathCorr` function from the `sem` package.

```
PathCorr(Phi=PHI,Beta=BETA,Gamma=GAMMA,Psi=PSI)
```

evaluate

$$\begin{pmatrix} 1 & \gamma_1 & \beta\gamma_1+\gamma_2 \\ \gamma_1 & 1 & \gamma_1\gamma_2+\beta \\ \beta\gamma_1+\gamma_2 & \gamma_1\gamma_2+\beta & 1 \end{pmatrix}$$

However one calculates it, this is a lot easier to look at and more informative than the covariance matrix of the unstandardized variables. In particular, notice $\rho_{1,3} = Cov(x, y_2) = \beta\gamma_1 + \gamma_2$. Looking back at Figure 4.1, observe that the correlation between $x$ and $y_2$ is composed of two parts that add up. The two parts correspond to the *direct effect* of $x$ on $y_2$, and the *indirect effect* of $x$ on $y_2$ through $y_1$. Further, the indirect effect is obtained by multiplying down the pathway from $x$ to $y_2$ through the *mediating variable* $y_2$. Later in this chapter, we will see the general version in Wright's Theorem and the Multiplication Theorem. First, we develop a couple of essential identifiability rules.

## 4.2 The Regression Rule and the Acyclic Rule

### 4.2.1 The Regression Rule

Taking $\boldsymbol{\beta} = \mathbf{0}$ in (4.1) on page 370 means that endogenous variables have no influence on other endogenous variables. The result is a centered multivariate regression model.

$$\mathbf{y}_i = \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i, \tag{4.6}$$

where $cov(\mathbf{x}_i) = \boldsymbol{\Phi}$ is $p \times p$, $cov(\boldsymbol{\epsilon}_i) = \boldsymbol{\Psi}$ is $q \times q$, and $\boldsymbol{\epsilon}_i$ is independent of $\mathbf{x}_i$. The covariance matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are positive definite.

It is no surprise that the parameters of a regression model are identifiable. Letting

$$\boldsymbol{\Sigma}_{1,1} = cov(\mathbf{x}_i) \qquad \boldsymbol{\Sigma}_{1,2} = cov(\mathbf{y}_i) \qquad \boldsymbol{\Sigma}_{2,2} = cov(\mathbf{y}_i),$$

we have

$$\boldsymbol{\Sigma}_{1,1} = \boldsymbol{\Phi} \qquad \boldsymbol{\Sigma}_{1,2} = \boldsymbol{\Phi}\boldsymbol{\Gamma}^\top \qquad \boldsymbol{\Sigma}_{2,2} = \boldsymbol{\Gamma}\boldsymbol{\Phi}\boldsymbol{\Gamma}^\top + \boldsymbol{\Psi}.$$

Because $\boldsymbol{\Phi} = \boldsymbol{\Sigma}_{1,1}$ is positive definite, $\boldsymbol{\Sigma}_{1,1}^{-1}$ exists. This makes it possible to solve for the parameter matrices, yielding

$$\boldsymbol{\Phi} = \boldsymbol{\Sigma}_{1,1} \qquad \boldsymbol{\Gamma} = \boldsymbol{\Sigma}_{1,2}^\top \boldsymbol{\Sigma}_{1,1}^{-1} \qquad \boldsymbol{\Psi} = \boldsymbol{\Sigma}_{2,2} - \boldsymbol{\Sigma}_{1,2}^\top \boldsymbol{\Sigma}_{1,1}^{-1} \boldsymbol{\Sigma}_{1,2}. \tag{4.7}$$

Thus the model parameters are identifiable everywhere in the parameter space. The following rule has been established[3].

**Rule 3a: Regression Rule** The parameters of the regression model (4.6) are identifiable.

Furthermore, the parameters are *just identifiable*. In other words, the model is *saturated*. Since the parameters are identifiable, this is established by showing that the number of

[3]It was also established earlier using a somewhat different notation; see (16) on page 26.

covariance structure equations equals the number of model parameters. Observe that the $q \times p$ matrix $\mathbf{\Gamma}$ contains $pq$ parameters, $\mathbf{\Phi}$ contains $p(p+1)/2$ parameters, and $\mathbf{\Psi}$ contributes $q(q+1)/2$ more. The covariance matrix $\mathbf{\Sigma}$ has $(p+q)(p+q+1)/2$ unique elements, equal to the number of covariance structure equations. A bit of high school algebra can be skipped by using Sage.

```
# Regression model is just identified.
var('p q')
factor(expand( p*q + p*(p+1)/2 + q*(q+1)/2 ))
```

evaluate

$\frac{1}{2}\,(p+q+1)(p+q)$

By the invariance principle, exact formulas for the MLEs can be obtained by putting hats on all the quantities in the solution (4.7). As given in expression (4) on page 14 (also see page 26), $\widehat{\mathbf{\Gamma}}$ also contains the ordinary least squares estimates of the slopes.

### 4.2.2 The Acyclic Rule

In the model equation $\mathbf{y}_i = \boldsymbol{\beta}\mathbf{y}_i + \mathbf{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i$, setting the diagonal elements of $\boldsymbol{\beta}$ to zero means that no variable may directly influence itself. However, there could easily be feedback loops, in which, for example, $y_1$ influences $y_2$, $y_2$ influences $y_3$, and $y_3$ in turn influences $y_1$. Such a model is termed *cyclic*, because of the cycle of causality.

An *acyclic* model is one without any such feedback loops[4]. Some of the simplest and most useful path models are acyclic – for example, the latent model of the Brand Awareness example (Example 1.2 in Chapter 1; see Figure 1.1). The blood pressure model of Figure 1.3 (page 149) and the guinea pig weight model of Figure 4.1 in this chapter are also acyclic. The following rule gives conditions under which the parmeters of an acyclic model are identifiable.

[4]Acyclic models are sometimes called *recursive* [10, 21].

**Rule 3b: Acyclic Rule** The parameters of the path analysis model (4.1) are identifiable if the model is acyclic (no feedback loops through straight arrows) and the following conditions hold.

- ⋆ Organize the variables that are not error terms into sets. Set 0 consists of all the exogenous variables. They may have non-zero covariances.
- ⋆ For $j = 1, \ldots, m$, each endogenous variable in set $j$ may be influenced by all the variables in sets $\ell < j$.
- ⋆ Error terms for the endogenous variables in a set may have non-zero covariances. All other covariances between error terms are zero[5].

Figure 4.2 illustrates these features.

- Set zero consists of the exogenous variables $x_1$, $x_2$ and $x_3$. They are correlated, as allowed by the rule.
- Set one consists of $y_1$ and $y_2$. These variables are influenced by the variables in set zero, and their error terms have non-zero covariance, as allowed (but not required).
- Set two is composed of $y_3$, $y_4$ and $y_5$. Each variable in this set is influenced by the variables in set one; $y_3$ and $y_5$ are also influenced by exogenous variables. Their error terms are correlated.
- Set three consists of $y_6$ and $y_7$. Each one is influenced by one variable from set two, and one exogenous variable from set zero. They also could have been influenced by any or all the variables in set one, but in this model they are not[6]. Their error terms are correlated.

Notice that the way the rule is stated, all the arrows (curved and double-headed as well as straight) are optional, except for the straight arrows from error terms to the endogenous variables. It would be possible to start drawing the path diagram without any of the optional arrows. Endogenous variables would be placed into sets such that they *definitely are not* influenced, directly or indirectly, by the variables in later sets. Variables with correlated error terms must be placed into the same set. Then the remaining arrows could be added to the picture, based on substantive modelling considerations. One could say that "really," there are arrows to each endogenous variable from all the variables in earlier sets, but some of the coefficients have been set to zero, so the arrows are invisible.

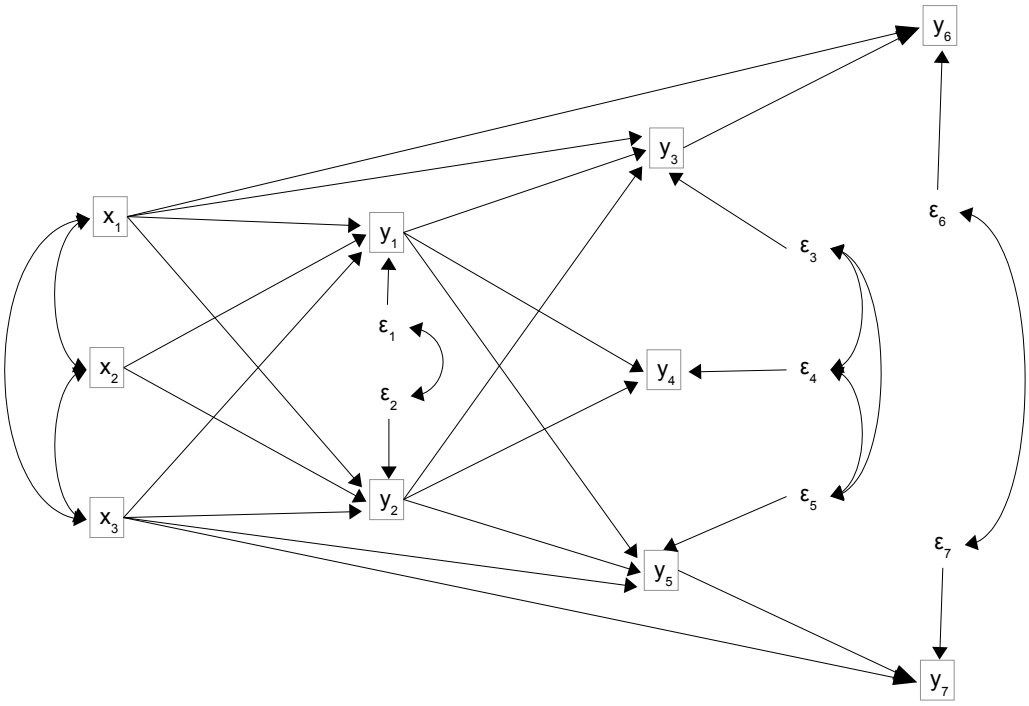
The following lemma will be used to prove the acyclic rule.

> ***Lemma**: In the centered multivariate regression model (4.6), the matrix* $\boldsymbol{\Sigma} = cov\begin{pmatrix}\mathbf{x}_i \\ \mathbf{y}_i\end{pmatrix}$ *has an inverse.*

[5] This condition is satisfied if $\boldsymbol{\Psi}$ is diagonal.

[6] It would have been too hard to draw.

Figure 4.2: An Acyclic Model




**Proof of the lemma** Following the notation of Seber [57], write $\mathbf{\Sigma}$ as a partitioned matrix.

$$\mathbf{\Sigma} = cov\begin{pmatrix}\mathbf{x}_i \\ \hline \mathbf{y}_i\end{pmatrix} = \left(\begin{array}{c|c} \mathbf{\Phi} & \mathbf{\Phi\Gamma}^\top \\ \hline \mathbf{\Gamma\Phi} & \mathbf{\Gamma\Phi\Gamma}^\top + \mathbf{\Psi} \end{array}\right) = \left(\begin{array}{c|c} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \hline \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{array}\right)$$

Seber's expression 14.17(a) on page 296 says that if $\mathbf{A}_{1,1}$ is non-singular,

$$|\mathbf{\Sigma}| = |\mathbf{A}_{1,1}| \; |\mathbf{A}_{2,2} - \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,2}|$$

Since $\mathbf{\Phi} = \mathbf{A}_{1,1}$ is positive definite, its inverse exists and its determinant is positive. Substituting for the other term,

$$\begin{aligned} \mathbf{A}_{2,2} - \mathbf{A}_{2,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,2} &= (\mathbf{\Gamma\Phi\Gamma}^\top + \mathbf{\Psi}) - (\mathbf{\Gamma\Phi})\mathbf{\Phi}^{-1}(\mathbf{\Phi\Gamma}^\top) \\ &= \mathbf{\Gamma\Phi\Gamma}^\top + \mathbf{\Psi} - \mathbf{\Gamma\Phi\Gamma}^\top \\ &= \mathbf{\Psi}, \end{aligned}$$

which has a positive determinant because it is positive definite. Thus the determinant of $\mathbf{\Sigma}$ is positive. It follows that $\mathbf{\Sigma}$ has an inverse. ■

**Proof of the Acyclic Rule** Sub-divide the endogenous variables in $\mathbf{y}_i$. For $j = 1, \ldots, m$, denote the endogenous variables in set $j$ by $\mathbf{y}_{i,j}$, and the corresponding error

terms by $\boldsymbol{\epsilon}_{i,j}$. Consider a set of regression models, in which $\mathbf{y}_{i,j}$ are the response variables. Let $\mathbf{x}_{i,j-1}$ denote the exogenous variables in $\mathbf{x}_0$, plus all the endogenous variables in $\mathbf{y}_{i,\ell}$ for $\ell < j$, pooled. These are the explanatory variables. The model equation(s) may be written

$$\mathbf{y}_{i,j} = \boldsymbol{\Gamma}_j \mathbf{x}_{i,j-1} + \boldsymbol{\epsilon}_{i,j}.$$

The collection of all parameters from these models correspond to the set of parameters of the path model. Some matrix elements may be zero, but that presents no problem.

By the lemma, $cov(\mathbf{x}_{i,j-1})$ is non-singular for $j = 2, \ldots, m$. Each $cov(\boldsymbol{\epsilon}_j)$ is also non-singular. This is true because the zero covariance between sets of error terms gives the overall covariance matrix of the errors a block diagonal structure, and the determinant of a block diagonal matrix is the product of the determinants of the blocks.

Thus, the regression rule applies at each stage, and the parameters in each regression model are identified. This identifies all the parameters of the acyclic path model. ■

Because the parameters of each regression model are *just identifiable*, so are the parameters of the acyclic path model, provided that the model includes all permissible straight arrows and covariances between error terms.

## 4.3 Cyclic Models

Like most of the identifiability rules, the acyclic rule gives a set of sufficient conditions for parameter identifiability. They are not necessary. This is a good thing, because cyclic models – models with one or more feedback loops – sometimes express something we believe to be true, and want to incorporate into the model. Supply and demand in economics is a prime example.

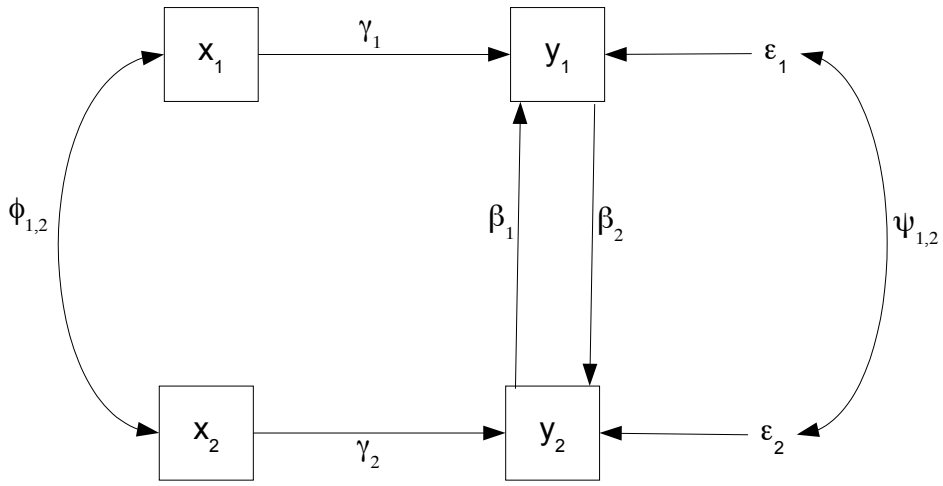
### 4.3.1 Duncan's non-recursive just identified model

Figure 4.3 shows a cyclic model from Chapter 5 of Duncan's *Introduction to Structural Equation Models* [21]. From a quick glance, distinguishing between $\beta_1$, $\beta_2$ and $\psi_{1,2}$ would seem an impossibility, but this kind of intuition can be unreliable.

It's always a good idea to check the parameter count rule first. There are three unique $\phi_{i,j}$, two $\gamma_j$, two $\beta_j$ and three unique $\psi_{i,j}$, for a total of ten parameters. The covariance matrix of the observable variables has $4(4+1)/2 = 10$ unique elements, so if the parameters are identifiable, they are just identifiable.

Duncan spends most of a chapter solving the covariance structure equations, and he does not really finish the job. It's preferable to use Sage. This avoids most of the work, but it's still a fairly big job. Readers who are not interested in the details of how Sage works may want to skip the rest of this section.

Figure 4.3: Duncan's Cyclic Model




The model equations are

$$\begin{array}{ccccccccc} \mathbf{y}_i & = & \boldsymbol{\beta} & \mathbf{y}_i & + & \boldsymbol{\Gamma} & \mathbf{x}_i & + & \boldsymbol{\epsilon}_i \\ \begin{pmatrix} y_{i,1} \\ y_{i,2} \end{pmatrix} & = & \begin{pmatrix} 0 & \beta_1 \\ \beta_2 & 0 \end{pmatrix} & \begin{pmatrix} y_{i,1} \\ y_{i,2} \end{pmatrix} & + & \begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix} & \begin{pmatrix} x_{i,1} \\ x_{i,2} \end{pmatrix} & + & \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \end{pmatrix}. \end{array} \tag{4.8}$$

The first operation is to load the `sem` package. In the work that follows, note that if a function has any capital letters, it's part of the package[7]. If it's all lower case, it's a generic Sage function.

```
# Duncan's (1975) just identified non-recursive model
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
```

evaluate

Now set up the parameter matrices. Observe that the displayed matrices agree with (4.8).

```
# Set up Phi, Gamma, Beta, Psi
PHI = SymmetricMatrix(2,'phi'); show(PHI)
GAMMA = DiagonalMatrix(2,'gamma'); show(GAMMA)
BETA = ZeroMatrix(2,2)
BETA[0,1] = var('beta1'); BETA[1,0] = var('beta2'); show(BETA)
PSI = SymmetricMatrix(2,'psi'); show(PSI)
```

[7] Calling `Contents()` (without any arguments) lists the functions in the `sem` package.

evaluate

$$\begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}$$

$$\begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix}$$

$$\begin{pmatrix} 0 & \beta_1 \\ \beta_2 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \psi_{11} & \psi_{12} \\ \psi_{12} & \psi_{22} \end{pmatrix}$$

The next step is to calculate $\boldsymbol{\Sigma}$, the covariance matrix of the observable variables. It would be a fairly big job by hand. In Sage, there is a scrollbar that lets you view the whole matrix. Here, it is set in a tiny typeface and it still does not fit on the page.

```
# Calculate the covariance matrix.
Sigma = PathCov(Phi=PHI,Beta=BETA,Gamma=GAMMA,Psi=PSI)
show(Sigma)
```

evaluate

$$\begin{pmatrix} \phi_{11} & \phi_{12} & -\frac{\beta_1\gamma_2\phi_{12}+\gamma_1\phi_{11}}{\beta_1\beta_2-1} & -\frac{\beta_2\gamma_1\phi_{11}+\gamma_2\phi_{12}}{\beta_1\beta_2-1} \\ \phi_{12} & \phi_{22} & -\frac{\beta_1\gamma_2\phi_{22}+\gamma_1\phi_{12}}{\beta_1\beta_2-1} & -\frac{\beta_2\gamma_1\phi_{12}+\gamma_2\phi_{22}}{\beta_1\beta_2-1} \\ -\frac{\beta_1\gamma_2\phi_{12}+\gamma_1\phi_{11}}{\beta_1\beta_2-1} & -\frac{\beta_1\gamma_2\phi_{22}+\gamma_1\phi_{12}}{\beta_1\beta_2-1} & \frac{\beta_1^2\gamma_2^2\phi_{22}+2\,\beta_1\gamma_1\gamma_2\phi_{12}+\gamma_1^2\phi_{11}+\beta_1^2\psi_{22}+2\,\beta_1\psi_{12}+\psi_{11}}{(\beta_1\beta_2-1)^2} & \frac{\beta_1\beta_2\gamma_1\gamma_2\phi_{12}+\beta_2\gamma_1^2\phi_{11}+\beta_1\gamma_2^2\phi_{22}+\gamma_1\gamma_2\phi_{12}+\beta_1\beta_2\psi_{12}+\cdots}{(\beta_1\beta_2-1)^2} \\ -\frac{\beta_2\gamma_1\phi_{11}+\gamma_2\phi_{12}}{\beta_1\beta_2-1} & -\frac{\beta_2\gamma_1\phi_{12}+\gamma_2\phi_{22}}{\beta_1\beta_2-1} & \frac{\beta_1\beta_2\gamma_1\gamma_2\phi_{12}+\beta_2\gamma_1^2\phi_{11}+\beta_1\gamma_2^2\phi_{22}+\gamma_1\gamma_2\phi_{12}+\beta_1\beta_2\psi_{11}+\beta_2\psi_{11}+\beta_1\psi_{22}+\psi_{12}}{(\beta_1\beta_2-1)^2} & \frac{\beta_2^2\gamma_1^2\phi_{11}+2\,\beta_2\gamma_1\gamma_2\phi_{12}+\gamma_2^2\phi_{22}+\beta_2^2\psi_{11}+2\,\beta_2\psi_{12}+\psi_{22}}{(\beta_1\beta_2-1)^2} \end{pmatrix}$$

Clearly, solving the ten equations in 10 unknowns by hand would not be easy, though it's possible since Duncan did it. The importance of $\beta_1\beta_2 \neq 1$ is also clear, because $\beta_1\beta_2 - 1$ is in most of the denominators. This quantity is the determinant of $\mathbf{I} - \boldsymbol{\beta}$. It is guaranteed not to equal zero by Theorem 1.1 on page 146. Thus, the covariance matrix exists, and the set of parameter vectors satisfying $\beta_1\beta_2 = 1$ defines a surface that is interior to the parameter space, but not part of it — a sort of hole in the parameter space.

Since the number of covariance structure equations is the same as the number of unknowns in this case, Sage's `solve` function is a good option[8]. The first task is to assemble a list of model parameters. The `Parameters` function from the `sem` package returns a list of the unique elements of a matrix that are not one or zero. This is a good alternative to typing them in.

[8]When the number of equations exceeds the number of unknowns, as is usually the case, the two main options are setting some of the equations aside, or using the Groebner basis methods of Chapter **??**.

```
# Solve 10 equations in 10 unknowns
# Assemble list of parameters
param = Parameters(PHI)
param.extend(Parameters(GAMMA)); param.extend(Parameters(BETA))
param.extend(Parameters(PSI)); param
```

evaluate

$$[\phi_{11}, \phi_{12}, \phi_{22}, \gamma_1, \gamma_2, \beta_1, \beta_2, \psi_{11}, \psi_{12}, \psi_{22}]$$

Assembling the list of equations to solve consists of going through the unique elements of $\mathbf{\Sigma}$, and setting each expression to a $\sigma_{ij}$. The `SetupEqns` function takes care of this task.

```
# Set up equations to solve
eqns = SetupEqns(Sigma)
for item in eqns: show(item)
```

evaluate

$$\phi_{11} = \sigma_{11}$$

$$\phi_{12} = \sigma_{12}$$

$$-\frac{\beta_1\gamma_2\phi_{12}+\gamma_1\phi_{11}}{\beta_1\beta_2-1} = \sigma_{13}$$

$$-\frac{\beta_2\gamma_1\phi_{11}+\gamma_2\phi_{12}}{\beta_1\beta_2-1} = \sigma_{14}$$

$$\phi_{22} = \sigma_{22}$$

$$-\frac{\beta_1\gamma_2\phi_{22}+\gamma_1\phi_{12}}{\beta_1\beta_2-1} = \sigma_{23}$$

$$-\frac{\beta_2\gamma_1\phi_{12}+\gamma_2\phi_{22}}{\beta_1\beta_2-1} = \sigma_{24}$$

$$\frac{\beta_1^2\gamma_2^2\phi_{22}+2\,\beta_1\gamma_1\gamma_2\phi_{12}+\gamma_1^2\phi_{11}+\beta_1^2\psi_{22}+2\,\beta_1\psi_{12}+\psi_{11}}{(\beta_1\beta_2-1)^2} = \sigma_{33}$$

$$\frac{\beta_1\beta_2\gamma_1\gamma_2\phi_{12}+\beta_2\gamma_1^2\phi_{11}+\beta_1\gamma_2^2\phi_{22}+\gamma_1\gamma_2\phi_{12}+\beta_1\beta_2\psi_{12}+\beta_2\psi_{11}+\beta_1\psi_{22}+\psi_{12}}{(\beta_1\beta_2-1)^2} = \sigma_{34}$$

$$\frac{\beta_2^2\gamma_1^2\phi_{11}+2\,\beta_2\gamma_1\gamma_2\phi_{12}+\gamma_2^2\phi_{22}+\beta_2^2\psi_{11}+2\,\beta_2\psi_{12}+\psi_{22}}{(\beta_1\beta_2-1)^2} = \sigma_{44}$$

Now try to solve the equations. The `solve` function returns a list of solutions (a list of lists), so the length of the result should be the number of solutions. Naturally, we are hoping for the length to be one.

```
# Try to solve
solut = solve(eqns,param); len(solut)
```

evaluate

10

```
# Oh wow, 10?
for item in solut: show(item)
```

evaluate

$\phi_{11} = \sigma_{11}$

$\phi_{12} = \sigma_{12}$

$-\frac{\beta_1\gamma_2\phi_{12}+\gamma_1\phi_{11}}{\beta_1\beta_2-1} = \sigma_{13}$

$-\frac{\beta_2\gamma_1\phi_{11}+\gamma_2\phi_{12}}{\beta_1\beta_2-1} = \sigma_{14}$

$\phi_{22} = \sigma_{22}$

$-\frac{\beta_1\gamma_2\phi_{22}+\gamma_1\phi_{12}}{\beta_1\beta_2-1} = \sigma_{23}$

$-\frac{\beta_2\gamma_1\phi_{12}+\gamma_2\phi_{22}}{\beta_1\beta_2-1} = \sigma_{24}$

$\frac{\beta_1^2\gamma_2^2\phi_{22}+2\,\beta_1\gamma_1\gamma_2\phi_{12}+\gamma_1^2\phi_{11}+\beta_1^2\psi_{22}+2\,\beta_1\psi_{12}+\psi_{11}}{(\beta_1\beta_2-1)^2} = \sigma_{33}$

$\frac{\beta_1\beta_2\gamma_1\gamma_2\phi_{12}+\beta_2\gamma_1^2\phi_{11}+\beta_1\gamma_2^2\phi_{22}+\gamma_1\gamma_2\phi_{12}+\beta_1\beta_2\psi_{12}+\beta_2\psi_{11}+\beta_1\psi_{22}+\psi_{12}}{(\beta_1\beta_2-1)^2} = \sigma_{34}$

$\frac{\beta_2^2\gamma_1^2\phi_{11}+2\,\beta_2\gamma_1\gamma_2\phi_{12}+\gamma_2^2\phi_{22}+\beta_2^2\psi_{11}+2\,\beta_2\psi_{12}+\psi_{22}}{(\beta_1\beta_2-1)^2} = \sigma_{44}$

Sage just returned the original ten equations; those were the ten items in the list. I was confused. But according to a post on `ask.sagemath.org`, this happens when `solve` can't solve the problem. Come to think of it, this is also what it does when it can't evaluate an integral. The post suggests that if the equations are polynomials, try the option `to_poly_solve=True` on `solve`. Now, our equations are not polynomials, but they will be if we multiply through by the denominators. A disadvantage of doing this is that it may introduce false solutions that hold when the denominators are zero. As long as one is aware of this and willing to take care of it, it's okay. Let us proceed.

```
# Multiply through by denominators
eqns[2] = eqns[2]*(beta1*beta2-1)
eqns[3] = eqns[3]*(beta1*beta2-1)
eqns[5] = eqns[5]*(beta1*beta2-1)
eqns[6] = eqns[6]*(beta1*beta2-1)
eqns[7] = eqns[7]*(beta1*beta2-1)^2
eqns[8] = eqns[8]*(beta1*beta2-1)^2
eqns[9] = eqns[9]*(beta1*beta2-1)^2
for item in eqns: show(item)
```

evaluate

$\phi_{11} = \sigma_{11}$

$\phi_{12} = \sigma_{12}$

$-\beta_1\gamma_2\phi_{12} - \gamma_1\phi_{11} = (\beta_1\beta_2 - 1)\sigma_{13}$

$-\beta_2\gamma_1\phi_{11} - \gamma_2\phi_{12} = (\beta_1\beta_2 - 1)\sigma_{14}$

$\phi_{22} = \sigma_{22}$

$-\beta_1\gamma_2\phi_{22} - \gamma_1\phi_{12} = (\beta_1\beta_2 - 1)\sigma_{23}$

$-\beta_2\gamma_1\phi_{12} - \gamma_2\phi_{22} = (\beta_1\beta_2 - 1)\sigma_{24}$

$\beta_1^2\gamma_2^2\phi_{22} + 2\,\beta_1\gamma_1\gamma_2\phi_{12} + \gamma_1^2\phi_{11} + \beta_1^2\psi_{22} + 2\,\beta_1\psi_{12} + \psi_{11} = (\beta_1\beta_2 - 1)^2\sigma_{33}$

$\beta_1\beta_2\gamma_1\gamma_2\phi_{12}+\beta_2\gamma_1^2\phi_{11}+\beta_1\gamma_2^2\phi_{22}+\gamma_1\gamma_2\phi_{12}+\beta_1\beta_2\psi_{12}+\beta_2\psi_{11}+\beta_1\psi_{22}+\psi_{12} = (\beta_1\beta_2 - 1)^2\sigma_{34}$

$\beta_2^2\gamma_1^2\phi_{11} + 2\,\beta_2\gamma_1\gamma_2\phi_{12} + \gamma_2^2\phi_{22} + \beta_2^2\psi_{11} + 2\,\beta_2\psi_{12} + \psi_{22} = (\beta_1\beta_2 - 1)^2\sigma_{44}$

Trying the `to_poly_solve=True` option,

```
# Now they are polynomial equations
solut = solve(eqns,param, to_poly_solve=True); len(solut)
```

evaluate

4

Now there seem to be four solutions, which is promising. Let us examine them, one at a time.

```
# First solution
for item in solut[0]: show(item)
```

evaluate

$\beta_1 = \frac{1}{c_{75}}$

$\beta_2 = c_{75}$

$\gamma_1 = 0$

$\gamma_2 = 0$

$\phi_{11} = \sigma_{11}$

$\phi_{12} = \sigma_{12}$

$\phi_{22} = \sigma_{22}$

$\psi_{11} = c_{76}$

$\psi_{12} = c_{77}$

$\psi_{22} = -c_{75}^2 c_{76} - 2\, c_{75} c_{77}$

This is not a single solution, but an infinite set of solutions. The constant $c_{75}$, which appears in the first two lines, has not been seen before. It can be anything as long as it is not zero. Regardless of what $c_{75}$ happens to be, the first two lines dictate that $\beta_1 \beta_2 = 1$. So this is a false solution, introduced when we multiplied through by denominators. There are other strange things about it (like $\gamma_1 = \gamma_2 = 0$), but it may be discarded without further consideration.

```
# Second solution
for item in solut[1]: show(item)
```

evaluate

$\beta_1 = \frac{\sigma_{13}}{\sigma_{14}}$

$\beta_2 = \frac{\sigma_{14}}{\sigma_{13}}$

$\gamma_1 = 0$

$\gamma_2 = 0$

$\phi_{11} = \sigma_{11}$

$\phi_{12} = \sigma_{12}$

$\phi_{22} = \sigma_{22}$

$\psi_{11} = c_{78}$

$\psi_{12} = -\frac{c_{78}\sigma_{14}}{\sigma_{13}}$

$\psi_{22} = \frac{c_{78}\sigma_{14}^2}{\sigma_{13}^2}$

This infinite family of solutions also implies $\beta_1 \beta_2 = 1$. Again, it is an artifact of multiplying by denominators, and may be discarded.

```
# Third solution
for item in solut[2]: show(item)
```

evaluate

$\beta_1 = \frac{\sigma_{12}\sigma_{23}}{\sigma_{14}\sigma_{22}}$

$\beta_2 = \frac{\sigma_{14}\sigma_{22}}{\sigma_{12}\sigma_{23}}$

$\gamma_1 = 0$

$\gamma_2 = 0$

$\phi_{11} = \sigma_{11}$

$\phi_{12} = \sigma_{12}$

$$\phi_{22} = \sigma_{22}$$

$$\psi_{11} = \frac{\sigma_{14}^2\sigma_{22}^2\sigma_{33} - 2\,\sigma_{12}\sigma_{14}\sigma_{22}\sigma_{23}\sigma_{34} + \sigma_{12}^2\sigma_{23}^2\sigma_{44}}{\sigma_{14}^2\sigma_{22}^2}$$

$$\psi_{12} = -\frac{\sigma_{14}^2\sigma_{22}^2\sigma_{33} - 2\,\sigma_{12}\sigma_{14}\sigma_{22}\sigma_{23}\sigma_{34} + \sigma_{12}^2\sigma_{23}^2\sigma_{44}}{\sigma_{12}\sigma_{14}\sigma_{22}\sigma_{23}}$$

$$\psi_{22} = \frac{\sigma_{14}^2\sigma_{22}^2\sigma_{33} - 2\,\sigma_{12}\sigma_{14}\sigma_{22}\sigma_{23}\sigma_{34} + \sigma_{12}^2\sigma_{23}^2\sigma_{44}}{\sigma_{12}^2\sigma_{23}^2}$$

At least this one is a single solution and not an infinite family, but again, multipkying the expressions for $\beta_1$ and $\beta_2$ yields one, so it's a false solution that may be discarded.

```
# Fourth and last solution.
for item in solut[3]: show(item)
```

evaluate

$$\beta_1 = \frac{\sigma_{12}\sigma_{13} - \sigma_{11}\sigma_{23}}{\sigma_{12}\sigma_{14} - \sigma_{11}\sigma_{24}}$$

$$\beta_2 = \frac{\sigma_{14}\sigma_{22} - \sigma_{12}\sigma_{24}}{\sigma_{13}\sigma_{22} - \sigma_{12}\sigma_{23}}$$

$$\gamma_1 = \frac{\sigma_{14}\sigma_{23} - \sigma_{13}\sigma_{24}}{\sigma_{12}\sigma_{14} - \sigma_{11}\sigma_{24}}$$

$$\gamma_2 = -\frac{\sigma_{14}\sigma_{23} - \sigma_{13}\sigma_{24}}{\sigma_{13}\sigma_{22} - \sigma_{12}\sigma_{23}}$$

$$\phi_{11} = \sigma_{11}$$

$$\phi_{12} = \sigma_{12}$$

$$\phi_{22} = \sigma_{22}$$

$$\psi_{11} = \frac{\left(\sigma_{24}^2\sigma_{33} - 2\,\sigma_{23}\sigma_{24}\sigma_{34} + \sigma_{23}^2\sigma_{44}\right)\sigma_{11}^2 + \left(\sigma_{14}^2\sigma_{33} - 2\,\sigma_{13}\sigma_{14}\sigma_{34} + \sigma_{13}^2\sigma_{44}\right)\sigma_{12}^2 - \left(\sigma_{14}^2\sigma_{23}^2 - 2\,\sigma_{13}\sigma_{14}\sigma_{23}\sigma_{24} + \sigma_{13}^2\sigma_{24}^2 - 2\,((\sigma_{24}\sigma_{34} - \sigma_{23}\sigma_{44})\sigma_{13}}{\sigma_{12}^2\sigma_{14}^2 - 2\,\sigma_{11}\sigma_{12}\sigma_{14}\sigma_{24} + \sigma_{11}^2\sigma_{24}^2}$$

$$\psi_{12} = -\frac{((\sigma_{24}\sigma_{34} - \sigma_{23}\sigma_{44})\sigma_{13} - (\sigma_{24}\sigma_{33} - \sigma_{23}\sigma_{34})\sigma_{14})\sigma_{12}^2 + \left((\sigma_{24}\sigma_{34} - \sigma_{23}\sigma_{44})\sigma_{13}\sigma_{22} - (\sigma_{24}\sigma_{33} - \sigma_{23}\sigma_{34})\sigma_{14}\sigma_{22} + \left(\sigma_{24}^2\sigma_{33} - 2\,\sigma_{23}\sigma_{24}\sigma_{34} + \sigma_{23}^2\right.\right.}{\sigma_{12}\sigma_{13}\sigma_{14}\sigma_{22} - \sigma_{12}^2\sigma_{14}\sigma_{23} - (\sigma_{13}\sigma_{22}\sigma_{24} - \sigma_{12}\sigma_{23}\sigma_{2}}$$

$$\psi_{22} = \frac{\left(\sigma_{24}^2\sigma_{33} - 2\,\sigma_{23}\sigma_{24}\sigma_{34} + \sigma_{23}^2\sigma_{44}\right)\sigma_{12}^2 - \left(\sigma_{22}\sigma_{24}^2 - \sigma_{22}^2\sigma_{44}\right)\sigma_{13}^2 + 2\left(\sigma_{22}\sigma_{23}\sigma_{24} - \sigma_{22}^2\sigma_{34}\right)\sigma_{13}\sigma_{14} - \left(\sigma_{22}\sigma_{23}^2 - \sigma_{22}^2\sigma_{33}\right)\sigma_{14}^2 + 2\,((\sigma_{24}\sigma_{34} - \sigma_{23}\sigma_{4}}{\sigma_{13}^2\sigma_{22}^2 - 2\,\sigma_{12}\sigma_{13}\sigma_{22}\sigma_{23} + \sigma_{12}^2\sigma_{23}^2}$$

Well, at least the numerator of $\beta_1$ is not the same as the denominator of $\beta_2$, so this one has a chance. In fact, the first four solutions agree with Duncan [21], pp. 69 and 70. Duncan does not give solutions for the last three parameters, instead arguing that the solutions exist. One can see why he gave up; the last three expressions are horrendous. However, maybe they can be simplified. In order to work with the results more easily, it is helpful to obtain the solutions in the form of a dictionary.

```
# Obtain solutions as dictionaries
solud = solve(eqns,param, to_poly_solve=True ,solution_dict=True)
len(solud)
```

evaluate

4

It's the same four solutions, but in the form of dictionaries rather than lists. A Sage dictionary is really just a Python dictionary. In this case, the *key* is the parameter for which the equations were solved, so `solution[key]` gives the solution in terms of the $\sigma_{ij}$ values. For example,

```
# First extract the solution we want.
sol = solud[3]
sol[beta1] # beta1 acts like an index in an array
```

evaluate

$$\frac{\sigma_{12}\sigma_{13}-\sigma_{11}\sigma_{23}}{\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24}}$$

There is another way to get the same thing.

```
# Another way: beta1 as a function of the dictionary
beta1(sol)
```

evaluate

$$\frac{\sigma_{12}\sigma_{13}-\sigma_{11}\sigma_{23}}{\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24}}$$

The advantage of the second way (asking for what you want as a function of the dictionary, with curved parentheses rather than square brackets) is that you can give it an expression in the parameters, rather than just a single parameter.

```
# Evaluate an expression
(beta1*beta2)(sol)
```

evaluate

$$\frac{(\sigma_{12}\sigma_{13}-\sigma_{11}\sigma_{23})(\sigma_{14}\sigma_{22}-\sigma_{12}\sigma_{24})}{(\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24})(\sigma_{13}\sigma_{22}-\sigma_{12}\sigma_{23})}$$

Well, at least it did not evaluate to one. We shall return to the product $\beta_1\beta_2$ later, but first let's try to simplify those solutions for $\psi_{ij}$. Sage's `factor` function will try to factor both numerator and denominator, potentially resulting in some cancellations. In general, `factor` does other good things too; it's much more useful than `simplify`. We may as well apply `factor` to all the items in the dictionary. It should affect only the $\psi_{ij}$, but you can never tell[9], and it can do no harm.

[9] Well, of course you can, but not without some effort.

```
# Factor the solution. Double = is for display, not assignment
for item in param: show( item == factor(sol[item]) )
```

evaluate

$$\beta_1 = \frac{\sigma_{12}\sigma_{13}-\sigma_{11}\sigma_{23}}{\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24}}$$

$$\beta_2 = \frac{\sigma_{14}\sigma_{22}-\sigma_{12}\sigma_{24}}{\sigma_{13}\sigma_{22}-\sigma_{12}\sigma_{23}}$$

$$\gamma_1 = \frac{\sigma_{14}\sigma_{23}-\sigma_{13}\sigma_{24}}{\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24}}$$

$$\gamma_2 = -\frac{\sigma_{14}\sigma_{23}-\sigma_{13}\sigma_{24}}{\sigma_{13}\sigma_{22}-\sigma_{12}\sigma_{23}}$$

$$\phi_{11} = \sigma_{11}$$

$$\phi_{12} = \sigma_{12}$$

$$\phi_{22} = \sigma_{22}$$

$$\psi_{11} = -\frac{\sigma_{11}\sigma_{14}^2\sigma_{23}^2-2\,\sigma_{11}\sigma_{13}\sigma_{14}\sigma_{23}\sigma_{24}+\sigma_{11}\sigma_{13}^2\sigma_{24}^2-\sigma_{12}^2\sigma_{14}^2\sigma_{33}+2\,\sigma_{11}\sigma_{12}\sigma_{14}\sigma_{24}\sigma_{33}-\sigma_{11}^2\sigma_{24}^2\sigma_{33}+2\,\sigma_{12}^2\sigma_{13}\sigma_{14}\sigma_{34}-2\,\sigma_{11}\sigma_{12}\sigma_{14}\sigma_{23}\sigma_{34}-2\,\sigma_{1}}{(\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24})^2}$$

$$\psi_{12} = \frac{\sigma_{12}\sigma_{14}^2\sigma_{23}^2-2\,\sigma_{12}\sigma_{13}\sigma_{14}\sigma_{23}\sigma_{24}+\sigma_{12}\sigma_{13}^2\sigma_{24}^2-\sigma_{12}\sigma_{14}^2\sigma_{22}\sigma_{33}+\sigma_{12}^2\sigma_{14}\sigma_{24}\sigma_{33}+\sigma_{11}\sigma_{14}\sigma_{22}\sigma_{24}\sigma_{33}-\sigma_{11}\sigma_{12}\sigma_{24}^2\sigma_{33}+2\,\sigma_{12}\sigma_{13}\sigma_{14}\sigma_{22}\sigma_{34}-\sigma_{1}^2}{(\sigma_{12}\sigma_{14}}$$

$$\psi_{22} = -\frac{\sigma_{14}^2\sigma_{22}\sigma_{23}^2-2\,\sigma_{13}\sigma_{14}\sigma_{22}\sigma_{23}\sigma_{24}+\sigma_{13}^2\sigma_{22}\sigma_{24}^2-\sigma_{14}^2\sigma_{22}^2\sigma_{33}+2\,\sigma_{12}\sigma_{14}\sigma_{22}\sigma_{24}\sigma_{33}-\sigma_{12}^2\sigma_{24}^2\sigma_{33}+2\,\sigma_{13}\sigma_{14}\sigma_{22}^2\sigma_{34}-2\,\sigma_{12}\sigma_{14}\sigma_{22}\sigma_{23}\sigma_{34}-2\,\sigma_{1}}{(\sigma_{13}\sigma_{22}-\sigma_{12}\sigma_{23})^2}$$

The numerators of the $\psi_{ij}$ are still awful, but it appears that the factoring helped in the denominators. The expressions run off the page and we don't have the Sage scrollbar, but it's possible to take a look at just the denominator of $\psi_{12}$.

```
# Factor the denominator of psi_12
factor(denominator(sol[psi12]))
```

evaluate

$$-(\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24})(\sigma_{13}\sigma_{22}-\sigma_{12}\sigma_{23})$$

Now we have something helpful. When solutions to covariance structure equations are fractions, it's important to find out if the denominators can be zero, or more precisely, for what parameter values they can be zero. All that's necessary is to substitute for the $\sigma_{ij}$ in terms of the original model parameters and simplify. For the present problem, it seems that we only need to check two quantities: $\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24}$ (the denominator of the solution for $\beta_1$) and $\sigma_{13}\sigma_{22}-\sigma_{12}\sigma_{23}$, the denominator of the solution for $\beta_2$.

In this case, the functions of $\sigma_{ij}$ in the denominators are small, and it would not be too much trouble to type them in. However, sometimes they can be quite messy. The `sem` package has a function called `SigmaOfTheta`, which goes through the unique elements of a symbolic covariance matrix and makes a dictionary in which by default the keys are the symbols $\sigma_{ij}$, and the entries are the symbolic expressions in the corresponding cells of the

matrix. I like to call the resulting dictionary `theta`. Then, one can easily evaluate any any expression in the $\sigma_{ij}$ as a function of $\boldsymbol{\theta}$, the vector of model parameters.

```
# Make dictionary theta
theta = SigmaOfTheta(Sigma)
```

evaluate

Now evaluate the two denominators at the model parameters.

```
denom1 = denominator(sol[beta1]); show(denom1)
denom2 = denominator(sol[beta2]); show(denom2)
```

evaluate

$\sigma_{12}\sigma_{14} - \sigma_{11}\sigma_{24}$

$\sigma_{13}\sigma_{22} - \sigma_{12}\sigma_{23}$

```
# First denominator as a function of the parameters
show(factor(denom1(theta)))
```

evaluate

$-\frac{\left(\phi_{12}^2-\phi_{11}\phi_{22}\right)\gamma_2}{\beta_1\beta_2-1}$

Well, well. The denominator (of this denominator) is guaranteed to be non-zero, so that's no problem. $\phi_{12}^2 - \phi_{11}\phi_{22}$ is minus the determinant of $\boldsymbol{\Phi} = cov(\mathbf{x}_i)$, which is non-zero everywhere in the parameter space because $\boldsymbol{\Phi}$ is positive definite. The only points at which identifiability of $\beta_1$ (and $\gamma_1$ and $\psi_{11}$ and $\psi_{12}$) fails are the ones with $\gamma_2 = 0$. By symmetry, one would expect something similar for the second denominator.

```
# Second denominator as a function of the parameters
show(factor(denom2(theta)))
```

evaluate

$\frac{\left(\phi_{12}^2-\phi_{11}\phi_{22}\right)\gamma_1}{\beta_1\beta_2-1}$

As expected, the second denominator will be non-zero and identifiability will hold as long as $\gamma_1 \neq 0$. This means the entire parameter vector is identifiable everywhere in the parameter space, provided $\gamma_1 \neq 0$ and $\gamma_2 \neq 0$. Glancing back at Figure 4.3, this makes perfect sense. Suppose that $\gamma_1 = \gamma_2 = 0$, so that the paths from $x_1$ to $y_1$ and $x_2$ to $y_2$ are missing. This means the $x$ and $y$ variables are independent, forming separate systems.

The sub-model based on $y_1$ and $y_2$ would have covariance structure equations with three equations in five parameters, so that identifiability is ruled out by the parameter count rule. Also consider the false solutions created by multiplying the original covariance structure equations by powers of $\beta_1\beta_2 - 1$ in order to clear the denominators. All three false solutions (families of false solutions) have $\gamma_1 = \gamma_2 = 0$. We definitely need both $\gamma_j$ parameters to be non-zero.

It's clear that the dictionary produced by `SigmaOfTheta` is very useful. Another thing it's good for is checking solutions, something we really should do for completeness. The process is to take a proposed solution for a parameter in terms of $\sigma_{ij}$ quantities, substitute for the $\sigma_{ij}$ in terms of the computed values in $\mathbf{\Sigma}$, and then simplifying to obtain the parameter in question. For the $\psi ij$ parameters, it's just too tedious to do by hand. We might as well put the whole thing in a loop. For each parameter, a tuple is produced. The first item is the parameter being checked, and the second item is the result of substituting for the $\sigma_{ij}$ quantities in the solution. If everything is okay, they should be equal.

```
# Check the solutions of the covariance structure equations.
for item in param:
    solution = sol[item]
    backsub = solution(theta) # Solution in terms of theta
    show(item, factor(backsub))
```

evaluate

$(\phi_{11}, \phi_{11})$

$(\phi_{12}, \phi_{12})$

$(\phi_{22}, \phi_{22})$

$(\gamma_1, \gamma_1)$

$(\gamma_2, \gamma_2)$

$(\beta_1, \beta_1)$

$(\beta_2, \beta_2)$

$(\psi_{11}, \psi_{11})$

$(\psi_{12}, \psi_{12})$

$(\psi_{22}, \psi_{22})$

The solution is verified. As a final note, the sometimes onerous task of checking for false solutions may also be done automatically in a loop. It's necessary to have the solutions in the form of dictionaries rather than lists.

```
# Automate the check for false solutions
for j in range(len(solud)):     # j = 0,1,2,3
    detIminusBeta = (1-beta1*beta2)(solud[j])
    detIminusBeta
```

evaluate

0

0

0

$$-\frac{(\sigma_{12}\sigma_{13}-\sigma_{11}\sigma_{23})(\sigma_{14}\sigma_{22}-\sigma_{12}\sigma_{24})}{(\sigma_{12}\sigma_{14}-\sigma_{11}\sigma_{24})(\sigma_{13}\sigma_{22}-\sigma_{12}\sigma_{23})}+1$$

The last quantity is the current `detIminusBeta`. Just checking,

```
factor(detIminusBeta(theta))
```

evaluate

$$-\beta_1\beta_2+1$$

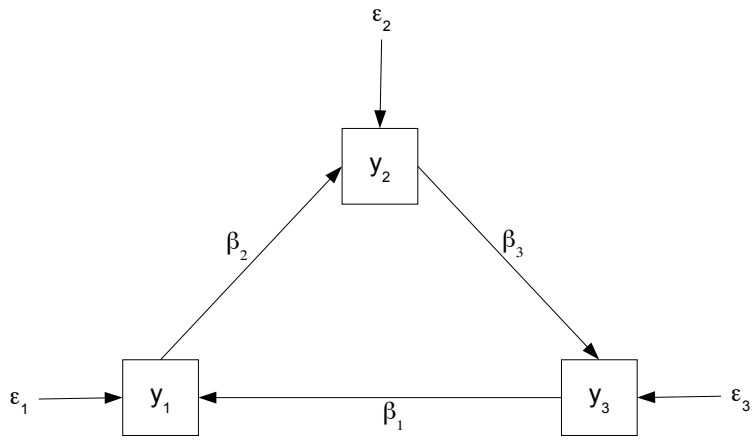
In this case, $|\mathbf{I}-\boldsymbol{\beta}|$ is nice and small, but the strategy illustrated here also applies to larger and more realistic models. If the model has a $\boldsymbol{\beta} \neq \mathbf{0}$ (endogenous variables influencing other endogenous variables), the existence of $(\mathbf{I}-\boldsymbol{\beta})^{-1}$ is guaranteed, but using it in the computation of $\boldsymbol{\Sigma}$ will result in some variances and covariances being fractions. Multiplying both sides of the equations involved by the denominators will result in a system of polynomial equations, which are easier to solve. However, the multiplication by denominators will usually induce false solutions, in which one or more denominators of the original covariance structure equations are zero. You can locate these false solutions easily by using Sage's `det` function to compute the determinant of $\mathbf{I}-\boldsymbol{\beta}$, which may be a messy expression, and then calculating it for each solution, as shown above. If it's zero, discard the solution.

### 4.3.2 The Triangle model

The parameters of cyclic models are not identifiable in general, and I do not know of any identifiability rules. They need to be investigated on a case by case basis. As with Duncan's model, the expressions involved can be messy, and Sage (or any other computer algebra system, really) is a useful tool.

Figure 4.4 shows the path diagram of a cyclic model with three endogenous variables no exogenous variables. Causal influence just keeps going around and around forever.

Figure 4.4: The Triangle Model




In scalar form, the model equations are

$$\begin{aligned} y_1 &= \beta_1 y_3 + \epsilon_1 \\ y_2 &= \beta_2 y_1 + \epsilon_2 \\ y_3 &= \beta_3 y_2 + \epsilon_3. \end{aligned}$$

In matrix form, the equations are

$$\begin{array}{ccccccc} \mathbf{y}_i & = & \boldsymbol{\beta} & & \mathbf{y}_i & + & \boldsymbol{\epsilon}_i \end{array}$$

$$\begin{pmatrix} y_{i,1} \\ y_{i,2} \\ y_{i,3} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \beta_1 \\ \beta_2 & 0 & 0 \\ 0 & \beta_3 & 0 \end{pmatrix} \begin{pmatrix} y_{i,1} \\ y_{i,2} \\ y_{i,3} \end{pmatrix} + \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \\ \epsilon_{i,3} \end{pmatrix}.$$

First we load the `sem` package and set up the model matrices.

```
# load sem package, set up  Beta, Psi
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
BETA = ZeroMatrix(3,3)
BETA[0,2] = var('beta1'); BETA[1,0] = var('beta2')
BETA[2,1] = var('beta3'); show(BETA)
PSI = DiagonalMatrix(3,'psi'); show(PSI)
```

evaluate

$$\begin{pmatrix} 0 & 0 & \beta_1 \\ \beta_2 & 0 & 0 \\ 0 & \beta_3 & 0 \end{pmatrix} \begin{pmatrix} \psi_1 & 0 & 0 \\ 0 & \psi_2 & 0 \\ 0 & 0 & \psi_3 \end{pmatrix}$$

The `sem` package has a special function to calculate the covariance matrix for models with no exogenous variables.

```
Sigma = NoGammaCov(BETA,PSI)
show(Sigma)
```

evaluate

$$\begin{pmatrix} \frac{\beta_1^2\beta_3^2\psi_2+\beta_1^2\psi_3+\psi_1}{(\beta_1\beta_2\beta_3-1)^2} & \frac{\beta_1^2\beta_2\psi_3+\beta_1\beta_3\psi_2+\beta_2\psi_1}{(\beta_1\beta_2\beta_3-1)^2} & \frac{\beta_1\beta_3^2\psi_2+\beta_2\beta_3\psi_1+\beta_1\psi_3}{(\beta_1\beta_2\beta_3-1)^2} \\ \frac{\beta_1^2\beta_2\psi_3+\beta_1\beta_3\psi_2+\beta_2\psi_1}{(\beta_1\beta_2\beta_3-1)^2} & \frac{\beta_1^2\beta_2^2\psi_3+\beta_2^2\psi_1+\psi_2}{(\beta_1\beta_2\beta_3-1)^2} & \frac{\beta_2^2\beta_3\psi_1+\beta_1\beta_2\psi_3+\beta_3\psi_2}{(\beta_1\beta_2\beta_3-1)^2} \\ \frac{\beta_1\beta_3^2\psi_2+\beta_2\beta_3\psi_1+\beta_1\psi_3}{(\beta_1\beta_2\beta_3-1)^2} & \frac{\beta_2^2\beta_3\psi_1+\beta_1\beta_2\psi_3+\beta_3\psi_2}{(\beta_1\beta_2\beta_3-1)^2} & \frac{\beta_2^2\beta_3^2\psi_1+\beta_3^2\psi_2+\psi_3}{(\beta_1\beta_2\beta_3-1)^2} \end{pmatrix}$$

Ouch. Verify that the denominators cannot be zero (Theorem 1.1).

```
# Check the determinant of (I-beta)
det(IdentityMatrix(3)-BETA)
```

evaluate

$$-\beta_1\beta_2\beta_3 + 1$$

The surface $\beta_1\beta_2\beta_3 = 1$ defines a hole in the parameter space. Here are the covariance structure equations.

```
# Covariance structure equations
eqns = SetupEqns(Sigma)
for item in eqns: show(item)
```

evaluate

$$\frac{\beta_1^2\beta_3^2\psi_2+\beta_1^2\psi_3+\psi_1}{(\beta_1\beta_2\beta_3-1)^2} = \sigma_{11}$$

$$\frac{\beta_1^2\beta_2\psi_3+\beta_1\beta_3\psi_2+\beta_2\psi_1}{(\beta_1\beta_2\beta_3-1)^2} = \sigma_{12}$$

$$\frac{\beta_1\beta_3^2\psi_2+\beta_2\beta_3\psi_1+\beta_1\psi_3}{(\beta_1\beta_2\beta_3-1)^2} = \sigma_{13}$$

$$\frac{\beta_1^2\beta_2^2\psi_3+\beta_2^2\psi_1+\psi_2}{(\beta_1\beta_2\beta_3-1)^2} = \sigma_{22}$$

$$\frac{\beta_2^2\beta_3\psi_1+\beta_1\beta_2\psi_3+\beta_3\psi_2}{(\beta_1\beta_2\beta_3-1)^2} = \sigma_{23}$$

$$\frac{\beta_2^2\beta_3^2\psi_1+\beta_3^2\psi_2+\psi_3}{(\beta_1\beta_2\beta_3-1)^2} = \sigma_{33}$$

If you think those equations look hard to solve, you are right. Evn though it's a system of only six equations in six unknowns, `solve` can't manage it, even when the equations are converted to polynomials. The problem eventually yields to the Gröbner basis methods

described in Chapter **??**. The details will be deferred until then. For the present, it will just be noted that the parameters are not identifiable. This is established below by a numerical example of two distinct parameter vectors that yield the same $\boldsymbol{\Sigma}$. See how naturally one can treat a matrix as what it is: a function of the parameters.

```
# Two numerical parameter vectors produce the same Sigma.
show(Sigma(beta1=1,beta2=2,beta3=3,psi1=1,psi2=1,psi3=1))
show(Sigma(beta1=11/46, beta2=9/22, beta3=46/27, psi1=11/46, psi2=9/44, psi3=46/81))
```

evaluate

$$\begin{pmatrix} \frac{11}{25} & \frac{7}{25} & \frac{16}{25} \\ \frac{7}{25} & \frac{9}{25} & \frac{17}{25} \\ \frac{16}{25} & \frac{17}{25} & \frac{46}{25} \end{pmatrix}$$

$$\begin{pmatrix} \frac{11}{25} & \frac{7}{25} & \frac{16}{25} \\ \frac{7}{25} & \frac{9}{25} & \frac{17}{25} \\ \frac{16}{25} & \frac{17}{25} & \frac{46}{25} \end{pmatrix}$$

Obviously, the second set of numerical values would be exceedingly difficult to guess. It turns out that for every set of parameter values, there is exactly one other set that produces the same $\boldsymbol{\Sigma}$. This will be shown in Chapter 8.

### 4.3.3 Pinwheel Models

You may be familiar with pinwheels, or at least perhaps you used to be. It's like a little toy windmill on a stick. The child blows on the wheel (or runs), and the passage of air turns the wheel. Figure 4.5 shows an example with three nodes[10]

I believe that Min Lim [43] was the first to investigate the identifiability of pinwheel models, in her 2010 Ph.D. thesis. Like most cyclic models, the pinwheel models are not particularly easy to deal with. For models with three or more blades, the Gröbner basis methods of Chapter 8 are needed. As far as I know, Lim was also the first to apply Gröbner basis technology to covariance structure equations. Others have followed and gotten credit for it. Her work remains unpublished.

The two-node model of Figure 4.6 provides an example that does not require more advanced methods. Surprisingly, the parameters are identifiable, a property that this model shares with all the pinwheel models. Maybe $y_1$ could be supply, $y_2$ could be demand, and $x$ could be the cost of raw materials.

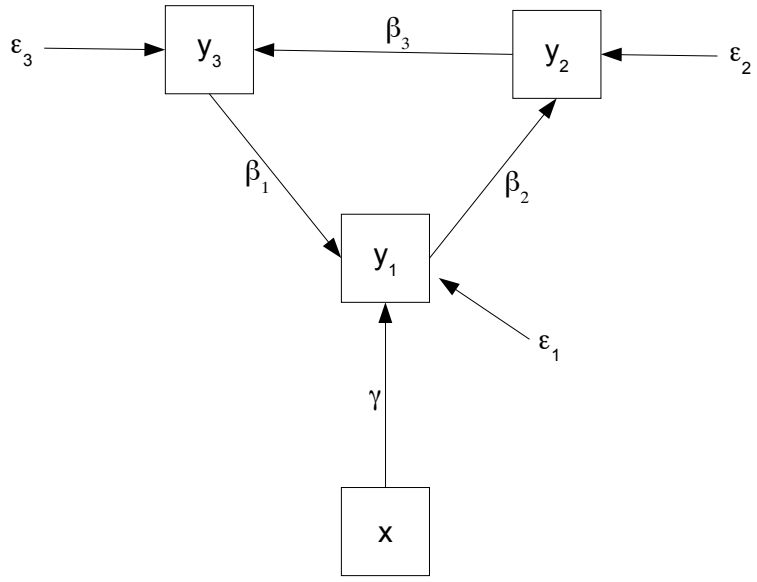
The model equations are

$$\begin{array}{ccccccccc} \mathbf{y}_i & = & \boldsymbol{\beta} & \mathbf{y}_i & + & \boldsymbol{\Gamma} & \mathbf{x}_i & + & \boldsymbol{\epsilon}_i \end{array}$$

$$\begin{pmatrix} y_{i,1} \\ y_{i,2} \end{pmatrix} = \begin{pmatrix} 0 & \beta_1 \\ \beta_2 & 0 \end{pmatrix} \begin{pmatrix} y_{i,1} \\ y_{i,2} \end{pmatrix} + \begin{pmatrix} \gamma \\ 0 \end{pmatrix} (x_i) + \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \end{pmatrix}.$$

[10] I was tempted to call them "blades," like the blades on a propellor.

Figure 4.5: Three-node Pinwheel Model




The first part of the Sage work is routine, and will be presented without comment.

```
# Two-node pinwheel model
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
```

evaluate

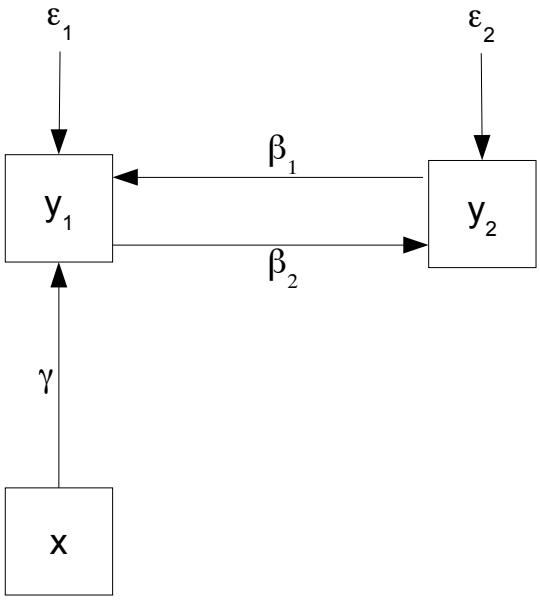
```
PHI = ZeroMatrix(1,1); PHI[0,0] = var('phi'); show(PHI)
GAMMA = ZeroMatrix(2,1)
GAMMA[0,0] = var('gamma'); show(GAMMA)
BETA = ZeroMatrix(2,2)
BETA[0,1] = var('beta1'); BETA[1,0] = var('beta2'); show(BETA)
# The default symbol for DiagonalMatrix is psi
PSI = DiagonalMatrix(2); show(PSI)
```

evaluate

$$\begin{pmatrix} \phi \end{pmatrix}$$

$$\begin{pmatrix} \gamma \\ 0 \end{pmatrix}$$

Figure 4.6: Two-node Pinwheel Model




$$\begin{pmatrix} 0 & \beta_1 \\ \beta_2 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix}$$

```
# Calculate the covariance matrix.
Sigma = PathCov(Phi=PHI,Beta=BETA,Gamma=GAMMA,Psi=PSI)
show(Sigma)
```

evaluate

$$\begin{pmatrix} \phi & -\frac{\gamma\phi}{\beta_1\beta_2-1} & -\frac{\beta_2\gamma\phi}{\beta_1\beta_2-1} \\ -\frac{\gamma\phi}{\beta_1\beta_2-1} & \frac{\gamma^2\phi+\beta_1^2\psi_2+\psi_1}{(\beta_1\beta_2-1)^2} & \frac{\beta_2\gamma^2\phi+\beta_2\psi_1+\beta_1\psi_2}{(\beta_1\beta_2-1)^2} \\ -\frac{\beta_2\gamma\phi}{\beta_1\beta_2-1} & \frac{\beta_2\gamma^2\phi+\beta_2\psi_1+\beta_1\psi_2}{(\beta_1\beta_2-1)^2} & \frac{\beta_2^2\gamma^2\phi+\beta_2^2\psi_1+\psi_2}{(\beta_1\beta_2-1)^2} \end{pmatrix}$$

```
# Set up covariance structure equations.
param = [phi, beta1, beta2, gamma, psi1, psi2] # List of model parameters
eqns = SetupEqns(Sigma)
for item in eqns: show(item)
```

evaluate

$\phi = \sigma_{11}$

$-\frac{\gamma\phi}{\beta_1\beta_2-1} = \sigma_{12}$

$-\frac{\beta_2\gamma\phi}{\beta_1\beta_2-1} = \sigma_{13}$

$\frac{\gamma^2\phi+\beta_1^2\psi_2+\psi_1}{(\beta_1\beta_2-1)^2} = \sigma_{22}$

$\frac{\beta_2\gamma^2\phi+\beta_2\psi_1+\beta_1\psi_2}{(\beta_1\beta_2-1)^2} = \sigma_{23}$

$\frac{\beta_2^2\gamma^2\phi+\beta_2^2\psi_1+\psi_2}{(\beta_1\beta_2-1)^2} = \sigma_{33}$

```
# Try to solve the equations
solut = solve(eqns,param); len(solut)
```

evaluate

4

Naturally, I looked at the four solutions, but that will not be shown. It's more efficient to obtain the solutions as dictionaries, and check the product $\beta_1\beta_2$ for each one. If $\beta_1\beta_2 = 1$, then $|\mathbf{I} - \boldsymbol{\beta}| = 0$. In this case, the solution is outside the parameter space and can be discarded.

```
# Get solutions as dictionaries and check whether beta1*beta2=1
solud = solve(eqns,param,solution_dict=True); len(solud)
for item in solud: (beta1*beta2-1)(item)
```

evaluate

0

0

$\frac{(\sigma_{13}\sigma_{22}-\sigma_{12}\sigma_{23})\sigma_{13}}{(\sigma_{13}\sigma_{23}-\sigma_{12}\sigma_{33})\sigma_{12}} - 1$

0

So it appears that only the third solution is in the parameter space and potentially valid. In the numbering system that starts with zero, that's `solud[2]`. I will be convenient to work with a copy of it; the copy is called `sol`, for no particular reason.

```
# There appears to be just one solution. Take a look.
sol = solud[2]
for item in param: show(item == factor(sol[item]))
```

evaluate

$$\phi = \sigma_{11}$$

$$\beta_1 = \frac{\sigma_{13}\sigma_{22} - \sigma_{12}\sigma_{23}}{\sigma_{13}\sigma_{23} - \sigma_{12}\sigma_{33}}$$

$$\beta_2 = \frac{\sigma_{13}}{\sigma_{12}}$$

$$\gamma = -\frac{\sigma_{13}^2\sigma_{22} - 2\,\sigma_{12}\sigma_{13}\sigma_{23} + \sigma_{12}^2\sigma_{33}}{(\sigma_{13}\sigma_{23} - \sigma_{12}\sigma_{33})\sigma_{11}}$$

$$\psi_1 = -\frac{\left(\sigma_{13}^2\sigma_{22} - 2\,\sigma_{12}\sigma_{13}\sigma_{23} + \sigma_{11}\sigma_{23}^2 + \sigma_{12}^2\sigma_{33} - \sigma_{11}\sigma_{22}\sigma_{33}\right)\left(\sigma_{13}^2\sigma_{22} - 2\,\sigma_{12}\sigma_{13}\sigma_{23} + \sigma_{12}^2\sigma_{33}\right)}{(\sigma_{13}\sigma_{23} - \sigma_{12}\sigma_{33})^2\sigma_{11}}$$

$$\psi_2 = \frac{\sigma_{13}^2\sigma_{22} - 2\,\sigma_{12}\sigma_{13}\sigma_{23} + \sigma_{12}^2\sigma_{33}}{\sigma_{12}^2}$$

This looks good. Just to be sure, it's helpful to substitute for the $\sigma ij$ in terms of the model parameters, and verify that each result equals the single parameter of interest. On very rare occasions, `solve` gives results that don't check out, and it's quite easy to check. In the dictionary `theta`, the keys are the $\sigma ij$, and the entries are the variances and covariances written as a function of the model parameters.

```
# Check the solutions by evaluating them at the model parameters.
theta = SigmaOfTheta(Sigma)
for item in param: item == factor( sol[item](theta) )
```

evaluate

$$\phi = \phi$$

$$\beta_1 = \beta_1$$

$$\beta_2 = \beta_2$$

$$\gamma = \gamma$$

$$\psi_1 = \psi_1$$

$$\psi_2 = \psi_2$$

This is what success looks like. All the parameters are identified, except possibly on a set of volume zero in the parameter space. In order to determine where the parameters might not be identifiable, it's necessary to evaluate the denominators in terms of the model parameters, and also to check the numerators of the variance parameters $\psi_1$ and $\psi_2$. The denominator of the solution for $\beta_1$ also appears in the denominators of $\gamma$ and $\psi_1$.

```
# Check denominators
d1 = denominator(sol[beta1]); show(d1)
factor(d1(theta))
```

evaluate

$\sigma_{13}\sigma_{23} - \sigma_{12}\sigma_{33}$

$-\frac{\gamma\phi\psi_2}{(\beta_1\beta_2-1)^2}$

So all is well provided $\gamma \neq 0$. This is one of those "obvious" things that one appreciates after the fact. Looking at the path diagram in Figure 4.6, it's "obvious" that if the arrow from $x$ to $y_1$ is eliminated, then $y_1$, $y_2$, $\epsilon_1$ and $\epsilon_2$ form a closed system with three covariances and four unknown parameters. By the parameter count rule, identifiability is ruled out almost everywhere[11] in the parameter space.

The quantity $\sigma_{12}$ appears in the denominators of $\beta_2$ and $\psi_2$. A glance at the covariance matrix shows $\sigma_{12} = -\frac{\gamma\phi}{\beta_1\beta_2-1}$, so it too will be non-zero provided $\gamma \neq 0$.

The last job is to check the numerators of $\psi_1$ and $\psi_2$; identifiability fails for any set of parameter values where either of these variances is equal to zero.

```
# Numerator of psi1: Where is it non-zero?
a = factor(numerator(sol[psi1])); show(a)
b = factor(a(theta)); show(b)
```

evaluate

$-(\sigma_{13}^2\sigma_{22} - 2\,\sigma_{12}\sigma_{13}\sigma_{23} + \sigma_{11}\sigma_{23}^2 + \sigma_{12}^2\sigma_{33} - \sigma_{11}\sigma_{22}\sigma_{33})(\sigma_{13}^2\sigma_{22} - 2\,\sigma_{12}\sigma_{13}\sigma_{23} + \sigma_{12}^2\sigma_{33})$

$\frac{\gamma^2\phi^3\psi_1\psi_2^2}{(\beta_1\beta_2-1)^4}$

Again, we are good provided $\gamma \neq 0$. The variance $\psi_2$ is also okay, because its numerator is the second factor in the numerator of $psi_1$. The final conclusion is simple and clean. The parameters of the 2-node pinwheel model are identifiable everywhere in the parameter space except where $\gamma = 0$.

**Identifiability of cyclic models** The examples in this section tell the essential story. The parameters of cyclic models might be identifiable, or they might not be. There are no useful general rules, except for the parameter count rule, which applies to everything. Each model needs to be investigated on a case-by-case basis. In general, the covariance structure equations are hairy. A computer algebra system like Sage is practically a necessity.

One limiting feature of the examples in this section is that they all feature the same number of covariance structure equations and unknown parameters. This is necessary for Sage's `solve` function to return the solutions we need. When there are more equations than unknowns (the typical case in structural equation modeling), there are really three ways to proceed. One way is to try to solve the equations by hand. Good luck with that, and even if it is within your powers, its too much to expect of most users on a routine basis. The second option is to set some of the covariance structure equations aside and give `solve` the same number of equations as unknowns. Especially for cyclic models where the expressions are messy, this can require an unreasonable amount of mathematical insight,

[11] This is measure theory talk. Here, it means except on a set of volume zero, which is almost everywhere with respect to Lebesgue measure.

or a lot of trial and error. The third alternative is to try the Gröbner basis methods of Chapter 8. This has a lot of promise, though it does not always work. Examples will be given in Chapter 8.

## 4.4 Direction of Causality

### 4.4.1 Deciding based on data

Almost always, the issue of whether $a \to b$ or $b \to a$ is a modeling decision. That is, the person putting the model together writes it down or draws a picture, and that's it. Frequently, it's not controversial, and nobody would argue. For example, a child's academic performance might be influenced by the parent's income, but influence going the other way is a lot more difficult to believe. In cases that are less clear, one can estimate the model parameters based on a data set, and if the model fits, one can at least assert that the data are consistent with a model in which, say, exercise tends to reduce arthritis pain.

The question arises, though, is this the best we can do? Is it possible to determine direction of causality empirically, through analysis of data? The answer is usually no, but sometimes yes.

**When the answer is no** Consider a model that obeys the conditions of the acyclic rule. In my experience, this includes most path models used in practice. Suppose further that the model includes all permissible straight arrows and covariances between error terms. Then as noted at the end of the proof of the acyclic rule, the parameters are just identifiable everywhere in the parameter space. For a saturated model like this, the parameters are one-to-one with the variances and covariances of the observable variables. In a sample data set, the same connection holds for the parameter estimates and the estimated variances and covariances, whether the estimation is method-of-moments or maximum likelihood under a normal model. If the objective of modeling is to fit the sample variances and covariances as well as possible[12], this is the best one can do. The fit is perfect.

Now observe that there are lots of different ways to group the variables and order the groups, including models with the causality flowing in the opposite direction as any model one might propose. All the models are saturated, and they all fit the data equally well, which is as well as possible. No conceivable data set can cast light on which way the arrows should go.

[12]In the normal case, the vector of sample variances and covariances is a jointly minimal sufficient statistic for the model parameters, and so is any one-to-one transform of them. This means that conditionally on the vector of MLEs, the distribution of the data is free of all model parameters. In other words, there's no more to learn. For non-normal models this might not be quite true, but one would have to specify the non-normal distribution(s) to make any progress.

**When the answer is yes** The conclusion above applies to unrestricted acyclic models. With restrictions on the values of the model parameters (presumably well justified by theory), it may be possible to decide on direction of causality based on a formal hypothesis test. Just as a proof of concept, consider a minimal, artificial example. There are two observable variables, $x$ and $y$. Under model one,

$$y = x + \epsilon,$$

with $Var(\epsilon) = \psi > 0$, and $\epsilon$ independent of $x$. Under model two,

$$x = y + \delta,$$

with $Var(\delta) = \psi > 0$, and $\delta$ independent of $y$.

So either $x$ is influencing $y$, or $y$ is influencing $x$. If model one holds, $Var(y) = Var(x) + \psi$, so that $Var(y) > Var(x)$. Under model two, the opposite conclusion holds. So direction of influence can be decided by testing the difference between variances. This works because the regression coefficient linking $x$ and $y$ is restricted to equal one for both models. If the regression coefficients were allowed to be different and unrestricted, both model would fit perfectly, and testing for direction of causality would be impossible.

For an example that is closer to being realistic, consider Duncan's cyclic model, described in Section 4.3.1 (see Figure 4.3 on page 381). This model is saturated, but it's not acyclic. The question of whether $y_1$ influences $y_2$, or the other way around (or both, or neither) can be resolved by testing $H_0 : \beta_1 = 0$ and $H_0 : \beta_2 = 0$. The same strategy would work for the pinwheel model of Figure 4.6.

### 4.4.2 One more acyclic example

**Example 4.4.1** *Direction of influence in an acyclic model*

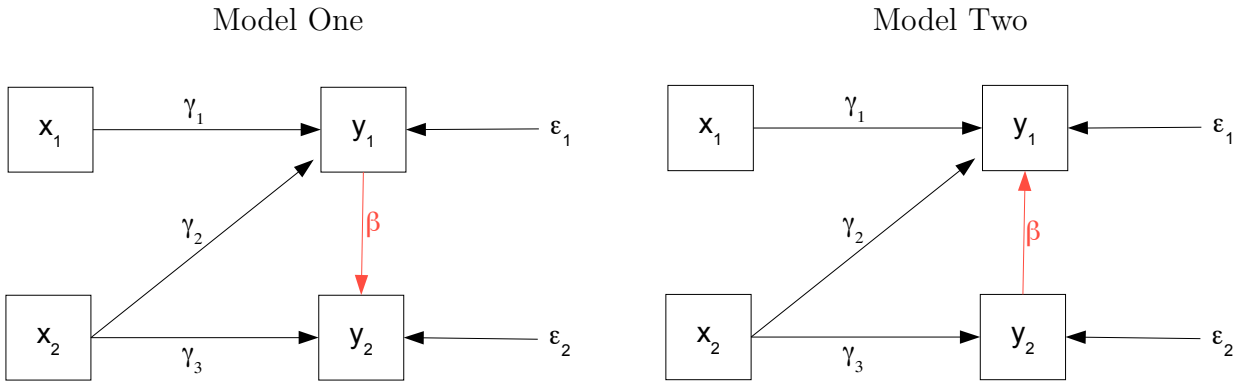
Consider the two models in Figure 4.7. These models differ only in the direction of influence between $y_1$ and $y_2$. Upon reflection, deciding between these models based on data would seem to be a real possibility. In Model Two, there is no one-way path connecting $x_1$ and $y_2$, so the covariance between the two variables should be zero[13]. This does not hold for Model One. There's more to it than that, though. It's time to get systematic.

Recall that when the parameters of a model are identifiable but the model is not saturated, the model implies equality constraints on the variances and covariances, with the number of equality constraints equal to the number of variances and covariances minus the number of parameters. A good way to judge the fit of a model to a data set is to assess how closely the *sample* variances and covariances obey these equality constraints — that is, the constraints that must hold for the true variances and covariances if the model is correct[14]. If two models imply different constraints, it is possible to decide whether one fits significantly better. Now we will find the equality constraints implied by the two models of Figure 4.7.

[13]This intuition will be formaized in Wright's multiplication theorem. See Section **??**

[14]See Chapter 1, Section 170. Chapter 7 treats the testing of model fit in further detail.

Figure 4.7: Two possible directions of influence




For both models, the parameter vector is $\boldsymbol{\theta} = (\phi_1, \phi_2, \gamma_1, \gamma_2, \gamma_3, \beta, \psi_1, \psi_2)$. That's eight parameters. The parameters are identifiable in both models by the acyclic rule, and there are $(4+1)4/2 = 10$ variances and covariances. This means that for both models, there are two equality constraints.

It's quite easy to do most of the calculations by hand for these simple models, but Sage will come in handy later in the process. So, we'll use Sage for the whole thing as much as possible. After the usual basic setup, the first job is to calculate both covariance matrices. Only the $\boldsymbol{\beta}$ matrix is different for the two models.

```
# Direction of causality in an acyclic model
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
```

evaluate

```
PHI = DiagonalMatrix(2,'phi'); show(PHI)
GAMMA = ZeroMatrix(2,2)
GAMMA[0,0] = var('gamma1'); GAMMA[0,1] = var('gamma2')
GAMMA[1,1] = var('gamma3'); show(GAMMA)
BETA1 = ZeroMatrix(2,2) # y1 -> y2
BETA1[1,0] = var('beta'); show(BETA1)
BETA2 = ZeroMatrix(2,2) # y2 -> y1
BETA2[0,1] = var('beta'); show(BETA2)
# The default symbol for DiagonalMatrix is psi
PSI = DiagonalMatrix(2); show(PSI)
```

evaluate

$$\begin{pmatrix} \phi_1 & 0 \\ 0 & \phi_2 \end{pmatrix}$$

$$\begin{pmatrix} \gamma_1 & \gamma_2 \\ 0 & \gamma_3 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & \beta \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix}$$

Calculating the two covariance matrices for comparison,

```
# Calculate Sigma1, with y1 -> y2
Sigma1 = PathCov(Phi=PHI,Beta=BETA1,Gamma=GAMMA,Psi=PSI)
show(Sigma1)
```

evaluate

$$\begin{pmatrix} \phi_1 & 0 & \gamma_1\phi_1 & \beta\gamma_1\phi_1 \\ 0 & \phi_2 & \gamma_2\phi_2 & (\beta\gamma_2+\gamma_3)\phi_2 \\ \gamma_1\phi_1 & \gamma_2\phi_2 & \gamma_1^2\phi_1+\gamma_2^2\phi_2+\psi_1 & \beta\gamma_1^2\phi_1+\beta\gamma_2^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_1 \\ \beta\gamma_1\phi_1 & (\beta\gamma_2+\gamma_3)\phi_2 & \beta\gamma_1^2\phi_1+\beta\gamma_2^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_1 & \beta^2\gamma_1^2\phi_1+\beta^2\gamma_2^2\phi_2+2\,\beta\gamma_2\gamma_3\phi_2+\gamma_3^2\phi_2+\beta^2\psi_1+\psi_2 \end{pmatrix}$$

```
# Calculate Sigma2, with y2 -> y1
Sigma2 = PathCov(Phi=PHI,Beta=BETA2,Gamma=GAMMA,Psi=PSI)
show(Sigma2)
```

evaluate

$$\begin{pmatrix} \phi_1 & 0 & \gamma_1\phi_1 & 0 \\ 0 & \phi_2 & (\beta\gamma_3+\gamma_2)\phi_2 & \gamma_3\phi_2 \\ \gamma_1\phi_1 & (\beta\gamma_3+\gamma_2)\phi_2 & \beta^2\gamma_3^2\phi_2+2\,\beta\gamma_2\gamma_3\phi_2+\gamma_1^2\phi_1+\gamma_2^2\phi_2+\beta^2\psi_2+\psi_1 & \beta\gamma_3^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_2 \\ 0 & \gamma_3\phi_2 & \beta\gamma_3^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_2 & \gamma_3^2\phi_2+\psi_2 \end{pmatrix}$$

For Model Two, as expected, $\sigma_{14} = Cov(x_1, y_2) = 0$, a constraint that does not hold for Model One as long as $\beta \neq 0$ and $\gamma_1 \neq 0$. That's one constraint on the covariance matrix for Model Two. The other one is that $\sigma_{12} = Cov(x_1, x_2) = 0$. This is shared by both models, because there is no curved, double-headed arrow joining $x_1$ and $x_2$. Thus, the two constraints on $\mathbf{\Sigma}_2$ are accounted for.

**The remaining constraint** To obtain the remaining constraint for Model One, we will first solve the covariance structure equations for the parameters. One of the equations will not be needed. Substituting the solutions back into that equation will yield the constraint.

The first few covariance structure equations for Model One can be solved by eye and brain (not even by hand). Inspecting the matrix `Sigma1` above, we obtain $\phi_1 = \sigma_{11}$ and $\gamma_1 = \sigma_{13}/\sigma_{11}$. But then, when we attempt to get $\beta$, we notice that $\beta = \sigma_{14}/\sigma_{13}$ only works if $\gamma_1 \neq 0$.

If $\gamma_1 = 0$, what happens should perhaps have been obvious from the beginning[15]. Take another look at Figure 4.7. If the arrow from $x_1$ to $y_1$ is missing from both path diagrams, then the variables $x_2$, $y_1$ and $y_2$ form isolated sub-models. In both cases, the parameters are identifiable by the acyclic rule. Furthermore, they are *just* identifiable. Their parameters are one-to-one with the sub-matrix of variances and covariances. They impose no further constraints on the covariance matrix, and it's impossible to distinguish between them. The conclusion is that *the direction of influence between $y_1$ and $y_2$ cannot be determined if $\gamma_1 = 0$.*

If you look back at the computed covariance matrices, you will see that for both models, $\sigma_{13} = 0$ if and only if $\gamma_1 = 0$. With a real data set, you could start by testing $H_0 : \sigma_{13} = 0$, thus testing the null hypothesis $\gamma_1 = 0$ without making a commitment to either model. If you did not reject that null hypothesis, it would be best to give up on an empirically based decision between $y_1 \to y_2$ and $y_2 \to y_1$. If you did reject the null hypothesis, it would be reasonable to proceed. This is progress.

It's not really necessary, but Sage makes it easy to look at the covariance matrices when $\gamma_1 = 0$. This is a nice way to consider special cases.

```
# How do the covariance matrices look with gamma1=0?
show( Sigma1(gamma1=0) )
show( Sigma2(gamma1=0) )
```

evaluate

$$\begin{pmatrix} \phi_1 & 0 & 0 & 0 \\ 0 & \phi_2 & \gamma_2\phi_2 & (\beta\gamma_2+\gamma_3)\phi_2 \\ 0 & \gamma_2\phi_2 & \gamma_2^2\phi_2+\psi_1 & \beta\gamma_2^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_1 \\ 0 & (\beta\gamma_2+\gamma_3)\phi_2 & \beta\gamma_2^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_1 & \beta^2\gamma_2^2\phi_2+2\,\beta\gamma_2\gamma_3\phi_2+\gamma_3^2\phi_2+\beta^2\psi_1+\psi_2 \end{pmatrix}$$

$$\begin{pmatrix} \phi_1 & 0 & 0 & 0 \\ 0 & \phi_2 & (\beta\gamma_3+\gamma_2)\phi_2 & \gamma_3\phi_2 \\ 0 & (\beta\gamma_3+\gamma_2)\phi_2 & \beta^2\gamma_3^2\phi_2+2\,\beta\gamma_2\gamma_3\phi_2+\gamma_2^2\phi_2+\beta^2\psi_2+\psi_1 & \beta\gamma_3^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_2 \\ 0 & \gamma_3\phi_2 & \beta\gamma_3^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_2 & \gamma_3^2\phi_2+\psi_2 \end{pmatrix}$$

[15] You can always pretend it was obvious. This is a great way to impress your friends. But are those people really your friends? Maybe you should re-think your agreement to help them cheat on the final exam.

If you examine the covariance matrices carefully, you will see that even when $\sigma_{14} = 0$, there is another way to get at $\beta$. This had to be the case, since the parameters of both models are identifiable everywhere in the parameter space. For both models, when $\gamma_1 = 0$, the solution for $\beta$ emerges as part of the solution of two linear equations in two unknowns. Given the way that the acyclic rule was proved, the equations had to be linear.

Let us continue, assuming $\gamma_1 \neq 0$. The next step is to solve eight covariance structure equations in eight unknowns. The `solve` function wants the same number of equations as unknowns and there are nine equations, but the equations are simple enough so that it will be clear which one should be set aside.

```
# Set up covariance structure equations for the first model
param = [phi1, phi2,gamma1,gamma2,gamma3,beta,psi1,psi2]
eqns = SetupEqns(Sigma1)
for item in eqns: show(item)
```

evaluate

$\phi_1 = \sigma_{11}$

$0 = \sigma_{12}$

$\gamma_1\phi_1 = \sigma_{13}$

$\beta\gamma_1\phi_1 = \sigma_{14}$

$\phi_2 = \sigma_{22}$

$\gamma_2\phi_2 = \sigma_{23}$

$(\beta\gamma_2 + \gamma_3)\phi_2 = \sigma_{24}$

$\gamma_1^2\phi_1 + \gamma_2^2\phi_2 + \psi_1 = \sigma_{33}$

$\beta\gamma_1^2\phi_1 + \beta\gamma_2^2\phi_2 + \gamma_2\gamma_3\phi_2 + \beta\psi_1 = \sigma_{34}$

$\beta^2\gamma_1^2\phi_1 + \beta^2\gamma_2^2\phi_2 + 2\,\beta\gamma_2\gamma_3\phi_2 + \gamma_3^2\phi_2 + \beta^2\psi_1 + \psi_2 = \sigma_{44}$

Item 1 (starting from zero) should be deleted because it's useless, and item 8 is a good choice to set aside because it's messy and not needed to solve. Once solutions have been obtained, the plan is to substitute them back into the unused equation, yielding the model-induced constraint on the $\sigma_{ij}$.

```
# Delete items 1 (starting from zero) and 8. Work with a
# copy of the list of equations. Trying Python syntax,
eq = list(eqns) # Without list, eq is just another name for eqns
del eq[8]; del eq[1]
for item in eq: show(item)
```

evaluate

$\phi_1 = \sigma_{11}$

$\gamma_1\phi_1 = \sigma_{13}$

$\beta\gamma_1\phi_1 = \sigma_{14}$

$\phi_2 = \sigma_{22}$

$\gamma_2\phi_2 = \sigma_{23}$

$(\beta\gamma_2 + \gamma_3)\phi_2 = \sigma_{24}$

$\gamma_1^2\phi_1 + \gamma_2^2\phi_2 + \psi_1 = \sigma_{33}$

$\beta^2\gamma_1^2\phi_1 + \beta^2\gamma_2^2\phi_2 + 2\,\beta\gamma_2\gamma_3\phi_2 + \gamma_3^2\phi_2 + \beta^2\psi_1 + \psi_2 = \sigma_{44}$

```
# Solve equations, obtaining the solution as a dictionary
solud = solve(eq,param,solution_dict=True); len(solud)
```

evaluate

1

There is one solution. Put it in an object named `sol`, which is a more convenient name than `solud[0]`. Keep in mind that `sol` is a dictionary.

```
# Take a look
sol = solud[0]
for item in param: show(item == sol[item])
```

evaluate

$\phi_1 = \sigma_{11}$

$\phi_2 = \sigma_{22}$

$\gamma_1 = \frac{\sigma_{13}}{\sigma_{11}}$

$\gamma_2 = \frac{\sigma_{23}}{\sigma_{22}}$

$\gamma_3 = -\frac{\sigma_{14}\sigma_{23} - \sigma_{13}\sigma_{24}}{\sigma_{13}\sigma_{22}}$

$\beta = \frac{\sigma_{14}}{\sigma_{13}}$

$\psi_1 = -\frac{\sigma_{13}^2\sigma_{22} + \left(\sigma_{23}^2 - \sigma_{22}\sigma_{33}\right)\sigma_{11}}{\sigma_{11}\sigma_{22}}$

$\psi_2 = -\frac{\left(\sigma_{24}^2 - \sigma_{22}\sigma_{44}\right)\sigma_{13}^2 - \left(\sigma_{23}^2 - \sigma_{22}\sigma_{33}\right)\sigma_{14}^2}{\sigma_{13}^2\sigma_{22}}$

The only covariance that appears in any denominator is $\sigma_{13}$, which is strictly positive provided $\gamma_1 \neq 0$. This solution is good.

Now we can obtain the remaining constraint on the $\sigma_{ij}$ values. The process is to take

the unused covariance structure equation

$$\beta\gamma_1^2\phi_1 + \beta\gamma_2^2\phi_2 + \gamma_2\gamma_3\phi_2 + \beta\psi_1 = \sigma_{34}$$

and for each model parameter, substitute the solution in `sol`. Then simplify. The result will be a relation among the $\sigma_{ij}$ that has to hold if the model is correct. By hand, this would not be a pleasant task, but Sage makes it easy. Incidentally, this is why the original complete set of covariance structure equations was preserved in `eqns` — so we could use the deleted equation 8.

```
# Obtain the constraint
constraint = factor( eqns[8](sol) )
show(constraint)
```

evaluate

$$-\frac{\sigma_{14}\sigma_{23}^2 - \sigma_{13}\sigma_{23}\sigma_{24} - \sigma_{14}\sigma_{22}\sigma_{33}}{\sigma_{13}\sigma_{22}} = \sigma_{34}$$

A lot of simplification was achieved by `factor`. Because of $\sigma_{13}$ in the denominator, the equality above applies only provided $\gamma_1 \neq 0$. A bit more generality can be obtained by multiplying both sides by the denominator. It also looks nicer without all the minus signs.

```
# Clear denominator, simplify
constraint  = constraint*(sigma13*sigma22) + sigma14*sigma23^2; constraint
```

evaluate

$$\sigma_{13}\sigma_{23}\sigma_{24} + \sigma_{14}\sigma_{22}\sigma_{33} = \sigma_{14}\sigma_{23}^2 + \sigma_{13}\sigma_{22}\sigma_{34}$$

The first observation is that this constraint is a real bear. To me, it's remarkable that for two models that seem so similar, one of them imposes only the simplest type of constraint (something equals zero), while the other imposes one zero, and one constraint that is exceedingly complicated. The constraint shown above involves seven variances and covariances, if I have not mis-counted.

Here's another, more general point. In cases like this, where a constraint involves a fraction that only makes sense when the denominator is non-zero, formally multiplying both sides of the equation by the denominator usually (always?) yields an equality that is true for *all* parameter values. Let's see if it works here.

To check, it helps to move everything to one side, yielding a polynomial in the $\sigma_{ij}$ that is equal to zero. A bit strangely (to me), this can be accomplished in Sage by factoring an equality like `constraint`. Nothing is successfully factored in this case, but it does get rid of the equals sign. The result will be called `constraintp`, the constraint in polynomial form.

```
# Obtain the constraint as a polynomial = to zero
constraintp = factor(constraint); constraintp
```

evaluate

$$-\sigma_{14}\sigma_{23}^2 + \sigma_{13}\sigma_{23}\sigma_{24} + \sigma_{14}\sigma_{22}\sigma_{33} - \sigma_{13}\sigma_{22}\sigma_{34}$$

If you recall, the `SigmaOfTheta` function in the `sem` package produces a dictionary that can be used to evaluate a function of the $\sigma_{ij}$ values in terms of the model parameters.

```
# Check: Is the constraint always true for Model One?
theta1 = SigmaOfTheta(Sigma1)
constraintp(theta1)
```

evaluate

$$-\beta\gamma_1\gamma_2^2\phi_1\phi_2^2+(\beta\gamma_2+\gamma_3)\gamma_1\gamma_2\phi_1\phi_2^2+\left(\gamma_1^2\phi_1+\gamma_2^2\phi_2+\psi_1\right)\beta\gamma_1\phi_1\phi_2-\left(\beta\gamma_1^2\phi_1+\beta\gamma_2^2\phi_2+\gamma_2\gamma_3\phi_2+\beta\psi_1\right)\gamma_1\phi_1\phi_2$$

Well, that's a mess. Multiply it out and hope for cancellation.

```
expand( constraintp(theta1) )
```

evaluate

$$0$$

That is satisfying. The constraint is true of Model One everywhere in the parameter space, not just where $\gamma_1 \neq 0$.

The constraint holds for Model One. It should *not* hold in general of Model Two, but is it true under some circumstances — that is, does the constraint hold anywhere in the parameter space under Model Two? It is gratifyingly easy to get the answer.

```
# Evaluate the constraint (in polynomial form) under Model Two.
theta2 = SigmaOfTheta(Sigma2)
factor( constraintp(theta2) )
```

evaluate

$$-\beta\gamma_1\phi_1\phi_2\psi_2$$

This is excellent. The constraint only holds under Model Two if $\gamma_1 = 0$ or $\beta = 0$ (or both). It has already been established that determining direction of influence is impossible when $\gamma_1 = 0$. If $\beta = 0$ the whole enterprise does not make sense, because in that case there is no causal connection between $y_1$ and $y_2$, in either direction.

The conclusion is that for this model, it is possible to make an empirically based decision on the direction of influence between $y_1$ and $y_2$, provided that $\gamma_1 \neq 0$ and $\beta \neq 0$. This is because the two different directions of influence imply different constraints on the covariance matrix of the observable variables. To summarize,

- Both models imply $\sigma_{12} = 0$.
- Model One ($y_1 \rightarrow y_2$) implies $\sigma_{13}\sigma_{23}\sigma_{24} + \sigma_{14}\sigma_{22}\sigma_{33} - \sigma_{14}\sigma_{23}^2 - \sigma_{13}\sigma_{22}\sigma_{34} = 0$. Model Two does not, provided $\gamma_1 \neq 0$ and $\beta \neq 0$.
- Model Two ($y_2 \rightarrow y_1$) implies $\sigma_{14} = 0$. Model One does not, provided $\gamma_1 \neq 0$ and $\beta \neq 0$.

**Data analysis strategy** To me, the following procedure makes sense.

1. First, test $H_0 : \sigma_{12} = 0$. It's easy to do, and if the null hypothesis is rejected, both models are thrown into question.
2. Next, test $H_0 : \sigma_{12} = 0$, true if and only if $\gamma_1 = 0$, for both models. If the null hypothesis is rejected, proceed.
3. Testing $H_0 : \beta = 0$ is tricky without actually fitting a structural equation model. Leave it for later.
4. Test $H_0 : \sigma_{14} = 0$. If it is rejected, Model Two is thrown into question, and Model One is supported.
5. Test $H_0 : \sigma_{13}\sigma_{23}\sigma_{24} + \sigma_{14}\sigma_{22}\sigma_{33} - \sigma_{14}\sigma_{23}^2 - \sigma_{13}\sigma_{22}\sigma_{34} = 0$. If it is rejected, Model One is thrown into question, and Model Two is supported.
6. Hope that the results of the last two tests support the same conclusion.

It's probably not so obvious how to test big messy hypotheses like the one in point 5. Mostly for that reason, an example with simulated data will be given, and the whole strategy will be illustrated using `lavaan`. After that, I will present another approach that involves comparing fit statistics for the two models.

### 4.4.3 The acyclic example with simulated data

Using R, a data set will be simulated from Model One, in which $y_1 \rightarrow y_2$. To make it a bit more interesting, the data will be non-normal. The true values of $\gamma_1$ and $\beta$ are nonzero, which is crucial to being able to distinguish between Models One and Two.

```
> # Simulate and analyze data from WhichWay Model One (y1 -> y2)
> rm(list=ls()); options(scipen=999)
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
>
> # Simulate data. Make it skewed.
> n = 150
```

```
> theta=2 # Mean of exponential x1 and x2. Variance is theta^2 = phi1=phi2
> gamma1 = 1; gamma2 = 0.75; gamma3 = 0.5; beta = 0.75
> psi1 = 50; psi2 = 100
> set.seed(9999)
>
> x1 = theta*rexp(n); x2 = theta*rexp(n)
> epsilon1 = sqrt(psi1)*rexp(n) # E(epsilon1) = sqrt(psi1), Var(epsilon1) = psi1
> epsilon2 = sqrt(psi2)*rexp(n) # E(epsilon2) = sqrt(psi2), Var(epsilon2) = psi2
> # Expected values of epsilons are not zero, so equations do have intercepts.
> y1 = gamma1*x1 + gamma2*x2 + epsilon1
> y2 = gamma3*x2 + beta*y1 + epsilon2
> datta = cbind(x1,x2,y1,y2)
> cor(datta)
            x1          x2        y1        y2
x1  1.00000000 -0.02037168 0.3246913 0.2537685
x2 -0.02037168  1.00000000 0.2283975 0.2739164
y1  0.32469126  0.22839750 1.0000000 0.6301020
y2  0.25376855  0.27391641 0.6301020 1.0000000
```

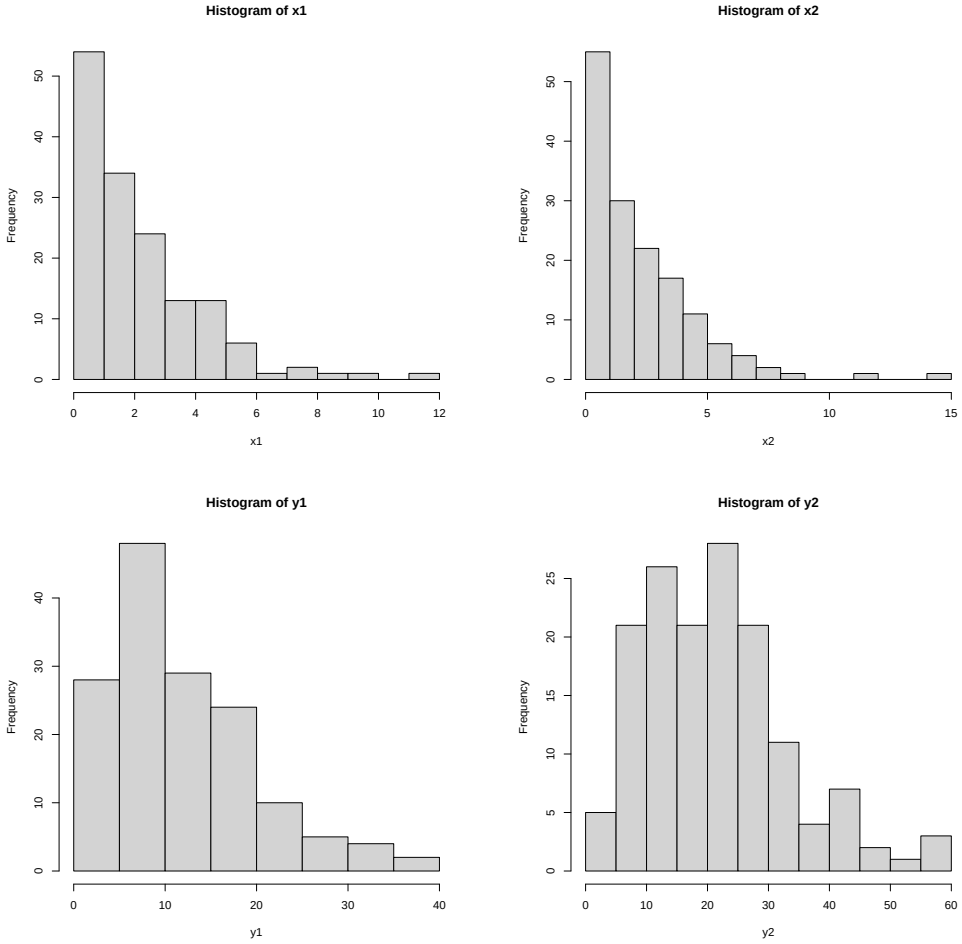
The correlations are modest, and typical of what is obtained in most social science research. I played around with the parameter values to achieve this goal.

As Figure 4.8 shows, the data are definitely not normal. Here is the code that produced the graphics. It can be convenient to write pdf files directly when you are doing a batch of them. For each one, it's necessary to open the file, issue the R command producing the plot, and then close the "device."

```
> # Writes graphics files to the working directory. There are also
> # png() and jpeg() functions.
> pdf(file = 'histx1.pdf'); hist(x1, breaks = 'fd'); dev.off()
> pdf(file = 'histx2.pdf'); hist(x2, breaks = 'fd'); dev.off()
> pdf(file = 'histy1.pdf'); hist(y1, breaks = 'fd'); dev.off()
> pdf(file = 'histy2.pdf'); hist(y2, breaks = 'fd'); dev.off()
```

It's time for lavaan. In lavaan, it is perfectly possible to specify a model with just variances and covariances, and no model equations. The MLEs are just the usual sample variances and covariances, with $n$ in the denominator rather than $n-1$. For each parameter (a variance or covariance), the output of `summary` automatically includes $z$-tests for the null hypothesis that the parameter equals zero. For covariances this makes sense, and it's something we particularly want for $\sigma_{12}$, $\sigma_{13}$ and $\sigma_{14}$. The complicated polynomial that equals zero under Model Two can be specified with the `:=` notation. The following Sage code saves typing in the polynomial. I just copy-pasted the Sage output. It never hurts to have the same parameter names in Sage and lavaan.

Figure 4.8: Histograms of the simulated data




```
# Handy for pasting into lavaan
print(constraintp)
```

evaluate

```
-sigma14*sigma23^2 + sigma13*sigma23*sigma24 + sigma14*sigma22*sigma33 -
sigma13*sigma22*sigma34
```

Now the lavaan session continues.

```
> # Model sigmod has just variances and covariances
> sigmod = 'x1~~sigma11*x1; x1~~sigma12*x2; x1~~sigma13*y1; x1~~sigma14*y2
+                           x2~~sigma22*x2; x2~~sigma23*y1; x2~~sigma24*y2
+                                           y1~~sigma33*y1; y1~~sigma34*y2
```

```
+                                                          y2~~sigma44*y2
+         # The constraint implied by Model Two. This polynomial should = 0
+         con := sigma13*sigma23*sigma24 + sigma14*sigma22*sigma33 -
+                sigma14*sigma23^2 - sigma13*sigma22*sigma34
+       ' # End of model string
> sigfit1 = lavaan(sigmod,datta); summary(sigfit1)
lavaan 0.6-7 ended normally after 74 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         10

  Number of observations                           150

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  x1 ~~
    x2      (sg12)   -0.089    0.358   -0.249    0.803
    y1      (sg13)    5.089    1.346    3.782    0.000
    y2      (sg14)    5.863    1.946    3.013    0.003
  x2 ~~
    y1      (sg23)    3.980    1.459    2.727    0.006
    y2      (sg24)    7.036    2.175    3.236    0.001
  y1 ~~
    y2      (sg34)   57.865    8.863    6.529    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    x1      (sg11)    3.943    0.455    8.660    0.000
    x2      (sg22)    4.874    0.563    8.660    0.000
    y1      (sg33)   62.296    7.193    8.660    0.000
    y2      (sg44)  135.380   15.632    8.660    0.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
```

```
con              394.688  404.124    0.977    0.329
```

Going through the output from top to bottom, first notice that the test statistic for model fit equals zero, as it must. Then, the test of $H_0 : \sigma_{12} = 0$ is comfortably non-significant. This is good, since $Cov(x_1, x_2) = 0$ under both models.

Recalling that $\sigma_{13} = \gamma\phi_1$ under both models, and that $\gamma_1 \neq 0$ is required for the two models to imply different covariance matrices, the $z$ statistic of 3.782 ($p \approx 0$) for $H_0 : \sigma_{13} = 0$ is good news.

Model Two implies $\sigma_{14} = 0$, while under Model One this covariance is *not* zero provided that $\gamma_1 \neq 0$ and $\beta \neq 0$. The test of $\sigma_{14}$ ($z = 3.013$, $p = 0.003$) comfortably supports Model One by the conventional $\alpha = 0.05$ standard.

Model One implies that the polynomial $\sigma_{14}\sigma_{23}^2 - \sigma_{13}\sigma_{23}\sigma_{24} - \sigma_{14}\sigma_{22}\sigma_{33} + \sigma_{13}\sigma_{22}\sigma_{34}$ equals zero, while under Model Two it is zero only if $\gamma_1 = 0$ or $\beta = 0$. The test (based on the multivariate delta method – see Appendix A, page 564) yields $z = 0.977$, $p = 0.329$, which does not indicate a true value different from zero. This also supports Model One.

If you look at the output carefully, you will notice something odd about the $z$-tests for the hypothesis that the variances are zero. It's a null hypothesis that's absurd in this case and so it doesn't really matter, but still it seems fishy that the test statistics are all identical ($z = 8.66$). The reason is that for a normal random variable, the asymptotic variance of the sample variance is $2\sigma^4/n$. This makes the standard error equal to $\widehat{\sigma}^2\sqrt{2/n}$. Dividing the parameter estimate by its standard error to test the null hypothesis of zero yields

$$z = \frac{\widehat{\sigma}^2}{\widehat{\sigma}^2\sqrt{2/n}} = \sqrt{\frac{n}{2}}.$$

So, the test statistic depends only upon the sample size. In the present example, $n = 150$, so $z = \sqrt{75} = 8.660254$, matching the lavaan output.

**Bootstrapped standard errors** Re-focusing on direction of causality, the conclusion would appear to be pretty clear. Model One ($y_1 \rightarrow y_2$) is more consistent with the data than Model Two. Since these are simulated data and we knew all along that Model One was correct, it's basically a success. On the other hand, if these were real data and we didn't know the truth, there could be room for a bit of a bit of discomfort. All the tests are based on the assumption that the data come from multivariate normal distribution. While it's common and widely accepted to use normal theory even when a normal assumption is obviously wrong (for example, in linear regression), the objection is still legitimate.

The lavaan software has an option to produce standard errors using a bootstrap. For a model with only variances and covariances, this produces results that do not depend on the distribution of the sample data. Here's how it works.

The `se='bootstrap'` option on the `lavaan` function causes the software to create bootstrap data sets by repeatedly sampling $n$ rows of the data matrix with replacement. For each bootstrap data set, it estimates the parameters by maximum likelihood, and saves the numbers. The result is a sort of data file, with one column for each parameter

and one row for each bootstrap data set. The sample variance-covariance matrix from this data file (which is what you get from `vcov`, by the way) is a very good estimate of the asymptotic covariance matrix of the parameter estimates, regardless of the distribution of the sample data. The square roots of the diagonal elements of the matrix are the bootstrap standard errors of the parameter estimates. As in the fully parametric case, tests and confidence intervals are based on an approximate normal distribution for the parameter estimates.

It's important to distinguish between normality of the data and asymptotic normality of the parameter estimates. While it's true that the estimated variances and covariances for a lavaan model like `sigmod` are obtained by numerical maximum likelihood under a multivariate normal assumption, it also happens that in this case, the MLEs can be derived analytically rather than numerically – and the resulting parameter estimates are just the usual sample variances and covariances. By Theorem A.1 in Appendix A, the asymptotic distribution of sample variances and covariances is multivariate normal, assuming only that the data come from a joint distribution with finite fourth moments. So we are on really solid ground.

The following takes a minute or so. A little delay is understandable, since a numerical search is being carried out a thousand times. Setting the seed of the random number generator will produce exactly the same numbers every time, if that's important.

```
> # Standard errors by bootstrap
> sigfit2 = lavaan(sigmod,datta, se='bootstrap'); summary(sigfit2)
lavaan 0.6-7 ended normally after 74 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         10

  Number of observations                           150

Model Test User Model:

  Test statistic                                 0.000
  Degrees of freedom                                 0

Parameter Estimates:

  Standard errors                            Bootstrap
  Number of requested bootstrap draws             1000
  Number of successful bootstrap draws            1000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  x1 ~~
    x2      (sg12)   -0.089    0.307   -0.291    0.771
    y1      (sg13)    5.089    1.393    3.653    0.000
```

```
    y2      (sg14)    5.863    2.699    2.173    0.030
  x2 ~~
    y1      (sg23)    3.980    1.449    2.747    0.006
    y2      (sg24)    7.036    2.091    3.364    0.001
  y1 ~~
    y2      (sg34)   57.865    9.044    6.398    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    x1      (sg11)    3.943    0.728    5.415    0.000
    x2      (sg22)    4.874    1.147    4.251    0.000
    y1      (sg33)   62.296    8.115    7.677    0.000
    y2      (sg44)  135.380   17.987    7.527    0.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    con             394.688  511.400    0.772    0.440
```

The standard errors are sometimes smaller than the ones based on a normal model, and sometimes larger. It's all in the same ballpark, though, and none of the conclusions change. Model One is supported over Model Two.

**An easier way** Our concluions from the foregoing analysis were based on on a fairly deep understanding of the two candidate models, acquired by carrying out a series of calculations with Sage. It's nice to have an example like this, because it shows how two models differing only in direction of causality can have different implications for the covariance matrix – and those different implications open the door to an empirical decision about which way causality should be flowing in a path diagram. However, it was not particularly easy. For more realistic models with dozens of variables and possibly more than two competing models to consider, a job like this could be so demanding that people just wouldn't do it.

So why try to understand anything? If there are two plausible models with causality flowing in different directions, why not just estimate the parameters of both models and see which one fits the data better? This is actually not as dumb as it sounds. Recall from Chapter 1 (starting on page **??**) that the standard test of model fit is actually testing a collection of equality constraints that the model imposes on the covariance matrix. Suppose that two models imply the same constraints, like the models in Example 4.4.1 with $\gamma_1 = 0$. Then the chi-squared fit statistics will be identical, signalling that there is no chance of deciding between the models based on data. On the other hand, if the chi-squared fit statistics are not the same, then the models have different implications for the covariance matrix (at least at their respective MLEs[16]). Why not just give it a go?

```
> # Fit both models, compare LR tests.
```

[16] One thing that Example 4.4.1 tells us is that two models can imply the same covariance matrix in one region of the parameter space, but different covariance matrices in another region.

```
> model1 = '
+           # Model Equations
+           y1 ~ gamma1*x1 + gamma2*x2
+           y2 ~ gamma3*x2 + beta*y1
+           # Variances
+           x1~~phi1*x1
+           x2~~phi2*x2
+           y1~~psi1*y1 # Var(epsilon1) = psi1
+           y2~~psi2*y2 # Var(epsilon2) = psi2
+          '
> fit1 = lavaan(model1,data=datta); summary(fit1)
lavaan 0.6-7 ended normally after 30 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                          8

  Number of observations                           150

Model Test User Model:

  Test statistic                                 1.185
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.553

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  y1 ~
    x1      (gmm1)    1.310    0.297    4.405    0.000
    x2      (gmm2)    0.841    0.267    3.143    0.002
  y2 ~
    x2      (gmm3)    0.723    0.339    2.135    0.033
    y1      (beta)    0.883    0.095    9.334    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
    x1      (phi1)    3.943    0.455    8.660    0.000
    x2      (phi2)    4.874    0.563    8.660    0.000
   .y1      (psi1)   52.287    6.038    8.660    0.000
   .y2      (psi2)   79.216    9.147    8.660    0.000
```

The chi-squared test of fit has two degrees of freedom, one for each constraint. The null hypothesis is not rejected ($G^2 = 1.185$, $p = 0.553$),indicating that the model fits acceptably. The $z$-tests for both $\gamma_1$ and $\beta$ are comfortably significant. Model One is supported.

```
> model2 = '
+           # Model Equations
+           y1 ~ gamma1*x1 + gamma2*x2 + beta*y2
+           y2 ~ gamma3*x2
+           # Variances
+           x1~~phi1*x1
+           x2~~phi2*x2
+           y1~~psi1*y1
+           y2~~psi2*y2
+          '
> fit2 = lavaan(model2,data=datta); summary(fit2)
lavaan 0.6-7 ended normally after 26 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                          8

  Number of observations                           150

Model Test User Model:

  Test statistic                                11.392
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.003

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  y1 ~
    x1      (gmm1)    0.730    0.245    2.983    0.003
    x2      (gmm2)    0.279    0.229    1.221    0.222
    y2      (beta)    0.381    0.043    8.782    0.000
  y2 ~
    x2      (gmm3)    1.444    0.414    3.488    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
```

```
    x1      (phi1)    3.943    0.455    8.660    0.000
    x2      (phi2)    4.874    0.563    8.660    0.000
   .y1      (psi1)   35.406    4.088    8.660    0.000
   .y2      (psi2)  125.222   14.459    8.660    0.000
```

This time, the test for model (lack of) fit does reject the null hypothesis, with $G^2 = 11.392$, $df = 2$ and $p = 0.003$. By the usual $\alpha = 0.05$ significance level, this indicates an inadequate model fit. Tests for $\gamma_1$ and $\beta$ are significant again, so it's a clear-cut case. Model One is supported over Model Two.

It's true that some detail was lost. You can tell from the degrees of freedom that each model imposes two constraints on the covariance, and they can't be the same because the fit statistics are different. What you can't tell (unless you think about it a bit) is that one of the constraints is the same for both models, and it is supported, while the other degree of freedom is occupied by two different constraints, and the one implied by MOdel One is supported, while the one implied by Model Two is not. Still, it's not so bad, and not really different from what happens when people explore different models until they find one that fits.

**Can't we do better?** Nevertheless, there is a problem with the testing strategy used here. Imagine a situation in which the test for one model was barely non-significant, so technically that model fits, while the test for the other model was barely significant, so technically that model does not fit. Maybe the fit is actually pretty similar for the two models. What we want is a single test for whether one model fits better than the other one. An analogous problem comes up in experimental design. Suppose a training program significantly improves job satisfaction for women, but the test for men is not significant? That's not good enough. You need to test for the interaction of program and gender.

A very natural way to compare the fit of two models is to look at the difference between the two fit statistics. It's a likelihood ratio test, and easy to carry out with the `anova` function.

```
> # Try likelihood ratio test
> anova(fit1,fit2)
Chi-Squared Difference Test

     Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit1  2 3411.5 3435.6  1.185
fit2  2 3421.7 3445.8 11.392     10.207       0

Warning message:
In lavTestLRT(object = new("lavaan", version = "0.6.7", call = lavaan(model = model1,  :
  lavaan WARNING: some models have the same degrees of freedom
```

The test statistic is "correct" in that it is the difference between the two chi-squared fit statistics, but we are warned that the degrees of freedom equal zero, the difference between degrees of freedom for the two models. This would appear to be a dead end. It's not a dead end, but it is a moderately long story. Read on.

### 4.4.4 Testing difference between non-nested models

In the usual large-sample likelihood ratio chi-squared test due to Wilks [70], the model under the null hypothesis is a special case of the unrestricted or "full" model. For example, some of the parameters might be zero under $H_0$, or a collection of linear combinations of the parameters might be equal to specified constants. It's common to call such models "nested." When models are nested, under general conditions the large-sample distribution of the $G^2$ likelihood ratio statistic (twice the difference between log likelihoods; see expression A.38) in Appendix A) is chi-squared, with degrees of freedom equal to the number of equalities specified by the null hypothesis.

If two models differ only in the direction of causality, they are not nested. While the mainstream classical theory does not apply, there is an attractive technology for testing the difference between non-nested models. The story begins with an idea by Cox[17] [18] about the large-sample distribution of the likelihood ratio. Then in 1982, White [69] gave some techical conditions under which Cox's test was valid. Staying largely within White's framework, an influential paper by Quang Vuong [65] provided a more comprehensive treatment of testing non-nested hypotheses. The account here mostly follows Vuong. There is also a 2015 paper by Merkle, You and Preacher [45] that applies Vuong's work specifically to structural equation models. Merkle et al. provide the R package `nonnest2` for computing the procedures, which is very welcome.

**Framework** There are two competing models that seek to explain a random sample of observable data vectors $\mathbf{d}_1, \ldots, \mathbf{d}_n$. For our purposes, there is no harm in assuming the data vectors are real and of length $k$. Because it's a random sample, they are identically distributed. Each model implies a common probability distribution for the data vectors. Denote these distributions by their cumulative distribution functions, $F(\mathbf{d}; \boldsymbol{\theta})$ and $G(\mathbf{d}; \boldsymbol{\gamma})$. The probability distribution $F$ has parameter vector $\boldsymbol{\theta} \in \Theta$ and density or probability mass function $f(\mathbf{d}; \boldsymbol{\theta})$. The probability distribution $G$ has parameter vector $\boldsymbol{\gamma} \in \Gamma$ and density or probability mass function $g(\mathbf{d}; \boldsymbol{\gamma})$.

The two competing models may be nested, non-nested or overlapping. Different versions of the theory will apply in these three cases.

- Nested: The meaning of $G$ nested within $F$ is that for every $\boldsymbol{\gamma} \in \Gamma$, there is a $\boldsymbol{\theta} \in \Theta$ with $G(\mathbf{d}; \boldsymbol{\gamma}) = F(\mathbf{d}; \boldsymbol{\theta})$ for every $\mathbf{d} \in \mathbb{R}^k$. This corresponds to the usual meaning of nested, while allowing for it to be not obvious that $G$ is just a restricted version of $F$.

- Non-nested: For every $\boldsymbol{\theta} \in \Theta$ and every $\boldsymbol{\gamma} \in \Gamma$, there is at least one $\mathbf{d} \in \mathbb{R}^k$ with $G(\mathbf{d}; \boldsymbol{\gamma}) \neq F(\mathbf{d}; \boldsymbol{\theta})$. That is, the two probability distributions never coincide exactly.

- Overlapping: There is at least one pair $(\boldsymbol{\gamma}_1, \boldsymbol{\theta}_1)$ with $G(\mathbf{d}; \boldsymbol{\gamma}_1) = F(\mathbf{d}; \boldsymbol{\theta}_1)$ for every $\mathbf{d} \in \mathbb{R}^k$, and at least one pair $(\boldsymbol{\gamma}_2, \boldsymbol{\theta}_2)$ with $G(\mathbf{d}; \boldsymbol{\gamma}_2) \neq F(\mathbf{d}; \boldsymbol{\theta}_2)$ for some $\mathbf{d} \in \mathbb{R}^k$. That is, the two models may or may not imply different distributions.

[17] That's Sir David Roxbee Cox (1924-2022), who is also responsible for the Cox proportional hazards model in survival analysis, among other good things.

Our goal is to be able to test for difference between structural equation models that have the same variables, but with straight arrows running in different directions in part or all of the two models. In Example 4.4.1 (Figure 4.7), the two models imply the same probability distribution if $\gamma_1 = 0$ or $\beta = 0$, and otherwise they imply different probability distributions. So, this is an overlapping overlapping case. In fact, two models that differ only in direction of causation will always be overlapping. They can't be nested, and trivially, if all the arrows that go in different directions for the two models have coefficients of zero (like $\beta 0$ in Example 4.4.1), then the two models imply the same probability distribution. This means they can't be fully non-nested. The only remaining possibility is that they are overlapping. Our interest is in comparing overlapping models, but we will stay general until we are forced to specialize.

**Null hypothesis** In a refreshing departure from most statistical theory, neither of the two competing models is assumed to be correct. Let the true distribution of the observable data be denoted by $H(\mathbf{d})$. This distribution is unknown, and completely unlimited except for the existence of a density or probability mass function $h(\mathbf{d})$. The true distribution can be either discrete or continuous[18], and you don't have to specify which.

Undeniably, the better model is the one whose implied probability distribution is closer to the truth. Difference between two distributions is measured by the *Kullback-Leibler divergence*, also called the Kullback-Leibler information criterion. The divergence between $H(\cdot)$ and $F(\cdot;\boldsymbol{\theta})$ for a particular $\boldsymbol{\theta}$ is defined as the expected value of the log of a ratio of densities, where the expected value is taken with respect to the true distribution. Assuming for notational convenience that the true distribution is continuous, the divergence is defined as

$$E\left(\log\frac{h(\mathbf{d})}{f(\mathbf{d};\boldsymbol{\theta})}\right) = \int\cdots\int \log\left(\frac{h(\mathbf{d})}{f(\mathbf{d};\boldsymbol{\theta})}\right) h(\mathbf{d})\, d\mathbf{d}. \tag{4.9}$$

The Kullback-Leibler divergence is a kind of squared distance, but it's not a metric because it does not obey the triangle inequality [40].

The value of $\boldsymbol{\theta}$ that minimizes (4.9), thus getting the distribution $F$ as close to the truth as possible, is denoted $\boldsymbol{\theta}_*$, and is called the "pseudo-true value" of $\boldsymbol{\theta}$. Since

$$E\left(\log\frac{h(\mathbf{d})}{f(\mathbf{d};\boldsymbol{\theta}_*)}\right) = E\left(\log h(\mathbf{d})\right) - E\left(\log f(\mathbf{d};\boldsymbol{\theta}_*)\right) \geq 0$$

(trust me on the inequality, or consult [40]), it follows that the larger $E\left(\log f(\mathbf{d};\boldsymbol{\theta}_*)\right)$ is, the "better" the model is. The null hypothesis is that our two competing models are equally good (or "equivalent") in this sense. In symbols, the null hypothesis is

$$H_0 : E\left(\log f(\mathbf{d};\boldsymbol{\theta}_*)\right) = E\left(\log g(\mathbf{d};\boldsymbol{\gamma}_*)\right). \tag{4.10}$$

**A big theorem** It is astonishing and beautiful that even when the true distribution of the data is not in the family of distributions defined by $F$, the maximum likelihood

[18] Or it can be something else, like a mixed discrete-continuous distribution. Quite a few technical details are being suppressed here because this is an undergraduate text.

estimate of $\boldsymbol{\theta}$ assuming $F$ still converges almost surely to something, and the target is $\boldsymbol{\theta}_*$. That is, *the MLE is consistent for the pseudo-true value of* $\boldsymbol{\theta}$. In symbols, $\widehat{\boldsymbol{\theta}}_n \stackrel{as}{\to} \boldsymbol{\theta}_*$.

White [68] proves this result (it's his Theorem 2.2), and Vuong cites White. Huber [32] proved it fifteen years earlier (it's his Theorem 1) under conditions that were much less restrictive and more believable than White's. Throughout his excellent and justifiably influential paper, Vuong basically makes all the same assumptions that White did. At the end of the whole story, we will return to the technical conditions under which all this stuff is proved.

**The log likelihood ratio** To test the hull hypothesis (4.10), it makes sense to look at the estimated difference between $E\left(\log f(\mathbf{d};\boldsymbol{\theta}_*)\right)$ and $E\left(\log g(\mathbf{d};\boldsymbol{\gamma}_*)\right)$. These expected values can be estimated. Note that for $i = 1, \ldots, n$, the $\log f(\mathbf{d}_i;\boldsymbol{\theta}_*)$ are independent random variables. Then the law of large numbers says that their sample mean converges to their expected value. That is, $\frac{1}{n}\sum_{i=1}^{n} \log f(\mathbf{d}_i;\boldsymbol{\theta}_*) \stackrel{as}{\to} E\left(\log f(\mathbf{d};\boldsymbol{\theta}_*)\right)$. The pseudo-true parameter vectors are of course unknown, so replace them with consistent estimators, the MLEs. Then difference between the two estimated expected values is

$$
\begin{aligned}
& \frac{1}{n}\sum_{i=1}^{n} \log f(\mathbf{d}_i;\widehat{\boldsymbol{\theta}}_n) - \frac{1}{n}\sum_{i=1}^{n} \log g(\mathbf{d}_i;\widehat{\boldsymbol{\gamma}}_n) \\
= \; & \frac{1}{n}\sum_{i=1}^{n} \Big(\log f(\mathbf{d}_i;\widehat{\boldsymbol{\theta}}_n) - \log g(\mathbf{d}_i;\widehat{\boldsymbol{\gamma}}_n)\Big) \\
= \; & \frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{f(\mathbf{d}_i;\widehat{\boldsymbol{\theta}}_n)}{g(\mathbf{d}_i;\widehat{\boldsymbol{\gamma}}_n)}\right) \\
= \; & \frac{1}{n}\log\prod_{i=1}^{n} \frac{f(\mathbf{d}_i;\widehat{\boldsymbol{\theta}}_n)}{g(\mathbf{d}_i;\widehat{\boldsymbol{\gamma}}_n)} \\
= \; & \frac{1}{n}\log\frac{\prod_{i=1}^{n} f(\mathbf{d}_i;\widehat{\boldsymbol{\theta}}_n)}{\prod_{i=1}^{n} g(\mathbf{d}_i;\widehat{\boldsymbol{\gamma}}_n)},
\end{aligned}
\tag{4.11}
$$

which is $\frac{1}{n}$ times the log of the likelihood ratio. Following Vuong's [65] notation, the log likelihood ratio will be denoted by $LR_n(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\gamma}}_n)$, as follows:

$$
LR_n(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\gamma}}_n) = \sum_{i=1}^{n} \log\left(\frac{f(\mathbf{d}_i;\widehat{\boldsymbol{\theta}}_n)}{g(\mathbf{d}_i;\widehat{\boldsymbol{\gamma}}_n)}\right)
$$

Recalling that the usual large-sample likelihood ratio test statistic is twice a log likelihood ratio, (4.11) certainly points to $LR_n(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\gamma}}_n)$ as a tool for testing difference between the two models. The question is, what's the distribution of $LR_n(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\gamma}}_n)$ under the null hypothesis?

Cox [18] suggested that based on asymptotic normality of $\widehat{\boldsymbol{\theta}}_n$ and $\widehat{\boldsymbol{\gamma}}_n$, the log likelihood ratio has a limiting normal distribution, and he proposed a $z$-test. However, Vuong [65] showed that this conclusion depends on the two competing models implying different probability distributions *at the pseudo-true values* $\boldsymbol{\theta}_*$ and $\boldsymbol{\gamma}_*$.

There are two cases. In Case One, the two distributions are the same at their respective pseudo-true values. That is, $G(\mathbf{d};\boldsymbol{\gamma}_*) = F(\mathbf{d};\boldsymbol{\theta}_*)$ for every $\mathbf{d} \in \mathbb{R}^k$. This implies that the densities $g(\mathbf{d};\boldsymbol{\gamma}_*) = f(\mathbf{d};\boldsymbol{\theta}_*)$, except possibly on a set of probability zero in $\mathbb{R}^k$. In this case, Vuong shows that the sequence of random variables $2\,LR_n(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\gamma}}_n)$ (the usual test statistic, if the larger quantity is in the numerator) converges to a target that is a weighted sum of chi-squares, with elaborate formulas for the weights and degrees of freedom.

In our setting (overlapping models), Case One will always be a possibility, because $\boldsymbol{\theta}_*$ and $\boldsymbol{\gamma}_*$ are unknown. If Case One holds, then the two models imply probability distributions for the observable data that become identical when they get as close as possible to the true distribution. In this case, there is really no point in trying to test for difference in fit. The limiting distribution of $2\,LR_n(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\gamma}}_n)$ is of little practical use, but we are very interested in detecting whether Case One holds. If we can rule it out with a significance test, then we can and should proceed to Case Two.

In Case Two, $g(\mathbf{d};\boldsymbol{\gamma}_*)$ and $f(\mathbf{d};\boldsymbol{\theta}_*)$ are not equal, and testing for a difference in model fit makes sense. As given by Vuong's [65] Theorem 3.3, if the null hypothesis (4.10) is true and the two models imply distributions that are equally close to the truth in the limit without actually being identical, then

$$\sqrt{n}\,LR_n(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\gamma}}_n) \xrightarrow{d} x \sim N(0, \omega_*^2), \tag{4.12}$$

where

$$\omega_*^2 = Var\left(\log \frac{f(\mathbf{d};\boldsymbol{\theta}_*)}{g(\mathbf{d};\boldsymbol{\gamma}_*)}\right),$$

and the variance is computed under the true distribution of the observable data.

Now if $f(\mathbf{d};\boldsymbol{\theta}_*) = g(\mathbf{d};\boldsymbol{\gamma}_*)$, $\omega_*^2 = Var(0) = 0$, and (4.12) does not make sense. If the densities are not equal at the pseudo-true parameter values, then $\omega_*^2 > 0$ and (4.12) holds. To obtain a useable test statistic, we need a consistent estimator of the variance $\omega_*^2$. It's very natural. Note that for $i = 1, \ldots, n$, the $\log \frac{f(\mathbf{d}_i;\boldsymbol{\theta}_*)}{g(\mathbf{d}_i;\boldsymbol{\gamma}_*)}$ are independent and identically distributed random variables, with their distribution determined by $H$, the true distribution of the observable data, whatever that might be.

### 4.4.5 The moral of the story

The moral of this story is that in general, direction of causality cannot be determined empirically from a fitted structural equation model. For an unrestricted acyclic model, it's always impossible. For some cyclic models and some restricted acyclic models, it may be possible.

## 4.5 A Big Theorem

### 4.5.1 The Multiplication Theorem

### 4.5.2 Direct and Indirect Effects

## 4.6 The Exercise and Arthritis Pain Example

# Chapter 5

# Robustness

The word *robust* means strong and healthy. Something robust is not easy to break. A statistical method is said to be robust to some assumed condition if that condition does not really matter, and the method performs acceptably without it. Robustness usually emerges for large samples. In structural equation modelling, the usual tests and confidence intervals are based on an assumed multivariate normality of the observable data. This chapter is concerned with robustness with respect to the normality assumption. The approach is to use normal theory methods when they are robust, and modify them as necessary when they are not. The chapter begins with a summary of findings and a set of recommendations for what to do with data. After that, the reader is invited to join a voyage of discovery, and see where the knowledge came from[1].

## 5.1 Summary and Recommendations

Even when the data are not normally distributed, maximum likelihood based on multivariate normality yields estimates that are consistent and asymptotically normal. However, the normal theory standard errors of the estimates can under-estimate the true standard deviations of the sampling distributions of the model parameters. When this happens it's a problem, because the standard errors determine the width of confidence intervals, and are the denominators of $z$-tests for the parameters. When a standard error is too small, the confidence interval will be too narrow, implying more certainty about the true value of the parameter than the data warrant. A standard error that is too small will also inflate the $z$ statistic, and lead to rejection of the null hypothesis too often when the null hypothesis is true.

Standard errors are not always too small. For parameters that appear on the straight arrows in path diagrams, standard errors are robust, and normal theory results can be trusted for non-normal data. Also, there are strong indications that the standard errors for covariance parameters (which would appear on curved, double-headed arrows) are also

[1] My mother said to never end a sentence with a preposition. Her warm and cheerful ghost stands at my shoulder now, reminding me. I do not always listen to what she says. Sometimes, a preposition is a good thing to end a sentence with.

robust when the true covariance is zero — under the special condition that the random variables involved are truly independent, and not just uncorrelated.

For model parameters that are variances, and for covariance parameters when the true value is not zero, lack of robustness is most evident when the non-normal distributions involved have heavy tails (high kurtosis). For this situation, robust estimates of the asymptotic covariance matrix (which include robust standard errors) are available. When normal theory method fail, robust methods do better for the same sample size, but may require a considerably larger sample size to achieve performance that is actually good.

## 5.2 Robustness

Again, a statistical method is said to be robust with respect to some feature of the model if the method works acceptably even when the condition does not hold. The most familiar example of robustness might be the tests and confidence intervals based on the $t$ distribution. Suppose you have a simple random sample from a univariate normal distribution. After a fair amount of work that depends critically on properties of the normal distribution (for example, the independence of the sample mean and sample variance), one arrives at the statistic

$$t = \frac{\bar{x}_n - \mu}{s_n/\sqrt{n}} \sim t(n-1),$$

where $\bar{x}_n$ is the sample mean and $s_n$ is the sample standard deviation with $n-1$ in the denominator. On the other hand, suppose you are unwilling to assume that the data are normally distributed. Instead, you assume only random sampling from a distribution with finite variance. Then using the Central Limit Theorem along with a Slutsky theorem for convergence in distribution[2], one arrives at

$$z_n = \frac{\bar{x}_n - \mu}{s_n/\sqrt{n}} \xrightarrow{d} z \sim N(0,1).$$

The formulas for $t$ and $z_n$ are identical, although their derivations are very different. Combining this with the well-known fact that the $t$ distribution approaches a standard normal as the degrees of freedom tend to infinity, the conclusion is that for large samples, the assumed normal distribution basically does not matter. It's fine to go ahead and use $t$-tests and $t$ confidence intervals, even for binary data. This is a pure success story. If the data really happen to be normal, then using the $t$ distribution is optimal in all sorts of theoretically satisfying ways. If the data are not normal and the sample size is large, then everything's okay anyway.

**Multivariate normality** In every structural equation modeling software I have seen (including lavaan), the default method of parameter estimation is maximum likelihood.

[2]See item 6c in Section A.5.3 of Appendix A, or just take it on faith that it's okay to substitute $s_n$ for $\sigma$ in the Central Limit Theorem if $n$ is large.

That's maximum likelihood based on multivariate normality. With this option, all the usual tests and confidence intervals come from classical large-sample likelihood theory, and depend on the assumption that the observable data come from a multivariate normal distribution.

Well, what if the data are not multivariate normal? The very clear case of categorical data is treated in Chapter 9. Otherwise, when the data are quantitative, common practice is to ignore the issue, and to just go ahead and use likelihood-based methods. The assumption (usually unspoken) is that they are robust enough so the results will not be misleading. Now it's time to take a closer look, and if necessary make some adjustments.

**Robust alternatives** Some methods are designed from the outset to be distribution-free. The best known are variations on *weighted least squares*. Instead of minimizing the minus log likelihood or something equivalent over $\boldsymbol{\theta} \in \Theta$, one minimizes

$$F(\boldsymbol{\theta}) = (\widehat{\boldsymbol{\sigma}}_n - \boldsymbol{\sigma}(\boldsymbol{\theta}))^\top \mathbf{W}^{-1} (\widehat{\boldsymbol{\sigma}}_n - \boldsymbol{\sigma}(\boldsymbol{\theta})), \tag{5.1}$$

where $\widehat{\boldsymbol{\sigma}}_n = vech(\widehat{\boldsymbol{\Sigma}}_n)$ and $\boldsymbol{\sigma}(\boldsymbol{\theta}) = vech(\boldsymbol{\Sigma}(\boldsymbol{\theta}))$. The matrix $\mathbf{W}$ contains *weights*. With $\mathbf{W} = \mathbf{I}$, minimizing $F(\boldsymbol{\theta})$ reflects the very natural idea of minimizing the sum of squared differences between (a) the unique sample variances and covariances, and (b) the corresponding population variances and covariances, written as a function of the model parameters. The problem is that this would give equal weight to all the variances and covariances. Better would be to weight them inversely according to their variances, so that the sample moments with the lowest variance count most. But the sample moments have covariances, too. Browne's [13] Asymptotically Distribution Free (ADF) estimation method makes $\mathbf{W}$ the estimated covariance matrix of $\widehat{\boldsymbol{\sigma}}_n$[3].

It all works out great in theory as $n \to \infty$, but a huge sample size may be required for the method to work properly. Finney and DiStefano [26] describe simulation studies in which sample sizes as large as $n = 5,000$ were required for acceptable results. The ADF estimation method is implemented in most structural equation modlling software including lavaan, but in practice almost nobody uses it.

Another possibility is to estimate the model parameters directly by method of moments. Identifiability means that $\boldsymbol{\theta} = g(\boldsymbol{\Sigma})$, for some function $g(\cdot)$ that is usually continuous. Letting $\widehat{\boldsymbol{\theta}} = g(\widehat{\boldsymbol{\Sigma}})$ yields an estimator that is consistent by the law of large numbers and continuous mapping, and asymptotically normal by Theorem A.1 and the multivariate delta method. The method of moments estimator (52) for double measurement regression with latent variables is an example. The main problem with this as a general approach is that it requires explicit solutions for the covariance structure equations. While (52) is a nice general expression that does not casually throw away any information in the sample covariance matrix, it's specific to the double measurement regression model. For other models, explicit solutions to the covariance structure equations are frequently not

[3]The quantity being estimated is $\frac{1}{n}\mathbf{L}$, for the matrix $\mathbf{L}$ of Theorem A.1 in Appendix A. Estimation of $\mathbf{L}$ is by method of moments.

available, even when their existence can be established[4]. Another very practical issue is that even when the data are non-normal, maximum likelihood based on a normal assumption produces estimates that are just as accurate or more accurate than distribution-free methods. For example, a 1990 paper by Satorra and Bentler [54] describes Monte Carlo work suggesting that normal-theory methods generally perform better than Browne's [13] asymptotically distribution-free method.

In my judgement, methods designed from the outset to be distribution free are just not very promising for structural equation models. Therefore, the rest of this chapter will focus on likelihood-based methods, clarifying the ways in which they are robust with respect to the assumption of multivariate normality, and seeking modifications only when necessary. Throughout, it will be taken for granted that the model equations accurately represent the way in which the data are generated. That is, the robustness to be treated here is robustness with respect to the assumption of a normal distribution, period.

## 5.3 Estimation

Let us distinguish between the *distributional* part of a structural equation model, and the *structural* part. The distributional part of the model (partly) specifies probability distributions for the random variables involved. The structural part is basically everything else. It consists principally of the model equations, but also includes properties like certain random variables having zero covariance with one another, or certain parameters being equal to one another or equal to zero. The structural part of the model is what leads to $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\theta})$.

In general, when a statistical model is correct, maximum likelihood estimates are well known to be consistent [67]. Our Theorem 5.1 says that if the structural part of a structural equation model is correct, the normal theory MLE is consistent, even when the true distribution of the data is not multivariate normal. Consistency (See Section A.5.2 in Appendix A) amounts to large-sample accuracy. It's not the only thing we need, but it certainly helps.

Suppose that a statistical model is incorrectly specified. In our world, this could mean that the structural part is wrong, the distributional part is wrong, or both. It turns out that under general conditions given in a 1967 paper by Peter Huber [32], the MLE converges to a definite target. This target has been called the "pseudo-true" parameter value by Vuong [65]; also see the closely related treatment by White [68].[5] The proof of Theorem 5.1 works by showing that the pseudo-true value and the true parameter value must be the same. Convergence of the MLE to the pseudo-true parameter value is established by Huber's (1967) Theorem 2, whose mild requirements[6] are satisfied provided

[4]Chapters 3 and 4 develop a set of sufficient conditions for the existence of a unique solution of the covariance structure equations. They make identifiability easy to verify in many cases, but they do not produce explicit solutions.

[5]White uses the terms quasi-likelihood and quasi-maximum likelihood, but never refers to the target as quasi-true, at least in his 1982 paper. Pseudo-true parameter values are discussed further in Section 4.4.4 of Chapter 4.

[6]The conditions may be mild, but the proof is very demanding. This is a price one often pays for

that the true distribution(s) of the structural equation model (whatever they are) have finite variance.

The proof of Theorem 5.1 may give the wrong impression, because this is not an epsilon-delta book. In fact, it would have been possible to just cite Browne's (1984) Proposition One, which says almost the same thing. However, Browne assumes that the parameter space is closed and bounded, something that is not even true of the univariate normal distribution. When there is an alternative, it's more satisfying to avoid such unrealistic assumptions[7].

**Conditions for Theorem 5.1** Assume that the centered structural equation model (1.5) holds, with the distributions of the observed random variables unspecified except that their covariance matrix exists. Denote a general parameter vector by $\boldsymbol{\theta} \in \Theta$, an open subset[8] of $\mathbb{R}^t$. The true parameter vector is $\boldsymbol{\theta}_0 \in \Theta$.

The common covariance matrix of the $n$ independent data vectors is the $k \times k$ positive definite matrix $\boldsymbol{\Sigma}_0$. This is the true covariance matrix. The model equations correctly imply that $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}(\boldsymbol{\theta}_0)$. The model parameters are identifiable from the covariance matrix at the true parameter vector. Specifically, letting $\boldsymbol{\sigma}_0 = vech(\boldsymbol{\Sigma}_0)$, there is a function $g(\cdot)$ with $\boldsymbol{\theta}_0 = g(\boldsymbol{\sigma}_0)$. Assume that $g(\boldsymbol{\sigma})$ is continuous at $\boldsymbol{\sigma} = \boldsymbol{\sigma}_0$[9].

**Theorem 5.1** *Under the stated conditions, the maximum likelihood estimator* $\widehat{\boldsymbol{\theta}}_n$ *converges almost surely to the true parameter vector* $\boldsymbol{\theta}_0$.

**Proof** For the centered model under consideration, maximizing the multivariate normal likelihood (A.20) is equivalent to minimizing $q(\boldsymbol{\Sigma}, \widehat{\boldsymbol{\Sigma}}_n) = tr(\widehat{\boldsymbol{\Sigma}}_n\boldsymbol{\Sigma}^{-1}) - \log|\boldsymbol{\Sigma}|$ over all symmetric and positive definite matrices $\boldsymbol{\Sigma}$. By Theorem A.2 in Appendix A, a unique minimum occurs at $\boldsymbol{\Sigma} = \widehat{\boldsymbol{\Sigma}}_n$. This holds regardless of what $\widehat{\boldsymbol{\Sigma}}_n$ happens to be, as long as it is positive definite. Replacing $\widehat{\boldsymbol{\Sigma}}_n$ with a general symmetric and positive definite matrix $\mathbf{S}$, the conclusion is that for fixed $\mathbf{S}$ and any symmetric, positive definite $\boldsymbol{\Sigma}$,

$$q(\boldsymbol{\Sigma}, \mathbf{S}) \geq q(\mathbf{S}, \mathbf{S}), \tag{5.2}$$

generality. Starting with Huber's theorem is roughly like launching your airplane from the top of a mountain.

[7]Browne does offer an alternative, a set of technical conditions that would need to be checked separately for each model. In practice, people are just not going to do it. Browne is not alone in assuming that the parameter space is closed and bounded [65, 68]. It's technically easier to work in a closed and bounded space, because then uniform convergence is easier to establish, and this in turn can be used to justify exchange of limiting operations. The resulting conclusions may well be true in a more general setting; Theorem 5.1 is an example.

[8]The parameter space is open provided the model variance-covariance matrices are positive definite.

[9]Continuity is very natural. There are $k(+1)/2$ covariance structure equations in $t < k(+1)/2$ unknown parameters. Suppose that it is possible to solve for the parameters using only $t$ of the equations; this assumption is used to justify the degrees of freedom of the chi-squared test for model fit in Section 1.5.3. Then the solutions are at worst ratios of polynomials (algebraic expressions) in the $\sigma_{ij}$. Furthermore, identifiability guarantees that the denominators are non-zero at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, yielding continuity. I suppose that a solution requiring more equations than parameters might involve some strange, non-continuous function, but I have never seen an example.

with equality if and only if $\boldsymbol{\Sigma} = \mathbf{S}$.

For the structural equation model given in the Conditions, let $\widehat{\boldsymbol{\theta}}_n$ denote the maximum likelihood estimator of $\boldsymbol{\theta}_0$, based on an assumed multivariate normal distribution for the observed data. $\widehat{\boldsymbol{\theta}}_n$ is obtained by minimizing $q(\boldsymbol{\Sigma}(\boldsymbol{\theta}), \widehat{\boldsymbol{\Sigma}}_n)$ over $\boldsymbol{\theta} \in \Theta$. The conditions of Huber's [32] Theorem 2 are satisfied, implying $\widehat{\boldsymbol{\theta}}_n \stackrel{a.s.}{\to} \boldsymbol{\theta}_* \in \Theta$.

To produce a contradiction, suppose that $\boldsymbol{\theta}_* \neq \boldsymbol{\theta}_0$. Eventually, we will see this means that for large enough $n$, the likelihood function is greater at a certain method of moments estimate than it is at the maximum likelihood estimate. This is impossible, by the definition of an MLE.

The true covariance matrix is $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}(\boldsymbol{\theta}_0)$. Therefore, (5.2) implies that $q\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}_*), \boldsymbol{\Sigma}_0\right) \geq q\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}_0), \boldsymbol{\Sigma}_0\right)$, with equality if and only if $\boldsymbol{\Sigma}(\boldsymbol{\theta}_*) = \boldsymbol{\Sigma}(\boldsymbol{\theta}_0)$[10]. But $\boldsymbol{\Sigma}(\boldsymbol{\theta}_*) = \boldsymbol{\Sigma}(\boldsymbol{\theta}_0)$ is impossible, because identifiability means that two distinct parameter vectors cannot produce the same covariance matrix. It follows that $q\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}_*), \boldsymbol{\Sigma}_0\right)$ is strictly greater than $q\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}_0), \boldsymbol{\Sigma}_0\right)$.

It may help to re-express the loss function $q(\cdot, \cdot)$ in terms of real vectors. This will make it clear that the rest of the proof consists of statements about certain points in an ordinary Euclidean space, specifically an open subset of $\mathbb{R}^{t+k(k+1)/2}$, where $t$ is the number of model parameters and $k$ is the number of observable variables.

Let $\mathbf{t}$ be a general point in the parameter space $\Theta \subset \mathbb{R}^t$, and let $\mathbf{s} = vech(\mathbf{S})$, where $\mathbf{S}$ is a $k \times k$ symmetric and positive definite matrix. This means that $\mathbf{s} \in \mathbb{R}^{k(k+1)/2}$. Define

$$r(\mathbf{t}, \mathbf{s}) = q\left(\boldsymbol{\Sigma}(\mathbf{t}), \mathbf{S}\right) = tr(\mathbf{S}\boldsymbol{\Sigma}(\mathbf{t})^{-1}) - \log|\mathbf{S}|. \tag{5.3}$$

Using this notation and letting $\boldsymbol{\sigma}_0 = vech(\boldsymbol{\Sigma}_0)$, we have established above that $r(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0) > r(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$. Also, letting $\widehat{\boldsymbol{\sigma}}_n$ denote $vech(\widehat{\boldsymbol{\Sigma}}_n)$, the MLE $\widehat{\boldsymbol{\theta}}_n$ is obtained by minimizing $r(\boldsymbol{\theta}, \widehat{\boldsymbol{\sigma}}_n)$ over $\boldsymbol{\theta} \in \Theta$.

Let $\mathcal{S} = \{\mathbf{s} \in \mathbb{R}^{k(k+1)/2} : \mathbf{s} = vech(\mathbf{S}),$ where $\mathbf{S}$ is a $k \times k$ positive definite matrix$\}$. Because each $\mathbf{S}$ is positive definite, $\mathcal{S}$ is an open set[11]. Therefore the set $\Theta \times \mathcal{S}$ is open too.

Since the true covariance matrix $\boldsymbol{\Sigma}_0$ is positive definite, $\boldsymbol{\sigma}_0 \in \mathcal{S}$. Therefore the combined vector $(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$ is an interior point of $\Theta \times \mathcal{S}$. This makes it possible to establish an open neighbourhood of $(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$, consisting of points corresponding to valid parameter vectors and valid covariance matrices. The same applies to the point $(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0)$.

The function $r(\mathbf{t}, \mathbf{s})$ in 5.3 describes a collection of additions, multiplications and division by non-zero constants, combined with the natural log, which is a continuous function. Therefore, $r(\cdot, \cdot)$ is continuous at any combined vector $(\mathbf{t}, \mathbf{s}) \in \Theta \times \mathcal{S}$.

Recalling that $r(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0) > r(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$, let $\epsilon = \left(\, r(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0) - r(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0) \,\right)/3$. Continuity of the function $r(\cdot, \cdot)$ at $(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$ means there exists $\delta_1 > 0$ such that if the point $(\mathbf{t}, \mathbf{s})$ belongs to a spherical neighbourhood with radius $\delta_1$, centered at $(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$, then $|r(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0) - r(\mathbf{t}, \mathbf{s})| < \epsilon$. Similarly, there is a spherical neighbourhood with radius $\delta_2$, centered at $(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0)$, such that $|r(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0) - r(\mathbf{t}, \mathbf{s})| < \epsilon$ for all $(\mathbf{t}, \mathbf{s})$ in this second neighbourhood.

[10] In other words, for a "sample" covariance matrix equal to the truth, the minus log likelihood is minimized at the true parameter value. It will be seen that identifiability makes the minimum unique.

[11] I found a nice clean proof of this in the answer to a question on Stack Exchange.

Let $\delta = \min(\delta_1, \delta_2)$. Further shrink $\delta$ so that (1) the spherical neighbourhood with radius $\delta$ centered at $(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$ is entirely within $\Theta \times \mathcal{S}$, (2) the spherical neighbourhood with radius $\delta$ centered at $(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0)$ is entirely within $\Theta \times \mathcal{S}$, and (3) the two neighbourhoods do not overlap.

Sample variances and covariances converge almost surely to the corresponding true variances and covariances. This yields the vector convergence $\widehat{\boldsymbol{\sigma}}_n \stackrel{a.s.}{\to} \boldsymbol{\sigma}_0$. We will need this to hold at the same time as $\widehat{\boldsymbol{\theta}}_n \stackrel{a.s.}{\to} \boldsymbol{\theta}_*$. Recall that the random vectors here are vectors of scalar random variables, and those scalar random variables are functions from some underlying sample space $\Omega$ into the real numbers. The convergence of $\widehat{\boldsymbol{\sigma}}_n$ to $\boldsymbol{\sigma}$ and $\widehat{\boldsymbol{\theta}}_n$ to $\boldsymbol{\theta}_*$ might hold on two different subsets of $\Omega$, each of probability one. However, the intersection of these two sets is also a set of probability one. Denoting the intersection by $A$, we have both $\lim_{n\to\infty} \widehat{\boldsymbol{\sigma}}_n(\omega) = \boldsymbol{\sigma}$ and $\lim_{n\to\infty} \widehat{\boldsymbol{\theta}}_n(\omega) = \boldsymbol{\theta}_*$ for each $\omega \in A \subseteq \Omega$, with $P(A) = 1$.

Identifiability means that $\boldsymbol{\theta}_0 = g(\boldsymbol{\sigma}_0)$. Let $\widetilde{\boldsymbol{\theta}}_n = g(\widehat{\boldsymbol{\sigma}}_n)$, yielding a method of moments estimator for $\boldsymbol{\theta}_0$. By assumption, $g(\cdot)$ is continuous at $\boldsymbol{\sigma}_0$, so that $\widetilde{\boldsymbol{\theta}}_n \stackrel{a.s.}{\to} \boldsymbol{\theta}_0$. That is, the method of moments estimator $\widetilde{\boldsymbol{\theta}}_n$ is strongly consistent.

The combined vector $(\widetilde{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n) \stackrel{a.s.}{\to} (\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$, meaning that there exists an integer $N_1$ such that for all $n > N_1$, the vector of estimates $(\widetilde{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n)$ stays within the neighbourhood surrounding $(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)$. Consequently, $|r(\widetilde{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n) - r(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0)| < \epsilon$. Also, $(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n) \stackrel{a.s.}{\to} (\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0)$ implies the existence of an integer $N_2$ such that for all $n > N_2$, $(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n)$ stays within the neighbourhood surrounding $(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0)$, so that $|r(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n) - r(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0)| < \epsilon$.[12]

Figure 5.1 shows a picture. It's to scale, with $r(\boldsymbol{\theta}_*, \boldsymbol{\sigma}_0) - r(\boldsymbol{\theta}_0, \boldsymbol{\sigma}_0) = 3\,\epsilon$. Let $N =$

Figure 5.1: Non-overlapping intervals


$\max(N_1, N_2)$. For all $n > N$, $r(\widetilde{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n)$ is in the left-hand (lower) interval, and $r(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n)$ is in the right-hand (higher) interval — with probability one. So $r(\widetilde{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n) < r(\widehat{\boldsymbol{\theta}}_n, \widehat{\boldsymbol{\sigma}}_n)$. That's impossible, because $\widehat{\boldsymbol{\theta}}_n$ is obtained by minimizing $r(\boldsymbol{\theta}, \widehat{\boldsymbol{\sigma}}_n)$ over $\boldsymbol{\theta} \in \Theta$, and $\widetilde{\boldsymbol{\theta}}_n \in \Theta$.

This contradiction shows that the assumption $\boldsymbol{\theta}_* \neq \boldsymbol{\theta}_0$ must be wrong. Therefore, $\boldsymbol{\theta}_* = \boldsymbol{\theta}_0$, yielding $\widehat{\boldsymbol{\theta}}_n \stackrel{a.s.}{\to} \boldsymbol{\theta}_0$, the true parameter vector. ■

The practical conclusion is that in terms of parameter estimation, it's acceptable to use normal maximum likelihood regardless of the true distribution of the data. If the normal assumption is correct or approximately so, then the estimates will share some of the optimal properties of maximum likelihood. Otherwise, it's still okay provided that the sample size is large.

[12] In general, $N_1$ and $N_2$ will both depend on $\omega \in \Omega$. This is no problem. Randomly choose an $\omega$ from $\Omega$. With probability one, $\omega \in A$, and the whole argument applies for every element of the set $A$.

## 5.4 Asymptotic Normality

For as long as possible, we will continue to rely on the deep theory in Huber's 1967 paper [32]. Recall that his topic is the large-sample behaviour of maximum likelihood estimates when the statistical model is possibly incorrect, and we are applying his results to the normal-theory MLE for structural equation models, assuming the distributions might not be normal, but the rest of the model is correct.

Having proved that a class of estimators including the MLE converges to a definite target, Huber goes on to show in his Theorem Three and its corollary that under conditions that are slightly less general but still not very restrictive, the large-sample distribution of the estimator around the target is approximately multivariate normal.

Theorem 5.1 of this textbook establishes that in our setting, Huber's large-sample target is identical to the true parameter vector. If we assume that the true probability distributions of our observed variables have finite fourth moments and choose the function $u(\cdot,\cdot)$ as described below, then we also have asymptotic normality.

Huber expresses his results using certain Greek letters that mean something entirely different in structural equation modelling. To minimize confusion, some of his notation has been modified[13] in the following.

**Corollary to Huber's corollary** Assume the conditions of Theorem 5.1, and also that the true joint distribution of the observed variables possess finite fourth moments. Then

$$\mathbf{t}_n = \sqrt{n}(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \xrightarrow{d} \mathbf{t} \sim N_r(\mathbf{0}, \mathbf{V}), \tag{5.4}$$

where $r$ is the number of unknown model parameters. The matrix $\mathbf{V}$ is calculated as follows.

1. In the multivariate normal density (A.19), center the observed data by lettting $\mathbf{d} = \mathbf{x} - \boldsymbol{\mu}$. The result is

$$f(\mathbf{d};\boldsymbol{\Sigma}) = \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}(2\pi)^{\frac{k}{2}}} \exp\left\{-\frac{1}{2}\mathbf{d}^\top\boldsymbol{\Sigma}^{-1}\mathbf{d}\right\}.$$

2. Writing the covariance matrix as a function of the model parameters and letting $f = f\left(\mathbf{d};\boldsymbol{\Sigma}(\boldsymbol{\theta})\right)$, calculate the $r \times 1$ gradient

$$u(\mathbf{d},\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial \log f}{\partial \theta_1} \\ \vdots \\ \frac{\partial \log f}{\partial \theta_r} \end{pmatrix}.$$

[13] Huber denotes a general parameter vector by $\theta$, while we use $\boldsymbol{\theta}$; that's fine. His large-sample target (the "pseudo-true" parameter vector) is $\theta_0$. We use $\boldsymbol{\theta}_0$ for the vector of true parameter values, and $\boldsymbol{\theta}_*$ for the pseudo-true parameter value. By Theorem 5.1, $\boldsymbol{\theta}_* = \boldsymbol{\theta}_0$, so our purposes, Huber's $\theta_0$ is our $\boldsymbol{\theta}_0$. However, Huber's $\psi(x,\theta)$ is our $u(\mathbf{d},\boldsymbol{\theta})$, his $\lambda(\theta)$ is our $h(\boldsymbol{\theta})$, and his matrix $\Lambda$ is our $\mathbf{A}$. We are not using the letter $C$ for anything in particular, so we stay with his notation there. Our $\mathbf{C}$ is the same as Huber's $C$.

3. Calculate the $r \times 1$ vector of functions $h(\boldsymbol{\theta}) = E\left(u(\mathbf{d}, \boldsymbol{\theta})\right) = [h_j]$, where the expected value is taken with respect to the unknown true distribution of $\mathbf{d}$. Not knowing the true distribution presents no difficulties. Just use expected value signs[14].

4. Calculate the $r \times r$ matrix of partial derivatives

$$\mathbf{A} = \left[\frac{\partial h_i}{\partial \theta_j}\right] = \begin{pmatrix} \frac{\partial h_1}{\partial \theta_1} & \frac{\partial h_1}{\partial \theta_2} & \cdots & \frac{\partial h_1}{\partial \theta_r} \\ \frac{\partial h_2}{\partial \theta_1} & \frac{\partial h_2}{\partial \theta_2} & \cdots & \frac{\partial h_2}{\partial \theta_r} \\ \vdots & \vdots & & \vdots \\ \frac{\partial h_r}{\partial \theta_1} & \frac{\partial h_r}{\partial \theta_2} & \cdots & \frac{\partial h_r}{\partial \theta_r} \end{pmatrix}$$

5. Recognizing that $u(\mathbf{d}, \boldsymbol{\theta})$ is an $r \times 1$ random vector, calculate its covariance matrix: $\mathbf{C} = cov\left(u(\mathbf{d}, \boldsymbol{\theta})\right)$. The covariance operation is carried out with respect to the unknown true distribution[15].

6. Finally, letting $\mathbf{A}_0$ denote the matrix $\mathbf{A}$ evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ and letting $\mathbf{C}_0$ denote $\mathbf{C}$ evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, calculate

$$\mathbf{V} = \mathbf{A}_0^{-1}\,\mathbf{C}_0\,\left(\mathbf{A}_0^{\top}\right)^{-1}. \tag{5.5}$$

This is the $\mathbf{V}$ in $\mathbf{t}_n = \sqrt{n}(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \xrightarrow{d} \mathbf{t} \sim N_r(\mathbf{0}, \mathbf{V})$. It's the asymptotic covariance matrix of $\mathbf{t}_n$, not $\widehat{\boldsymbol{\theta}}_n$. The asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}_n$ is $\frac{1}{n}\mathbf{V}$.

**Example 5.4.1** *A simple random sample*

Let $x_1 \ldots, x_n$ be independent and identically distributed random variables from a distribution with expected value $\mu$ and variance $\sigma^2$, not necessarily normal. There are $r = 2$ parameters. The pair of true parameter values $(\mu, \sigma^2)$ is a point in the parameter space $\Theta = \{(\theta_1, \theta_2) : -\infty < \theta_1 < \infty, \theta_2 > 0\}$. Using normal maximum likelihood even though the data may not be normal, we obtain

$$\widehat{\mu} = \overline{x}_n, \text{ and } \widehat{\sigma}^2_n = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x}_n)^2.$$

It is clear that so far, pretending that the data are normally distributed has done no harm. In addition to being MLEs, $\widehat{\mu}$ and $\widehat{\sigma}^2$ are natural method of moments estimators, and consistency follows from the law of large numbers (and continuous mapping, in the case of the variance) without any fancy machinery.

The objective will be to calculate a robust asymptotic covariance matrix for $(\widehat{\mu}, \widehat{\sigma}^2)$, following the 6-step recipe outlined for structural equation models. It will be necessary to

[14] Actually, all the resulting expressions are in terms of variances and covariances of the observed data. You can read these directly from $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, since that part of the model is correct.

[15] Again, just use expected value signs. This is the source of the higher order (up to fourth order) central and product moments in the final answer.

make minor adjustments for the fact that this is not a centered structural equation model, but still it's a good example. Numbering of the steps corresponds to the numbering in the recipe. The main difference in notation is that the data vector $\mathbf{d}$ is replaced by the scalar random variable $x$.

1. Because the mean is one of the parameters here, it will not be hidden by centering. Write the normal density as

$$f = f(x;\theta_1,\theta_2) = \frac{1}{\theta_2^{1/2}\sqrt{2\pi}} \exp -\frac{1}{2}(x-\theta_1)^2\,\theta_2^{-1}.$$

2. Now calculate the gradient

$$u(x,\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial \log f}{\partial \theta_1} \\ \frac{\partial \log f}{\partial \theta_2} \end{pmatrix} = \begin{pmatrix} (x-\theta_1)\theta_2^{-1} \\ \frac{1}{2}(x-\theta_1)^2\theta_2^{-2} - \frac{1}{2}\theta_2^{-1} \end{pmatrix}.$$

3. Take the expected value, yielding $h(\boldsymbol{\theta}) = E\left(u(x,\boldsymbol{\theta})\right)$

$$h(\boldsymbol{\theta}) = E\left(u(d,\boldsymbol{\theta})\right) = \begin{pmatrix} (E(x)-\theta_1)\,\theta_2^{-1} \\ \frac{1}{2}E(x-\theta_1)^2\,\theta_2^{-2} - \frac{1}{2}\theta_2^{-1} \end{pmatrix}$$

   The expected values are taken with respect to the true distribution, whatever it is. So the expected value of $x$ is $\mu$, not $\theta_1$. Similarly, $E(x-\theta_1)^2 \neq \sigma^2$. The symbols $\theta_1$ and $\theta_2$ are variables, with respect to which we will presently differentiate. Another calculation step yields

$$h(\boldsymbol{\theta}) = \begin{pmatrix} (\mu-\theta_1)\,\theta_2^{-1} \\ \frac{1}{2}\sigma^2\theta_2^{-2} + \frac{1}{2}(\mu-\theta_1)^2\theta_2^{-2} - \frac{1}{2}\theta_2^{-1} \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}.$$

4. Now partially differentiate, yielding the Jacobian

$$\mathbf{A} = \begin{pmatrix} \frac{\partial h_1}{\partial \theta_1} & \frac{\partial h_1}{\partial \theta_2} \\ \frac{\partial h_2}{\partial \theta_1} & \frac{\partial h_2}{\partial \theta_2} \end{pmatrix} = \begin{pmatrix} -\theta_2^{-1} & (\theta_1-\mu)\,\theta_2^{-2} \\ (\theta_1-\mu)\,\theta_2^{-2} & -\sigma^2\theta_2^{-3} - (\theta_1-\mu)^2\theta_2^{-3} + \frac{1}{2}\theta_2^{-2} \end{pmatrix}$$

5. Recognizing that the quantity $u(x,\boldsymbol{\theta})$ from step 2 is a random vector, we need to calculate $\mathbf{C} = cov(u(x,\boldsymbol{\theta}))$. In Step 6, this matrix is going to be evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. It's a little more convenient to evaluate at the true parameter values first, obtaining

$$\mathbf{C}_0 = cov\left(u(d,\boldsymbol{\theta}_0)\right) = cov\begin{pmatrix} \frac{d-\mu}{\sigma^2} \\ \frac{(d-\mu)^2}{2\sigma^4} - \frac{1}{2\sigma^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sigma^2} & \frac{E(d-\mu)^3}{2\sigma^6} \\ \frac{E(d-\mu)^3}{2\sigma^6} & \frac{E(d-\mu)^4-\sigma^4}{4\sigma^8} \end{pmatrix}$$

6. The last step is to calculate the matrix $\mathbf{V} = \mathbf{A}_0^{-1}\,\mathbf{C}_0\,\left(\mathbf{A}_0^{\top}\right)^{-1}$. Evaluating the matrix $\mathbf{A}$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, we get

$$\mathbf{A}_0 = \begin{pmatrix} -\frac{1}{\sigma^2} & 0 \\ 0 & -\frac{1}{2\,\sigma^4} \end{pmatrix}$$

And finally,

$$\mathbf{V} = \mathbf{A}_0^{-1}\,\mathbf{C}_0\,\mathbf{A}_0^{-1} = \begin{pmatrix} \sigma^2 & E(x-\mu)^3 \\ E(x-\mu)^3 & E(x-\mu)^4 - \sigma^4 \end{pmatrix}$$

This is the matrix $\mathbf{V}$ that appears in $\sqrt{n}(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \xrightarrow{d} \mathbf{t} \sim N_2(\mathbf{0}, \mathbf{V})$.

It is instructive to compare compare $\mathbf{V}$ with its normal theory counterpart, the inverse of the Fisher information in one observation. In Table 5.1, the expressions are divided by $n$, yielding asymptotic covariance matrices for the sample mean and the sample variance.

Table 5.1: Two asymptotic covariance matrices for $(\overline{x}_n, \widehat{\sigma}_n^2)$

| Inverse of Fisher Information | Robust Huber |
|---|---|
| $\begin{pmatrix} \frac{\sigma^2}{n} & 0 \\ 0 & \frac{2\sigma^4}{n} \end{pmatrix}$ | $\begin{pmatrix} \frac{\sigma^2}{n} & \frac{E(x-\mu)^3}{n} \\ \frac{E(x-\mu)^3}{n} & \frac{E(x-\mu)^4-\sigma^4}{n} \end{pmatrix}$ |

For the normal distribution, $E(x-\mu)^3 = 0$ and $E(x-\mu)^4 = 3\sigma^4$, so when the distribution really is normal, the robust asymptotic covariance matrix corresponds to the usual answer. Note that the *skewness* of a distribution is defined as $\frac{E(x-\mu)^3}{\sigma^3}$, and the *kurtosis*[16] is defined as $\frac{E(x-\mu)^4}{\sigma^4}$. Thus, the robust asymptotic covariance matrix depends on the true distribution only through its skewness and kurtosis.

**Computing it** The recipe for $\mathbf{V}$ is reasonably easy to follow for a univariate random sample. However, even for a small structural equation model, the expressions get quite messy if you follow the directions in a straightforward way using only standard multivariable calculus. This is even true with Sage, because while Sage can easily calculate partial derivatives and perform matrix operations, it currently is not very good at expected values and covariances. Fortunately, with some thought it's possible to get expressions that

[16] Kurtosis is a way of expressing how heavy-tailed the distribution is. A heavy-tailed distribution has relatively large probability out on the tails, and will frequently generate extreme observations that seem like outliers. The kurtosis of the normal distribution is 3, and *excess* kurtosis is defined as kurtosis minus 3, to facilitate comparison with the normal.

are general and fairly compact — for example using Jacobi's formula for the derivative of a determinant.

The matrix $\mathbf{V}$ is the *true* asymptotic covariance matrix of $\mathbf{t}_n$. It's a function of the true parameters, and also of various central moments and product moments of the true distribution. What's needed for practical purposes is a consistent estimate of $\mathbf{V}$. That's not a problem if you have a formula for $\mathbf{V}$. Just estimate the model parameters with the MLEs, and use method of moments to estimate the expected values. You never even need to guess at the true distribution of the observable variables. All you need is estimates of some of the higher moments.

**The Satorra-Bentler Estimator** Huber's distribution-free asymptotic covariance matrix for the vector of MLEs is not the only available choice. In a 2022 paper, Savalei and Rosseel [56] lay out a dizzying array of alternatives, many designed for use with data sets with missing data, where the missing data are assumed to arise by certain mechanisms. Here, the problem of missing data will be set aside[17], and we will focus on a classical solution due to Satorra and Bentler [54]. Their estimate seems to have first appeared in an earlier paper by Bentler and Dykstra [6], using a different notation. In the following, their notation is largely replaced with the notation of this text[18].

Assume a structural model leading to $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\theta})$. The true parameter vector is $\boldsymbol{\theta}_0$, and its normal theory MLE is $\widehat{\boldsymbol{\theta}}_n$. The sample covariance matrix (with $n$ in the denominator) is denoted by $\widehat{\boldsymbol{\Sigma}}_n$. Also, let $\widehat{\boldsymbol{\sigma}}_n = vech(\widehat{\boldsymbol{\Sigma}}_n)$, $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\theta}) = vech(\boldsymbol{\Sigma}(\boldsymbol{\theta}))$ and $\boldsymbol{\sigma}_0 = \boldsymbol{\sigma}(\boldsymbol{\theta}_0)$. In an expression based on Equation 1.19 on page 171, let

$$b(\boldsymbol{\sigma}) = tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) - k - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}|.$$

The number of moments (unique variances and covariances) is $m = k(k+1)/2$. Define the $m$ by $m$ Hessian matrix $\mathbf{H}$ as

$$\mathbf{H} = \left[\frac{\partial^2 b}{\partial\sigma_i\partial\sigma_j}\right]_{(\widehat{\boldsymbol{\sigma}},\boldsymbol{\sigma})=(\boldsymbol{\sigma}_0,\boldsymbol{\sigma}_0)}.$$

After differentiation, $\mathbf{H}$ is a matrix of expressions in the $\sigma_j$ and $\widehat{\sigma}_j$. The notation says to then evaluate the result at $\sigma_j = \sigma_j(\boldsymbol{\theta}_0)$ and $\widehat{\sigma}_j = \sigma_j(\boldsymbol{\theta}_0)$ for $j = 1, \ldots, m$. Note that since $\boldsymbol{\sigma}_0 = \boldsymbol{\sigma}(\boldsymbol{\theta}_0)$, the final $\mathbf{H}$ is a function of the true parameter vector $\boldsymbol{\theta}_0$.

[17] In lavaan and most other software I have seen, the default way of treating missing data is *listwise deletion*, in which an entire case (respondent, subject) is omitted if any data are missing. That's okay if there are just a few cases with missing data. For more extensive missing data, one can calculate the sample covariance matrix with *pairwise deletion*. In pairwise deletion, each individual variance or covariance is calculated using all available data. In R, the `var` function's `use='pairwise.complete.obs'` option will do the trick. The resulting covariance matrix is then used as input to the software. With pairwise deletion and a large volume of missing data, you might start to wonder that $n$ should be. If you find yourself wondering, the situation is serious enough to consider the alternatives outlined by Savalei and Rosseel [56]. They are all easy to compute, since Rosseel is the author of lavaan

[18] Their $F_{ML}(\theta)$ is replaced by our $b(\boldsymbol{\theta})$. Their $S$ is our $\widehat{\boldsymbol{\Sigma}}$. Their $V$ is replaced by $\mathbf{H}$, because it's a Hessian, and because we are using $\mathbf{V}$ for something else. Their $\Gamma$ is our $\mathbf{L}$

Then, let

$$\boldsymbol{\Delta} = \left[\frac{\partial \sigma_i(\boldsymbol{\theta})}{\partial \theta_j}\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \text{ and } \mathbf{J} = \boldsymbol{\Delta}^\top \mathbf{H} \boldsymbol{\Delta}.$$

The "bread" in the Sattora-Bentler sandwich estimator is assembled from the matrices $\mathbf{J}$, $\mathbf{H}$ and $\boldsymbol{\Delta}$, in a way that will be seen. The "meat" is a critical matrix that Satorra and Bentler denote by $\Gamma$. In this book, we are using $\boldsymbol{\Gamma}$ for a matrix of regression coefficients in the latent variable model. Satorra and Bentler's $\Gamma$ is the same as the matrix $\mathbf{L}$ that appears in our Theorem A.1, in Appendix A. There, we have (repeating from page 564)

$$\sqrt{n}\left(vech(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma})\right) \xrightarrow{d} \mathbf{t} \sim N(\mathbf{0}, \mathbf{L}).$$

In Theorem A.1, the data $\mathbf{d}_1, \ldots, \mathbf{d}_n$ have common expected value $\boldsymbol{\mu} = E(\mathbf{d}_1)$ and covariance matrix $\boldsymbol{\Sigma} = cov(\mathbf{d}_1)$. Letting $\mathbf{w} = vech\{(\mathbf{d}_1 - \boldsymbol{\mu})(\mathbf{d}_1 - \boldsymbol{\mu})^\top\}$, the matrix $\mathbf{L}$ is given by $\mathbf{L} = cov(\mathbf{w})$. So $\mathbf{L}$ is not quite the asymptotic covariance matrix of $\widehat{\boldsymbol{\sigma}} = vech(\widehat{\boldsymbol{\Sigma}})$. The asymptotic covariance matrix of $\widehat{\boldsymbol{\sigma}}$ is $\frac{1}{n}\mathbf{L}$.

With this background, Satorra and Bentler's [54] expression (2.11) on p. 239 says

$$\sqrt{n}\left(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0\right) \xrightarrow{d} \mathbf{t} \sim N(\mathbf{0}, \mathbf{J}^{-1}\boldsymbol{\Delta}^\top \mathbf{H}\mathbf{L}\mathbf{H}\boldsymbol{\Delta}\mathbf{J}^{-1}).$$

Thus, using the notation *acov* for the asymptotic covariance matrix,

$$acov(\widehat{\boldsymbol{\theta}}_n) = \frac{1}{n}\mathbf{J}^{-1}\boldsymbol{\Delta}^\top \mathbf{H}\mathbf{L}\mathbf{H}\boldsymbol{\Delta}\mathbf{J}^{-1}. \tag{5.6}$$

Here is a rough justification of the Satorra-Bentler estimator. Denote the $r \times r$ asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}_n$ by $\frac{1}{n}\mathbf{V}$. The model says that $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\theta})$. If the model is correct, this implies $\widehat{\boldsymbol{\sigma}} = \boldsymbol{\sigma}(\widehat{\boldsymbol{\theta}}_n)$, approximately for large $n$. In the notation of the multivariate delta method, the function $\boldsymbol{\sigma}(\boldsymbol{\theta})$ is $\mathbf{g}$, and $\dot{\boldsymbol{\sigma}}$ is $\dot{\mathbf{g}}$. Then

$$\begin{aligned} acov(\widehat{\boldsymbol{\sigma}}) &= \dot{\boldsymbol{\sigma}}\, acov(\widehat{\boldsymbol{\theta}}_n)\, \dot{\boldsymbol{\sigma}}^\top \\ &= \boldsymbol{\Delta}\frac{1}{n}\mathbf{L}\boldsymbol{\Delta}^\top. \end{aligned}$$

Consequently,

$$\begin{aligned} \frac{1}{n}\mathbf{L} = \boldsymbol{\Delta}\frac{1}{n}\mathbf{V}\boldsymbol{\Delta}^\top \quad &\Longrightarrow \quad \mathbf{L} = \boldsymbol{\Delta}\mathbf{V}\boldsymbol{\Delta}^\top \\ &\Longrightarrow \quad \mathbf{J}^{-1}\boldsymbol{\Delta}^\top \mathbf{H}\mathbf{L}\mathbf{H}\,\boldsymbol{\Delta}\mathbf{J}^{-1} = \mathbf{J}^{-1}\boldsymbol{\Delta}^\top \mathbf{H}\boldsymbol{\Delta}\mathbf{V}\boldsymbol{\Delta}^\top \mathbf{H}\,\boldsymbol{\Delta}\mathbf{J}^{-1} \\ &\Longrightarrow \quad \mathbf{J}^{-1}\boldsymbol{\Delta}^\top \mathbf{H}\mathbf{L}\mathbf{H}\,\boldsymbol{\Delta}\mathbf{J}^{-1} = \underbrace{(\boldsymbol{\Delta}^\top \mathbf{H}\boldsymbol{\Delta})^{-1}\boldsymbol{\Delta}^\top \mathbf{H}\boldsymbol{\Delta}}_{\mathbf{I}}\,\mathbf{V}\,\underbrace{\boldsymbol{\Delta}^\top \mathbf{H}\,\boldsymbol{\Delta}(\boldsymbol{\Delta}^\top \mathbf{H}\boldsymbol{\Delta})^{-1}}_{\mathbf{I}} \\ &\Longrightarrow \quad \frac{1}{n}\mathbf{V} = acov(\widehat{\boldsymbol{\theta}}_n) = \frac{1}{n}\mathbf{J}^{-1}\boldsymbol{\Delta}^\top \mathbf{H}\mathbf{L}\mathbf{H}\,\boldsymbol{\Delta}\mathbf{J}^{-1}, \end{aligned}$$

which is Satorra and Bentler's formula.

**Huber and Satorra-Bentler agree** On the surface, we have two sandwich-type formulas for the asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}_n$. From Expression 5.5, the Huber version is $\frac{1}{n}\mathbf{A}_0^{-1}\,\mathbf{C}_0\,\left(\mathbf{A}_0^{\top}\right)^{-1}$, while as given above, Satorra and Bentler's double decker sandwich formula is $\frac{1}{n}\mathbf{J}^{-1}\boldsymbol{\Delta}^{\top}\mathbf{HLH}\,\boldsymbol{\Delta}\mathbf{J}^{-1}$. In Huber's very general theory, not all the partial derivaties in Satorra and Bentler's sandwich estimator need to exist, but for a linear structural equation model with a multivariate normal likelihood, they do. As a result, the Huber and Satorra-Bentler asymptotic covariance matrices are equal when evaluated at the true parameter values [56]. When they are evaluated at the MLEs (the lavaan default) they produce equal estimates of $acov(\widehat{\boldsymbol{\theta}}_n)$. In R's lavaan package, they are available through the `se = 'robust.huber.white'` or `se = 'robust.sem'` option in the `lavaan` function. The standard errors are square roots of the diagonal elements of $\frac{1}{n}\widehat{\mathbf{V}}$, and the `vcov` function returns the entire matrix.

One should not expect particularly good performance for small sample sizes, quite apart from the fact that all the theory is asymptotic. As the number of observed variables increases, the number of relevant moments and product moments (like $E\left\{(x_j-\mu_j)(x_k-\mu_k)^3\right\}$ which would be estimated by $\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\overline{x}_j)(x_{ik}-\overline{x}_k)^3$) increases very fast. While the storage and processing requirements are no longer much of an issue with modern computers, the sample size required for all those estimates to be accurate at the same time could be substantial. Also, estimating something like the fourth moment of a heavy-tailed distribution will naturally require a large sample. At least a few extreme observations are guaranteed, and they are going to have a noticeable effect on the MLEs. A large amount of data may be required to get a reading on the shape of the distribution out on the tails. All this has no impact on the theory, because the theory is all about what happens as $n \to \infty$. For applications to real data, it can matter.

**Bootstrap** The bootstrap provides another way to estimate $\frac{1}{n}\mathbf{V}$, one that avoids all the partial derivatives and expected values. As described in Appendix A[19], the idea behind the bootstrap is that if the sample size is large enough, the sample closely resembles the population from which it is selected. In that case, sampling from the sample with replacement (re-sampling) is a lot like sampling from the original population.

The `se='bootstrap'` option of the `lavaan` function causes the software to create bootstrap data sets by repeatedly sampling $n$ rows of the data matrix with replacement. For each bootstrap data set, it estimates the parameters by maximum likelihood, and saves the numbers. The result is a sort of data file, with one column for each parameter and one row for each bootstrap data set. The sample variance-covariance matrix from this data file (which is what you get from `vcov`, by the way) is a very good estimate of the asymptotic covariance matrix of the parameter estimates, regardless of the distribution of the sample data. The square roots of the diagonal elements of the matrix are the bootstrap standard errors of the parameter estimates. The tests and confidence intervals are based on an approximate normal distribution for the parameter estimates, a property that the corollary to Huber's corollary guarantees under very general conditions.

The bootstrap is a beautiful tool; it's flexible, intuitive and easy to code. The main

[19] Or anyway, it will be described once I put a little bootstrap section in there.

downside is that it takes a while to run. By default, lavaan will draw 1,000 bootstrap samples, and that involves estimating the parameters by numerical maximum likelihood 1,000 times. You might have to wait a minute or two. Also, because it's based on random number generation, the numerical answers will vary slightly if you re-run the analysis. To get exactly the same numbers each time, use `set.seed()` before fitting the model. Give the `set.seed()` function a large integer argument.

## 5.5 Does it make a difference?

This is a serious question. Do robust standard errors actually improve the quality of inference when the data are not normal? Or, given the consistency and asymptotic normality of maximum likelihood estimators for non-normal data, are the classical normal-theory methods good enough? Does it matter?

There are different answers to this question, and I will spoil the surprise by telling you my answer. Sometimes it matters, and sometimes it does not. Table 5.1 tells a small version of the story. Look at the upper left entries of the two asymptotic covariance matrices. In both cases, it's $\frac{\sigma^2}{n}$. Everybody knows that the variance of $\overline{x}_n$ is $\frac{\sigma^2}{n}$, and the central limit theorem tells us that the distribution of $\overline{x}_n$ approximately normal for large samples, regardless of the distribution of the data[20]. The lower right entries, representing the variance of $\widehat{\sigma}^2_n$, are a different matter. What happens here will depend on the true distribution of the data. If the value of $E(x-\mu)^4 - \sigma^4$ is greater for the true distribution than for the normal, then the normal theory standard error of $\widehat{\sigma}^2_n$ will under-estimate the true value. The result will be 95% confidence limits that are too narrow, and capture the true parameter value less than 95% of the time. Similarly, $z$-tests will have a denominator that is too small. As a result, tests will reject a true null hypothesis with probability higher than the supposed Type I error rate. Tests and confidence intervals involving both parameters at once will be affected by any skewness captured in the off-diagonal element of the matrix.

So even in the same model, normal theory inference could be okay for some parameters, and badly flawed for others. This is perfectly clear for the simple random sample of Example 5.4.1. The principle also holds for structural equation models, but the details are a bit more involved.

[20] Okay, the distribution must have a finite variance or the CLT does not apply. For example, the sample mean of a standard Cauchy is also standard Cauchy, and definitely not normal. However, distributions like this are basically mathematical curiosities. All actual measurements are bounded. This means that in the real world, expected values of all orders exist, period.

### 5.5.1 Theoretical answers

**Anderson and Amemiya** In a 1988 article in the *Annals of Statistics*[21], Anderson and Amemiya [4] address a factor analysis model in which the observable variables are not normally distributed. This is only the measurement sub-model of our general structural equation model (1.1) on page 137, but it still covers a lot of territory. For example, regression with latent explanatory variables and observable response variables can be expressed as a factor analysis model. See Chapter 3.

Anderson and Amemiya's main message is on page 759: "...the asymptotic standard errors of the factor loading estimators computed by standard computer packages are valid for virtually any type of nonnormal factor analysis." Factor loadings are the coefficients linking the observed variables to the latent variables. To be honest, I cannot see how their argument does not apply to all the parameters in the model, but I may be missing something.

**Satorra and Bentler** The previously cited 1990 paper by Satorra and Bentler [54] offers a more nuanced picture, providing specific conditions under which robustness should hold. Consider their expression for the asymptotic covariance matrix of $\widehat{\boldsymbol{\theta}}_n$, given in (5.6) on page 437.

$$acov(\widehat{\boldsymbol{\theta}}_n) = \frac{1}{n}\mathbf{J}^{-1}\boldsymbol{\Delta}^\top\mathbf{HLH}\boldsymbol{\Delta}\mathbf{J}^{-1}.$$

The key is the "meat" of the sandwich, the matrix $\mathbf{L} = n \cdot acov(vech(\widehat{\boldsymbol{\Sigma}}))$. For readers who are interested in the primary sources, Satorra and Bentler denote the matrix $\mathbf{L}$ by $\Gamma$, a symbol that is used by most other articles in the research literature.

The asymptotic covariance matrix will be estimated using the sandwich formula above, but how should the component matrices be estimated? The matrices $\mathbf{H}$, $\boldsymbol{\Delta}$ and $\mathbf{J}$ are all functions of $\boldsymbol{\theta}$, so they are estimated by evaluating them at the normal theory MLE $\widehat{\boldsymbol{\theta}}_n$, which by Theorem 5.1 is consistent regardless of the distribution. Only the matrix $\mathbf{L}$ depends on the third and fourth-order moments of the data. If the data are normal, those higher moments are a function of $\boldsymbol{\Sigma}$, but if they are non-normal, this will not be the case in general. So for a nice robust estimator, one uses a straightforward method of moments estimate of $\mathbf{L}$.

But does it matter? Satorra and Bentler approach this question in a very straightforward way. They say okay, what if we used the "wrong" estimate of $\mathbf{L}$? In particular, what if we were to estimate $\mathbf{L}$ using a function of $\widehat{\boldsymbol{\Sigma}}_n$, one that converges to a version of $\mathbf{L}$ with third and fourth-order moments that are correct for the multivariate normal distribution? Denote this normal-theory target by $\mathbf{L}^*$. If the limiting results for a test or estimator are the same when $\mathbf{L}^*$ is substituted for $\mathbf{L}$, Satorra and Bentler say that it's *asymptotically robust* with respect to normality.

[21] The *Annals of Statistics* is undisputedly the top journal in the field. Suppose that the God of the Hebrew and Christian Old Testament thought of ten more commandments. He would try to publish them in the *Annals*. Give how influential His work has been up till now, there is a reasonable chance He would be successful, if the reviewers did not detect any technical errors.

It should be clear that when one compares the two matrices

$$\mathbf{J}(\boldsymbol{\theta})^{-1}\boldsymbol{\Delta}(\boldsymbol{\theta})^{\top}\mathbf{H}(\boldsymbol{\theta})\,\mathbf{L}\,\mathbf{H}(\boldsymbol{\theta})\boldsymbol{\Delta}(\boldsymbol{\theta})\mathbf{J}(\boldsymbol{\theta})^{-1} \quad \text{and} \quad \mathbf{J}(\boldsymbol{\theta})^{-1}\boldsymbol{\Delta}(\boldsymbol{\theta})^{\top}\mathbf{H}(\boldsymbol{\theta})\,\mathbf{L}^{*}\,\mathbf{H}(\boldsymbol{\theta})\boldsymbol{\Delta}(\boldsymbol{\theta})\mathbf{J}(\boldsymbol{\theta})^{-1},$$

for a particular model, some of the corresponding elements might be the same, while others might be different. That is, normal theory standard errors might be robust for some parameters, but not others.

**Satorra and Bentler's Corollary 3.1** Their main result is Corollary 3.1 (p. 242) in [54]. There is a substantial payoff to giving the details. I will mostly use Satorra and Bentler's notation, but the corollary is stated here for the special case of maximum likelihood, and some of the conclusions in the original corollary are omitted.

Let the vector of observable variables have finite fourth moments, with $z = A\xi$, $cov(z) = \Sigma(\theta)$, and $cov(\xi) = \Phi$. Let $A$ be partitioned as $A = (A_1|\,,\ldots,|A_L)$, and divide $\xi$ into sub-vectors $\xi_1,\ldots,\xi_L$ in such a way that the products $A_j\xi_j$ can be formed. Suppose that the $\xi_j$ are independent (not just uncorrelated), and that each $\xi_j$ is either multivariate normal, or has covariance matrix $\Phi_{jj}$ that is *unrestricted*, and not functionally related to either $A$ or to $\Phi_{\ell\ell}$ for $\ell \neq j$. Then except for dependence on $\Phi$, the (multivariate normal) asymptotic distribution of the MLE of the matrix $A$ is free of the distribution of $\xi$.

Now, if $\xi$ is multivariate normal, then the distribution of $\widehat{\theta}$ depends only on $A$ and $\Phi$, and not on any higher moments of $\xi$. Corollary 3.1 says that under the stated conditions, this is also true of the MLE of $A$, regardless of the distribution of $\xi$. In that case, normal theory tests and confidence intervals for elements of $A$ will be valid for large samples. As Satorra and Bentler [54] observe,"...when the corollary applies, to perform statistical inference, we can simply use the normal theory estimate of $\Gamma$, $G^*$, instead of $G$." p. 242. Again, their $\Gamma$ is our $\mathbf{L}$.

At first glance, it may not be so obvious how to fit our general model into the Satorra-Bentler framework. For convenient reference, here are the equations of our centred surrogate model (1.5).

$$\begin{aligned}\mathbf{y} &= \boldsymbol{\beta}\mathbf{y} + \boldsymbol{\Gamma}\mathbf{x} + \boldsymbol{\epsilon} \\ \mathbf{F} &= \begin{pmatrix}\mathbf{x}\\ \hline \mathbf{y}\end{pmatrix} \\ \mathbf{d} &= \boldsymbol{\Lambda}\mathbf{F} + \mathbf{e}\end{aligned}$$

Satorra and Bentler's vector $\xi$ is latent, and $z = A\xi$ looks like the measurement (factor analysis) component of our model, except there is no error term. Actually there is, because $\xi$ includes error terms as well as latent exogenous variables. Write $z = \mathbf{d}$ and $\xi = \begin{pmatrix}\mathbf{F}\\ \hline \mathbf{e}\end{pmatrix}$. Then using the fact that partitioned matrices obey the usual rules of matrix multiplication,

$$z = A\xi = (\boldsymbol{\Lambda}|\mathbf{I}_k)\begin{pmatrix}\mathbf{F}\\ \hline \mathbf{e}\end{pmatrix} = \boldsymbol{\Lambda}\mathbf{F} + \mathbf{e}. \tag{5.7}$$

The latent variable part of the model is trickier, because $z = A\xi$ appears to make no provision for latent variables affecting other latent variables. However, $\xi$ is composed only of latent *exogenous* variables and error terms. Solving $\mathbf{y} = \boldsymbol{\beta}\mathbf{y} + \boldsymbol{\Gamma}\mathbf{x} + \boldsymbol{\epsilon}$ for $\mathbf{y}$ yields $\mathbf{y} = (\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\Gamma}\mathbf{x} + (\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\epsilon}$. Sub-divide the factor loadings in $\boldsymbol{\Lambda}$ into the ones that link $\mathbf{d}$ to $\mathbf{x}$, and the ones that link $\mathbf{d}$ to $\mathbf{y}$. The result is

$$\boldsymbol{\Lambda}\mathbf{F} = (\boldsymbol{\Lambda}_1|\boldsymbol{\Lambda}_2)\left(\frac{\mathbf{x}}{\mathbf{y}}\right) = \boldsymbol{\Lambda}_1\mathbf{x} + \boldsymbol{\Lambda}_2\mathbf{y}.$$

Then,

$$\begin{aligned}
\mathbf{d} &= \boldsymbol{\Lambda}\mathbf{F} + \mathbf{e} \\
&= \boldsymbol{\Lambda}_1\mathbf{x} + \boldsymbol{\Lambda}_2\mathbf{y} + \mathbf{e} \\
&= \boldsymbol{\Lambda}_1\mathbf{x} + \boldsymbol{\Lambda}_2\left((\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\Gamma}\mathbf{x} + (\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\epsilon}\right) + \mathbf{e} \\
&= \boldsymbol{\Lambda}_1\mathbf{x} + \boldsymbol{\Lambda}_2(\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\Gamma}\mathbf{x} + \boldsymbol{\Lambda}_2(\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\epsilon} + \mathbf{e} \\
&= \left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2(\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\Gamma}\right)\mathbf{x} + \boldsymbol{\Lambda}_2(\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\epsilon} + \mathbf{e} \\
&= \left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2(\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\Gamma} \,|\, \boldsymbol{\Lambda}_2(\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\epsilon} \,|\, \mathbf{I}_k\right)\left(\frac{\mathbf{x}}{\frac{\boldsymbol{\epsilon}}{\mathbf{e}}}\right). \qquad (5.8)
\end{aligned}$$

It's a different $\xi$ vector and $A$ matrix than in (5.7), but it works. We have $z = A\xi$, with

$$\begin{aligned}
A &= (A_1|A_2|A_3) = \left(\boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2(\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\Gamma} \,|\, \boldsymbol{\Lambda}_2(\mathbf{I}_q - \boldsymbol{\beta})^{-1}\boldsymbol{\epsilon} \,|\, \mathbf{I}_k\right) \text{ and} \\
\xi &= \left(\frac{\frac{\xi_1}{\xi_2}}{\xi_3}\right) = \left(\frac{\frac{\mathbf{x}}{\boldsymbol{\epsilon}}}{\mathbf{e}}\right).
\end{aligned}$$

A notable feature of (5.8) is that all the parameters in $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ belong to the matrix $A$, and are covered by the robustness result of Corollary 3.1. These are exactly the parameters that would appear on straight arrows in a path diagram. They will be called *straight arrow parameters* in the following useful principle.

**The Satorra-Bentler Principle** Assume the centered structural equation model (1.5), further restricted so that the parameters are identifiable at the true parameter values. Let the exogenous vectors $\mathbf{x}$, $\boldsymbol{\epsilon}$ and $\mathbf{e}$ be independent, and let each of these vectors either (a) be multivariate normal, or (b) have a covariance matrix that is unrestricted, functionally unrelated to the covariance matrices of the other exogenous vectors, and functionally unrelated to any of the straight arrow parameters. Then the normal theory estimated asymptotic covariance matrices of the straight arrow parameters are robust with respect to the assumption of multivariate normality.

Note that the Satorra-Bentler principle is carefully limited. It gives conditions for robustness of the asymptotic covariance matrix of the estimated straight-arrow parameters — that is, for the elements of $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\Gamma}}$ and $\widehat{\boldsymbol{\Lambda}}$. What happens with the standard errors of $\widehat{\boldsymbol{\Phi}}$, $\widehat{\boldsymbol{\epsilon}}$

and $\widehat{\mathbf{\Omega}}$ is unspecified. The expectation is that they will usually be too small. As we shall see, this is often borne out in simulations. However, Satorra and Bentler do *not* prove lack of robustness for variance and covariance parameters. It's just that their proof of robustness was successful with these parameters excluded.

According to the Satorra-Bentler principle, robustness does *not* always hold for straight-arrow parameters. If $\mathbf{x}$, $\boldsymbol{\epsilon}$ and $\mathbf{e}$ are not multivariate normal, then their covariance matrices must be unrestricted, and not functions of one another or of any straight arrow parameters. It is okay for straight arrow parameters to be functions of one another.

Satorra and Bentler's main message is very much like Anderson and Amemiya [4], except for the exceptions just noted.

### 5.5.2 Simulations

Satorra and Bentler [54] illustrate their theory with a simulation in which, using the default normal theory methods, the standard errors for factor loadings are okay, but the standard errors for error variances are too large. This supports the Satorra-Bentler principle. A fair number of other published (and unpublished) simulation studies have examined the performance of normal theory inference for structural equation models when the data are not normally distributed. The consensus view is expressed in a review by Finney and DiStefano [26], who say "Whereas parameter estimates are unaffected by non-normality, their associated significance tests are incorrect if ML estimation is applied to non-normal data. Specifically, the ML-based standard errors underestimate the true variation of the parameter estimates." (p. 274). Their conclusion is the same for the chi-squared test of model fit: "$\ldots \chi^2$ is inflated under conditions of moderate non-normality $\ldots$" (p. 273). This view is shared by Rosseel [48], the creator of lavaan, who writes (p. 27)

> An alternative strategy is to use maximum likelihood (ML) for estimating the model parameters, even if the data are known to be non-normal. In this case, the parameter estimates are still consistent (if the model is identified and correctly specified), but the standard errors tend to be too small (as much as 25-50%), meaning that we may reject the null hypothesis (that a parameter is zero) too often. In addition, the model $\chi^2$ test statistic tends to be too large, meaning that we may reject the model too often.

This blanket conclusion contradicts the theoretical work of both Anderson and Amemiya [4] and the theoretical work of Satorra and Bentler [54] – as well as their simulation study, which was admittedly small-scale. This level of disagreement is uncommon in the field of statistics, and it needs to be resolved.

My own reading of the published simulation studies cited in Finney and DiStefano's article [26] is that in general, they illustrate poor performance for normal theory methods at least part of the time, and better performance for some alternative. However, bad performance of normal theory methods does not always hold, and the details are complicated. Not only do the models, sample sizes and types of non-normality vary greatly, but also the criteria for good or poor performance can be quite different from study to study.

Rather than going into all these details, I will report some new simulation studies[22]. My simulation studies are not quite comprehensive (for example, they do not include the same range of sample sizes for all distributions and all models), but I believe they illustrate what is going on. The discussion, accompanied by simulations, will be divided into three sections: Standard Errors, Tests of fit, and Tests of general hypotheses.

## 5.6 Standard Errors

The battle lines are drawn. There are three credible and partly competing versions of what should happen when the data are non-normal, and standard errors are produced according to normal theory methods. As a reminder to students with other things on their minds, standard errors are estimated standard deviations. They are the denominators of $z$-tests for the parameters, and they determine the width of confidence intervals. Underestimating them will cause tests to reject a true null hypothesis too often, and lead to confidence intervals that are too narrow, indicating more certainty than the data really warrant. Accurate standard errors are very important.

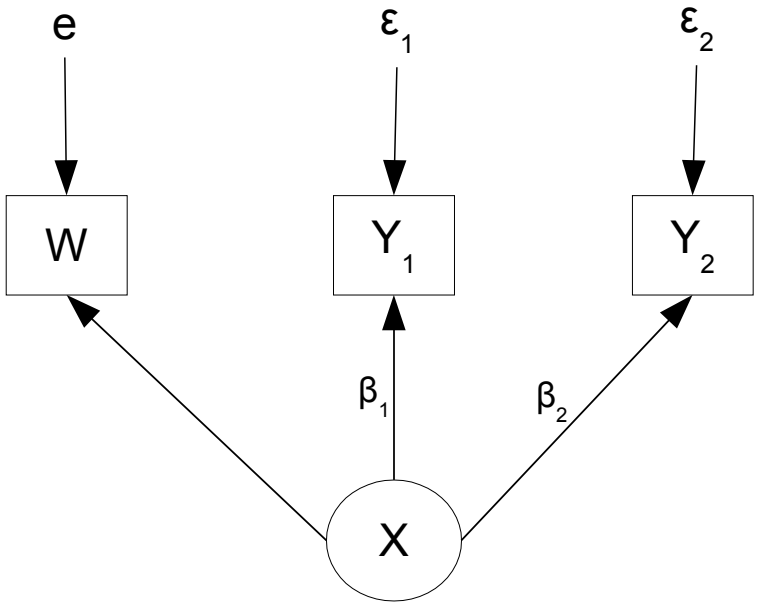
The theoretical work of Anderson and Amemiya [4] says that the standard errors for factor loadings should be okay, and does not make a clear prediction about the other parameters. With qualifications, Satorra and Bentler [54] agree with Anderson and Amemiya about the factor loadings. The Satorra-Bentler principle further implies that standard errors of the coefficients on the straight arrows in the latent variable model will be okay. Their work leaves open the possibility that standard errors for the other parameters (model variances and covariances) will *not* be okay. The consensus reading of a set of empirical simulations studies [26] is that none of it is okay, and all the standard errors will be inflated.

This section will describe a set of simulations (and a few calculations) designed to find out who is right. The simulations will also assess the performance of robust standard errors based on the sandwich estimators of Huber [32] (see Expression 5.5), Satorra and Bentler [54] (Expression 5.6), and the bootstrap. As described earlier, with complete data and the default lavaan settings, the Huber and Satorra-Bentler estimates of $acov(\widehat{\boldsymbol{\theta}}_n)$ are equal. Therefore, though the simulations all use `se = 'robust.huber.white'`, the resulting confidence intervals are also Sattora-Bentler, and identical to what is produced by `se = 'robust.sem'`. They will be described as "Sandwich" confidence intervals. When the normal theory standard errors are *not* robust, the sandwich and bootstrap standard errors are the main alternatives, and it's important to see how well they work. Maybe one is consistently better than the other.

[22] It's quite easy to do simulation studies with R and lavaan. In the bad old days, researchers were writing their own Fortran code to calculate the MLEs and invert the Fisher information matrix. They had the commercial software LISREL, but they couldn't put it in a loop.

### 5.6.1 Extra Response Variable Regression Model

The first simulation will be based on a centered version of Example 0.11.1 from Chapter 0. The path diagram of Figure 17 is reproduced below for convenience.




The model equations are, independently for $i = 1, \ldots, n$.

$$\begin{aligned} w_i &= x_i + e_i \\ y_{i,1} &= \beta_1 x_i + \epsilon_{i,1} \\ y_{i,2} &= \beta_2 x_i + \epsilon_{i,2} \end{aligned} \tag{5.9}$$

where $e_i$, $\epsilon_{i,1}$ and $\epsilon_{i,2}$ are all independent, $Var(x_i) = \phi$, $Var(\epsilon_{i,1}) = \psi_1$, $Var(\epsilon_{i,2}) = \psi_2$, $Var(e_i) = \omega$, and all expected values are zero.

The regression parameters $\beta_1$ and $\beta_2$ are links between latent and observed variables, so it is also correct to call them factor loadings. The theoretical work says their standard errors should be okay, while a summary of the empirical work says their standard errors should be too small. As for the other model parameters, $\phi$, $\psi_1$ and $\psi_2$, the empirical summary [26] says their standard errors should be too small. This is also possible according to the theoretical work of Satorra and Bentler [54]. Specifically, the Satorra-Bentler principle does not apply because these are not straight-arrow parameters.

**Confidence intervals** When is a standard error "too" small? A 95% confidence interval is the parameter estimate plus or minus 1.96 times the standard error, so a standard error that is too small will produce confidence intervals that are too narrow, and fail to include the true parameter value too often. We'll judge a standard error too small if its associated 95% confidence interval captures the true parameter value in significantly[23] less than 95%

[23] In a simulation study, the true parameter values are known, because we just put them in the code. Confidence intervals are based on randomly generated data, and whether one of them happens to incude

percent of the simulated data sets. Confidence intervals and hypothesis tests are one to one, so the conclusions will apply to $z$-tests as well. Note that the normal theory, sandwich and bootstrap confidence intervals are centered on the same MLE, so if one performs better in this way, it means that its standard error is better.

**One simulated data set from Model 5.9** Here's an R session, with a realistic sample size of $n = 200$. The latent variable $x$ and the error terms have exponential distributions, which are right skewed (skewness=2, compared to zero for the normal) and heavy tailed (excess kurtosis=6, compared to zero for the normal). We'll say that the *base distribution* for the simulation is exponential. The observable variables are linear combinations of exponential random variables. Their distribution doesn't have a good name, but it's right skewed and heavy tailed. First setting the true parameter values,

```
> rm(list=ls()); options(scipen=999)
> # Parameters: Make the 3 reliabilities equal.
> beta1 = 0.5; beta2 = 0.7; phi = 4; omega = 1; psi1 = 0.25; psi2 = 0.49
```

The parameter values are chosen so that the reliabilities of the three observed variables (as measures of the latent variable $x$) are equal. Each reliability is the proportion of variance in the observed variable that come from the latent variable. The reliability of $w$ is $\frac{\phi}{\phi+\omega}$, the reliability of $y_1$ is $\frac{\beta_1^2\phi}{\beta_1^2\phi+\psi_1}$, and the reliability of $y_2$ is $\frac{\beta_2^2\phi}{\beta_2^2\phi+\psi_2}$. What makes these quantities intersting for our purposes is that the reliability of $w$ is made up entirely of variances, so the Satorra-Bentler principle does not predict robustness. However, the reliabilities of $y_1$ and $y_2$ include factor loadings as well; what should happen to their standard errors is unclear.

```
> # Calculate true reliabilities
> rel1 = phi/(phi+omega); rel2 = beta1^2*phi/(beta1^2*phi+psi1)
> rel3 = beta2^2*phi/(beta2^2*phi+psi2)
> namz = c("beta1", "beta2", "phi", "omega", "psi1", "psi2", "rel1", "rel2", "rel3")
> truth = c(beta1, beta2, phi, omega, psi1, psi2, rel1, rel2, rel3)
> names(truth)=namz; truth
beta1 beta2   phi omega  psi1  psi2  rel1  rel2  rel3
 0.50  0.70  4.00  1.00  0.25  0.49  0.80  0.80  0.80
```

Now generate the data, multiplying standard exponentials by constants to obtain exponentials with the desired variance.

```
> n = 200; set.seed(9999)
> # Generate exogenous variables
> x = sqrt(phi)*rexp(n); e = sqrt(omega)*rexp(n)
> epsilon1 = sqrt(psi1)*rexp(n); epsilon2 = sqrt(psi2)*rexp(n)
> # Model equations
```

the true parameter is like a coin toss. Of course we can and should apply hypothesis testing to this. How about a nice $z$-test? The "sample size" of the test is the number of simulations.

```
> w = x + e
> y1 = beta1*x + epsilon1
> y2 = beta2*x + epsilon2
> # Put data in a data frame
> simdat = data.frame(w,y1,y2)
```

Next, define and fit the model. The default is maximum likelihood estimation with classical normal theory standard errors.

```
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
> mod  = 'y1 ~ beta1*x # Latent variable model
+         y2 ~ beta2*x
+         x =~ 1.0*w  # Measurement model
+         # Variances (covariances would go here too)
+         x~~phi*x      # Var(x) = phi
+         w ~~ omega*w  # Var(e) = omega
+         y1 ~~ psi1*y1 # Var(epsilon1) = psi1
+         y2 ~~ psi2*y2 # Var(epsilon2) = psi2
+         # Reliabilities
+         reliab1 := phi/(phi+omega)
+         reliab2 := beta1^2*phi/(beta1^2*phi+psi1)
+         reliab3 := beta2^2*phi/(beta2^2*phi+psi2)
+        '
> fit1 = lavaan(mod, data = simdat)
```

Instead of looking at `summary`, it will be more convenient to use the `parameterEstimates` function.

```
> p1 = parameterEstimates(fit1); p1
       lhs op                             rhs   label   est    se      z pvalue ci.lower ci.upper
1       y1  ~                               x   beta1 0.496 0.027 18.489      0    0.443    0.548
2       y2  ~                               x   beta2 0.660 0.037 17.986      0    0.588    0.732
3        x =~                               w         1.000 0.000     NA     NA    1.000    1.000
4        x ~~                               x     phi 4.115 0.528  7.801      0    3.081    5.149
5        w ~~                               w   omega 1.152 0.165  6.965      0    0.828    1.477
6       y1 ~~                              y1    psi1 0.194 0.035  5.547      0    0.126    0.263
7       y2 ~~                              y2    psi2 0.428 0.067  6.370      0    0.296    0.559
8  reliab1 :=                 phi/(phi+omega) reliab1 0.781 0.035 22.044      0    0.712    0.851
9  reliab2 := beta1^2*phi/(beta1^2*phi+psi1) reliab2 0.839 0.032 26.231      0    0.776    0.901
10 reliab3 := beta2^2*phi/(beta2^2*phi+psi2) reliab3 0.807 0.034 23.900      0    0.741    0.874
> is.data.frame(p1)
[1] TRUE
```

As you can see, `parameterEstimates` produces a data frame. It has the estimates and standard errors, but what we want are the lower and upper 95% confidence limits in the last two columns – except for the limits in the third row. These correspond to the factor loading of $w$, which was set to one. It's easy to extract the numbers of interest.

```
> ci1 = p1[-3,9:10] # Upper and lower confidence limits
> cbind(namz,ci1,truth)
    namz  ci.lower  ci.upper truth
1  beta1 0.4429876 0.5480442  0.50
2  beta2 0.5883902 0.7323060  0.70
4    phi 3.0810623 5.1488579  4.00
5  omega 0.8280601 1.4766564  1.00
6   psi1 0.1257016 0.2630545  0.25
7   psi2 0.2961988 0.5594830  0.49
8   rel1 0.7117645 0.8506853  0.80
9   rel2 0.7759922 0.9013214  0.80
10  rel3 0.7412517 0.8736893  0.80
```

Displaying the confidence intervals alongside the true values, we carefully check and find that every confidence interval contains the true value – this time. Automating the process,

```
> hit1 = as.numeric(ci1[,1] < truth & truth < ci1[,2]) # Binary for in ci
> hit1
[1] 1 1 1 1 1 1 1 1 1
```

Now you see how the simulation will work. Just put the simulation and model fitting in a loop, and save `hit1` every time. For comparison, do the same thing with sandwich and bootstrap standard errors.

```
> fit2 = lavaan(mod, data = simdat, se='robust.huber.white')
> fit3 = lavaan(mod, data = simdat, se='bootstrap')
```

If you're doing a simulation study of what happens when a model is mis-specified, it's always advisable to run a version in which all the assumptions are satisfied. In our case, this just means replacing `rexp` with `rnorm`. The purpose is just to verify that the code is correct and the sample size is large enough.

### Normal base distribution

Table 5.2 shows the results for normal data. The top panel of the table shows empirical coverage rates. These are proportions of the 1,000 simulated data sets for which the 95% confidence interval contained the true parameter value. The bottom panel of the table shows corresponding $z$-tests of the null hypothesis that the true coverage probability is 0.95. Denoting the empirical coverage rate by $\widehat{p}$ and the number of simulations (Monte Carlo sample size) by $m$, the test statistic is

$$z = \frac{\sqrt{m}(\widehat{p} - 0.95)}{\sqrt{0.95(1-0.95)}}.$$

In the bottom panel of Table 5.2, twenty-seven tests are being conducted. If they were independent (which they are not) and all the null hypotheses were true, the probability

Table 5.2: Coverage of 95% confidence intervals for the extra response variable model (5.9), $n = 200$, Normal base distribution, 1,000 simulated data sets

| | | | | | | | Reliability of | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta_1$ | $\beta_2$ | $\phi$ | $\omega$ | $\psi_1$ | $\psi_2$ | $w$ | $y_1$ | $y_2$ |
| Normal Theory | 0.952 | 0.951 | 0.950 | 0.945 | 0.942 | 0.943 | 0.953 | 0.938 | 0.941 |
| Sandwich | 0.950 | 0.951 | 0.945 | 0.941 | 0.941 | 0.935 | 0.945 | 0.934 | 0.935 |
| Bootstrap | 0.951 | 0.953 | 0.948 | 0.939 | 0.936 | 0.936 | 0.950 | 0.939 | 0.945 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | 0.290 | 0.145 | 0.000 | -0.725 | -1.161 | -1.016 | 0.435 | -1.741 | -1.306 |
| Sandwich | 0.000 | 0.145 | -0.725 | -1.306 | -1.306 | -2.176 | -0.725 | -2.322 | -2.176 |
| Bootstrap | 0.145 | 0.435 | -0.290 | -1.596 | -2.031 | -2.031 | 0.000 | -1.596 | -0.725 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

of rejecting at least one null hypothesis incorrectly would be $1 - 0.95^{27} \approx 0.75$. To avoid worry about little deviations from 0.95 that are really due to chance, we'll use a Bonferroni correction for the 27 tests. Instead of 1.96, the critical value will be 3.11. This will hold the probability of at least one false conclusion to under 0.05.

Table 5.2 shows that when the normal assumption is correct, the normal theory standard errors yield confidence intervals with the correct coverage. This is no surprise. The robust sandwich and bootstrap standard errors also perform well. The lowest empirical coverage is 0.934 for the reliability of $y_1$ with a robust standard error. The corresponding $z$ statistic of $-2.322$ does not exceed the Bonferroni critical value. Everything is fine.

### Exponential base distribution

Table 5.3 shows that things are not so fine when the data are skewed and heavy tailed. There is some good news, though. Scanning across the first row of numbers, observe that the empirical coverage for the factor loadings $\beta_1$ and $\beta_2$ is very close to 0.95 for the normal theory method. This is what the Satorra-Bentler principle predicts, and it is consistent with Anderson and Amemiya [4]. In contrast, coverage of the variance parameters and the reliabilities in the top row is substantially below 0.95, with corresponding $z$ statistics all in the double digits. The fact that the reliabilities of $y_1$ and $y_2$ involve factor loadings as well as variances did not save them.

For all the parameters, coverage for the sandwich and bootstrap confidence intervals was quite similar. While the the sandwich and bootstrap intervals performed well for the factor loadings and did better than normal theory for the other parameters, coverage was still significantly lower than 0.95 for the variance parameters and reliabilities — with the possible exception of the bootstrap for the reliabilities of $y_1$ and $y_2$.

One possibility is that the sample size of $n = 200$ still isn't big enough. Maybe a larger sample size is needed for asymptotic normality, or maybe a larger sample size is

Table 5.3: Coverage of 95% confidence intervals for the extra response variable model (5.9), $n = 200$, Exponential base distribution, 1,000 simulated data sets

| | | | | | | | Reliability of | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta_1$ | $\beta_2$ | $\phi$ | $\omega$ | $\psi_1$ | $\psi_2$ | $w$ | $y_1$ | $y_2$ |
| Normal Theory | 0.958 | 0.955 | 0.724 | 0.802 | 0.824 | 0.804 | 0.811 | 0.826 | 0.801 |
| Sandwich | 0.957 | 0.947 | 0.886 | 0.904 | 0.905 | 0.904 | 0.913 | 0.923 | 0.912 |
| Bootstrap | 0.948 | 0.947 | 0.891 | 0.906 | 0.902 | 0.909 | 0.927 | 0.940 | 0.937 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | 1.161 | 0.725 | -32.792 | -21.474 | -18.282 | -21.184 | -20.168 | -17.992 | -21.619 |
| Sandwich | 1.016 | -0.435 | -9.286 | -6.674 | -6.529 | -6.674 | -5.369 | -3.918 | -5.514 |
| Bootstrap | -0.290 | -0.435 | -8.561 | -6.384 | -6.965 | -5.949 | -3.337 | -1.451 | -1.886 |

* Bonferroni critical value for 27 two-sided $z$-tests is 3.11.

needed for accurate estimation of the standard deviations of the sampling distributions. Or maybe both. It was easy enough to explore this by running the simulation job 10,000 times (without bootstrapping!), generating 10,000 sets of parameter estimates. This gives very accurate pictures of the sampling distributions. Also, the sample standard deviation of 10,000 parameter estimates gives a very close approximation of the true standard deviation of the sampling distribution.

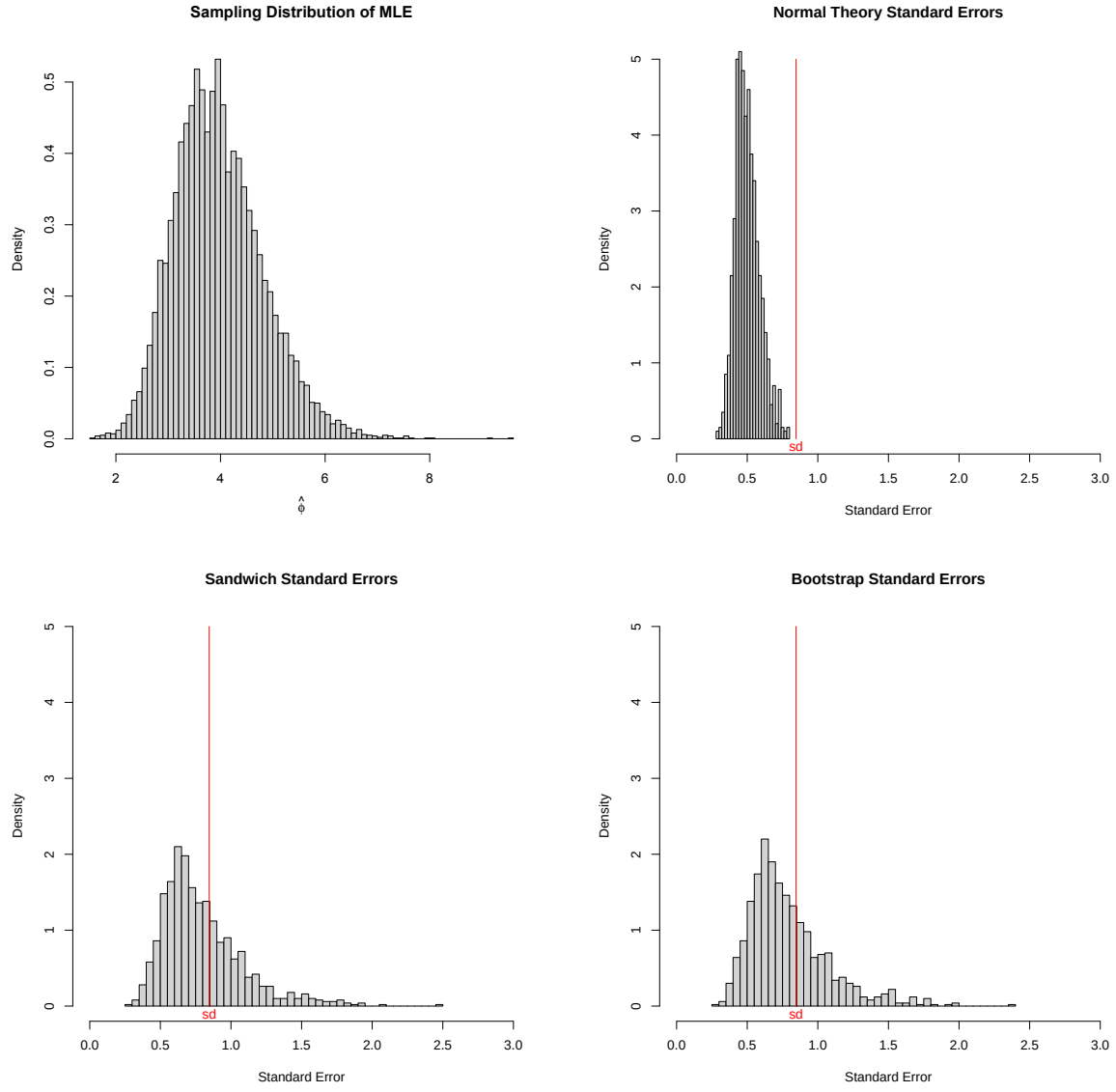
The estimates and standard errors for $Var(x) = \phi$ are a good example. The top left section of Figure 5.2 shows a histogram of the 10,000 randomly generated $\widehat{\phi}$ values. It is clear that asymptotic normality has mostly kicked in, but it's not quite there yet; the distribution has a perceptible right skew and a few high outliers.

The sample standard deviation of the 10,000 simulated $\widehat{\phi}$ values was 0.846. This is guaranteed to be very close to the true standard deviation of the sampling distribution of $\widehat{\phi}$. A separate run generated 1,000 estimates of this quantity by three methods: normal theory, sandwich and bootstrap. The top right and bottom sections of Figure 5.2 shows histograms of the standard errors, with the true quantity being estimated shown in red.

The normal theory standard errors seem to be converging to a value that is smaller than the truth. The distributions of the sandwich and bootstrap standard errors are more or less centered on the truth, but they are dispersed. It seems reasonable that a larger sample size would cause the sandwich and bootstrap standard errors to become more concentrated around the correct value, improving the performance of the sandwich and bootstrap confidence intervals.

The distributions of sandwich and bootstrap standard errors look very similar. In fact, the individual numbers are similar, not just the distributions. Figure 5.3 shows a scatterplot matrix of the normal theory, sandwich and bootstrap standard errors. In the sandwich versus bootstrap plot, the points are tightly clustered around the line $y = x$, with a correlation of 0.998.

Figure 5.2: Sampling distribution and standard errors of $\widehat{\phi}$ for extra response variable model (5.9), exponential base distribution, $n = 200$. The true parameter value is $\phi = 4$. The true standard deviation of the MLE is marked in red.

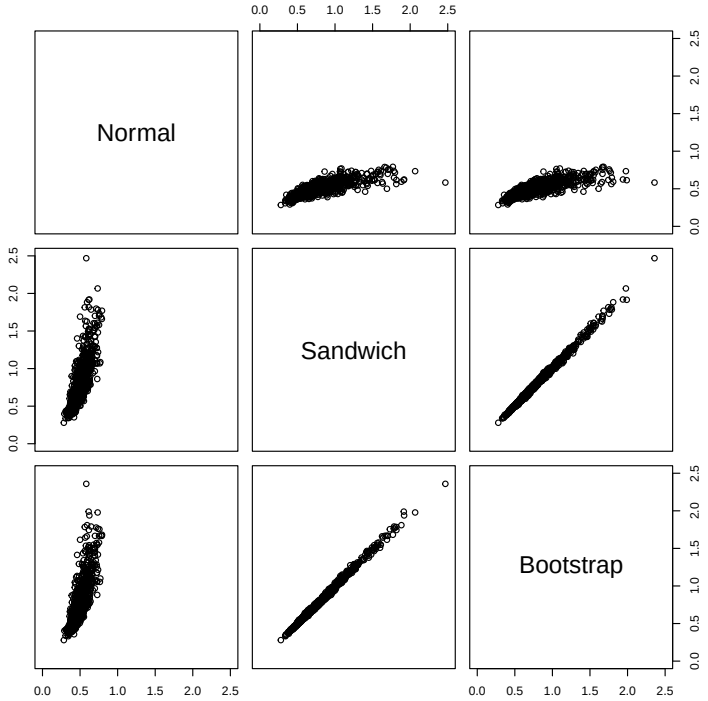



In general, sandwich and bootstrap standard errors tend to be similar for real as well as simulated data sets. However, a small shift in the average standard error for one of the methods can produce a meaningful difference in confidence interval coverage, while maintaining a very high correlation and tight-looking scatterplot.

After playing around with different sample sizes, I found that $n = 1,000$ was needed for the sandwich and bootstrap confidence intervals to perform well for all the parameters with the exponential base distribution. Table 5.4 shows the results.

Everything is clearly okay, except for the normal theory confidence intervals of variance

Figure 5.3: Scatterplot of normal theory, sandwich and bootstrap standard errors for extra response variable model (5.9), exponential base distribution, $n = 200$.




parameters and reliabilities. The performance of the sandwich and bootstrap standard errors is good news, but it still could be a cause for discomfort. While the theory we are using applies as $n \to \infty$ and $n = 1,000$ is nowhere near infinity, still the implication is that sometimes, robust methods such as sandwich and the bootstrap can require sample sizes much larger than the ones that researchers typically employ.

**A really small sample size** In Figure 5.2, the normal theory standard errors are inaccurate, but also more tightly clustered than the sandwich and bootstrap standard errors. This suggests that when they are on target, they might converge to the right answer faster. Table 5.5 shows what happens with the exponential base distribution for $n = 50$, a sample size that would seem radically too small for large-sample multivariate methods. The surprising feature of Table 5.5 is the good performance of the normal theory confidence intervals for $\beta_1$ and $\beta_2$. They were better than the sandwich confidence intervals. We need to keep an eye on this, and see if it happens for other models. Also, the bootstrap appeared to out-perform the sandwich for $\beta_2$ in this case. This might be a random blip, though it was statistically significant.

Table 5.4: Coverage of 95% confidence intervals for the extra response variable model (5.9), $n = 1,000$, Exponential base distribution, 1,000 simulated data sets

| | | | | | | | Reliability of | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta_1$ | $\beta_2$ | $\phi$ | $\omega$ | $\psi_1$ | $\psi_2$ | $w$ | $y_1$ | $y_2$ |
| Normal Theory | 0.958 | 0.950 | 0.764 | 0.799 | 0.824 | 0.822 | 0.797 | 0.813 | 0.805 |
| Sandwich | 0.960 | 0.946 | 0.942 | 0.935 | 0.937 | 0.933 | 0.940 | 0.948 | 0.943 |
| Bootstrap | 0.954 | 0.943 | 0.941 | 0.932 | 0.942 | 0.932 | 0.940 | 0.947 | 0.953 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | 1.161 | 0.000 | -26.988 | -21.909 | -18.282 | -18.572 | -22.200 | -19.878 | -21.039 |
| Sandwich | 1.451 | -0.580 | -1.161 | -2.176 | -1.886 | -2.467 | -1.451 | -0.290 | -1.016 |
| Bootstrap | 0.580 | -1.016 | -1.306 | -2.612 | -1.161 | -2.612 | -1.451 | -0.435 | 0.435 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

### Scaled beta base distribution

It has been suggested that when normal theory structural equation methods fail with non-normal data, the cause is primarily kurtosis (heavy tails) in the data. This idea goes back at least to a 1984 paper by Browne [13], who cites earlier work. Accordingly, it's helpful to try the simulations with a base distribution that is distinctly non-normal, but also lacks the heavy tails of the exponential distribution. We'll use a beta distribution with $\alpha = 3$ and $\beta = 1$, so the density is $f(x) = 3x^2$ for $0 < x < 1$. This distribution is left skewed and quite non-normal, with a skewness of -0.86, and an excess kurtosis of 0.095. The kurtosis is quite similar to the normal. In the simulations, beta random variables are generated with R's `rbeta` function, and then multiplied by constants to yield exogenous variables and error terms with the desired variances.

The observed variables are linear combinations of these scaled beta random variables. Their distribution is nameless, but left skewed and light tailed. Figure 5.4 shows the density of a representative, the sum of two scaled betas with unit variance. To me, this looks like the smoothed version of a questionnaire or response scale with limited range, where people tend to use the upper half of the scale. Student evaluations of university classes are often like this.

Staying with the same extra variable model (5.9) for the simulations, Table 5.6 shows the coverage of 95% confidence intervals for the moderately large sample size of $n = 200$. Note that with this non-normal but light tailed distribution, *the normal theory methods performed very well, even for the variance parameters and reliabilities.* This supports the idea [13] that the problems exposed by the exponential base distribution come from high kurtosis, and not departure from normality per se. We'll seek more evidence for this encouraging possibility.

Before exploring other models, we should look at what happens with the scaled beta base distribution when the sample size is small. Table 5.7 shows the results for $n = 50$.

Table 5.5: Coverage of 95% confidence intervals for the extra response variable model (5.9), $n = 50$, Exponential base distribution, 1,000 simulated data sets

| | | | | | | | Reliability of | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta_1$ | $\beta_2$ | $\phi$ | $\omega$ | $\psi_1$ | $\psi_2$ | $w$ | $y_1$ | $y_2$ |
| Normal Theory | 0.951 | 0.943 | 0.764 | 0.804 | 0.801 | 0.791 | 0.813 | 0.793 | 0.811 |
| Sandwich | 0.931 | 0.916 | 0.826 | 0.841 | 0.833 | 0.838 | 0.882 | 0.848 | 0.857 |
| Bootstrap | 0.940 | 0.937 | 0.831 | 0.831 | 0.826 | 0.831 | 0.944 | 0.919 | 0.927 |
| | | | | | $z$ Statistics* | | | | |
| Normal Theory | 0.145 | -1.016 | -26.988 | -21.184 | -21.619 | -23.070 | -19.878 | -22.780 | -20.168 |
| Sandwich | -2.757 | -4.933 | -17.992 | -15.815 | -16.976 | -16.251 | -9.866 | -14.800 | -13.494 |
| Bootstrap | -1.451 | -1.886 | -17.266 | -17.266 | -17.992 | -17.266 | -0.871 | -4.498 | -3.337 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

The main finding is that the robustness of the normal theory standard errors is evident for the factor loadings $\beta_1$ and $\beta_2$, but mostly not for the variance parameters and reliabilities. The bootstrap and sandwich standard errors have no particular advantage. In particular, the apparent differences in coverage for $\beta_1$ and $\beta_2$ are not significantly different for the three types of standard error.

### 5.6.2 Double Measurement Regression Model

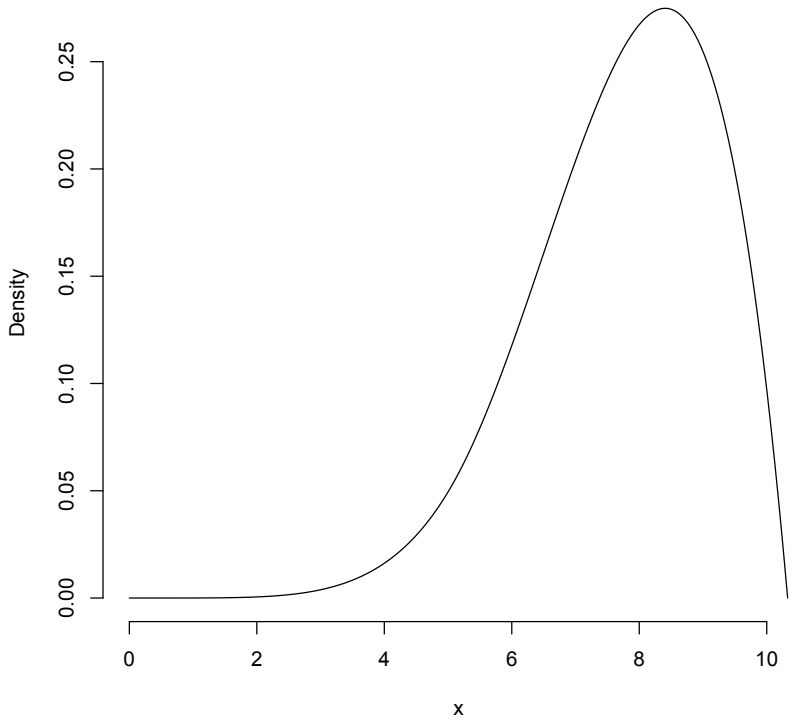
The next set of simulations will be based on a simple double measurement model of the kind described in Chapter 0, Section 0.10.3. Figure 5.5 is a scalar version of Figure 15 on page 85, accompanied by the model equations.
For the simulations, the parameter values will be

| $\beta$ | $\phi$ | $\psi$ | $\omega_{11}$ | $\omega_{22}$ | $\omega_{33}$ | $\omega_{44}$ | $\omega_{13}$ | $\omega_{24}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 1 | 2 | 2 | 2 | 2 | 1 | 0 |

Here is the code used to simulate a single data set. It was put in a loop to generate sets of 1,000 results, as in the earlier simulations.

```
# Simulate one data set from the double measurement regression model
rm(list=ls())
# install.packages("lavaan", dependencies = TRUE) # Only need to do this once
library(lavaan)

# Set true parameter values
# Covariance between measurement errors will come from e = u + delta,
# with Var(u) = v and Var(delta)=omega_ij. So, for example.
# Var(e1) = Var(u1+delta1) = v1 + omega13, and
```

Figure 5.4: Density of the sum of two independent scaled betas




```
# Cov(e1,e3) = Cov(u1+delta1,u3+delta1) = Var(delta1) = omega13

beta = 1; phi = 4; psi = 1
# Choose var_j values to make the omega_jj equal
var1 = 1; var2 = 2; var3 = 1; var4 = 2; omega13 = 1; omega24 = 0
omega11 = var1+omega13; omega22 = var2+omega24
omega33 = var3+omega13; omega44 = var4+omega24
k = 1 # Scaling constant, to make the variance of the base distribution equal one

namz = c("beta", "phi", "psi", "omega11", "omega22", "omega33", "omega44",
         "omega13", "omega24")
truth = c(beta, phi, psi, omega11, omega22, omega33, omega44, omega13, omega24)
names(truth)=namz; truth

# Generate a data set
n = 200; set.seed(9999)
x  = sqrt(phi)*k*rnorm(n)   # For a variance of phi
```

Table 5.6: Coverage of 95% confidence intervals for the extra response variable model (5.9), $n = 200$, Scaled beta base distribution, 1,000 simulated data sets

| | | | | | | | Reliability of | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta_1$ | $\beta_2$ | $\phi$ | $\omega$ | $\psi_1$ | $\psi_2$ | $w$ | $y_1$ | $y_2$ |
| Normal Theory | 0.952 | 0.943 | 0.943 | 0.942 | 0.939 | 0.943 | 0.945 | 0.944 | 0.936 |
| Sandwich | 0.944 | 0.936 | 0.940 | 0.939 | 0.940 | 0.939 | 0.946 | 0.948 | 0.935 |
| Bootstrap | 0.954 | 0.940 | 0.944 | 0.935 | 0.938 | 0.938 | 0.946 | 0.952 | 0.936 |
| | | | | | $z$ Statistics* | | | | |
| Normal Theory | 0.290 | -1.016 | -1.016 | -1.161 | -1.596 | -1.016 | -0.725 | -0.871 | -2.031 |
| Sandwich | -0.871 | -2.031 | -1.451 | -1.596 | -1.451 | -1.596 | -0.580 | -0.290 | -2.176 |
| Bootstrap | 0.580 | -1.451 | -0.871 | -2.176 | -1.741 | -1.741 | -0.580 | 0.290 | -2.031 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

```
u1 = sqrt(var1)*k*rnorm(n); u2 = sqrt(var2)*k*rnorm(n)
u3 = sqrt(var3)*k*rnorm(n); u4 = sqrt(var4)*k*rnorm(n)
delta1 = sqrt(omega13)*k*rnorm(n); delta2 = sqrt(omega24)*k*rnorm(n)
e1 = u1+delta1; e2 = u2+delta2; e3 = u3+delta1; e4 = u4+delta2
epsilon = sqrt(psi)*k*rnorm(n)
# Model equations
y = beta*x + epsilon
w1 = x+e1; w2 = x+e2; v1 = y+e3; v2 = y+e4
datta = data.frame(w1,w2,v1,v2)

mod =   '# Latent variable model
         y ~ beta*x
         # Measurement model
         x =~ 1*w1 + 1*w2
         y =~ 1*v1 + 1*v2
         # Variances
         x ~~ phi*x; y ~~ psi*y
         w1 ~~ omega11*w1; w2 ~~ omega22*w2
         v1 ~~ omega33*v1; v2 ~~ omega44*v2
         # Covariances
         w1 ~~ omega13*v1; w2 ~~ omega24*v2
        '
fit1 = lavaan(mod,datta)
summary(fit1)
p1 = parameterEstimates(fit1)
ci1 = p1[c(1,6:13),9:10] # Upper and lower confidence limits
hit1 = as.numeric(ci1[,1] < truth & truth < ci1[,2]) # Binary for in ci
cbind(namz,ci1,truth,hit1)
```

Table 5.7: Coverage of 95% confidence intervals for the extra response variable model (5.9), $n = 50$, Scaled beta base distribution, 1,000 simulated data sets

| | | | | | | | Reliability of | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta_1$ | $\beta_2$ | $\phi$ | $\omega$ | $\psi_1$ | $\psi_2$ | $w$ | $y_1$ | $y_2$ |
| Normal Theory | 0.946 | 0.956 | 0.921 | 0.930 | 0.924 | 0.917 | 0.931 | 0.913 | 0.923 |
| Sandwich | 0.936 | 0.933 | 0.905 | 0.919 | 0.910 | 0.900 | 0.916 | 0.896 | 0.914 |
| Bootstrap | 0.947 | 0.938 | 0.910 | 0.913 | 0.898 | 0.894 | 0.939 | 0.924 | 0.937 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | -0.580 | 0.871 | -4.208 | -2.902 | -3.772 | -4.788 | -2.757 | -5.369 | -3.918 |
| Sandwich | -2.031 | -2.467 | -6.529 | -4.498 | -5.804 | -7.255 | -4.933 | -7.835 | -5.223 |
| Bootstrap | -0.435 | -1.741 | -5.804 | -5.369 | -7.545 | -8.125 | -1.596 | -3.772 | -1.886 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

**Normal base distribution**

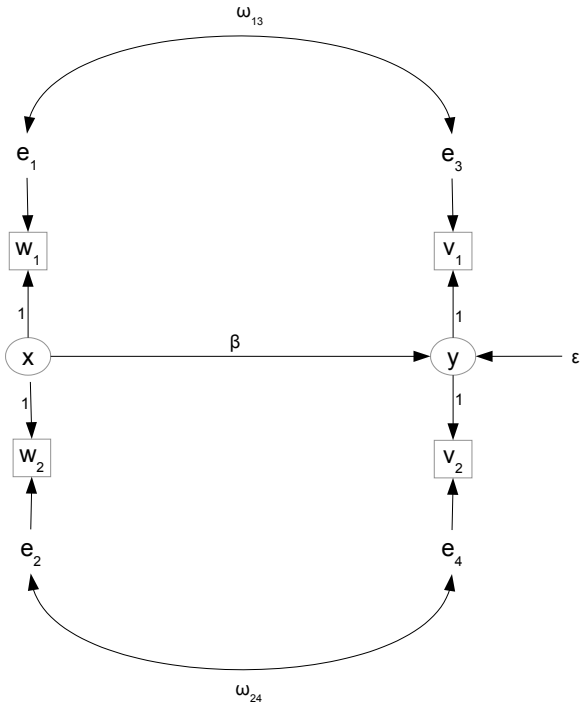
For $n = 200$ with the normal assumption satisfied, all the confidence intervals performed well. The results will not be shown. In my opinion, $n = 50$ is far too small to expect anything good to happen, but the results are given in Table 5.8 anyway. Even with this small sample size, the results are good for $\beta$, the parameter of primary interest. In fact, though there is a systematic tendency to under-coverage for the normal theory confidence intervals, with the Bonferroni correction it only reaches statistical significance for the parameters $\phi = Var(x)$ and $\omega_{13} = Cov(e_1, e_3)$. There is clearly no advantage (and some disadvantage) to using the sandwich or bootstrap methods when the data really are normal. Overall, the normal theory confidence intervals do surprisingly well, considering

Table 5.8: Coverage of 95% confidence intervals for the double measurement regression model (5.5), $n = 50$, Normal base distribution, 1,000 simulated data sets

| | $\beta$ | $\phi$ | $\psi$ | $\omega_{11}$ | $\omega_{22}$ | $\omega_{33}$ | $\omega_{44}$ | $\omega_{13}$ | $\omega_{24}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.955 | 0.921 | 0.943 | 0.938 | 0.930 | 0.932 | 0.932 | 0.928 | 0.940 |
| Sandwich | 0.943 | 0.909 | 0.930 | 0.920 | 0.918 | 0.932 | 0.919 | 0.917 | 0.931 |
| Bootstrap | 0.941 | 0.909 | 0.923 | 0.927 | 0.920 | 0.934 | 0.920 | 0.918 | 0.939 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | 0.725 | -4.208 | -1.016 | -1.741 | -2.902 | -2.612 | -2.612 | -3.192 | -1.451 |
| Sandwich | -1.016 | -5.949 | -2.902 | -4.353 | -4.643 | -2.612 | -4.498 | -4.788 | -2.757 |
| Bootstrap | -1.306 | -5.949 | -3.918 | -3.337 | -4.353 | -2.322 | -4.353 | -4.643 | -1.596 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

Figure 5.5: The Scalar Double Measurement Model




$$
\begin{aligned}
w_1 &= x + e_1 \\
w_2 &= x + e_2 \\
v_1 &= y + e_3 \\
v_2 &= y + e_4, \\
y &= \beta x + \epsilon,
\end{aligned}
$$

where $Var(x) = \phi$, $Var(\epsilon) = \psi$, $Var(e_j) = \omega_{jj}$ for $j = 1, \ldots, 4$, $Cov(e_1, e_3) = \omega_{13}$ and $Cov(e_2, e_4) = \omega_{24}$.

the small sample size.

### Exponential base distribution

For the heavy-tailed exponential distribution, the Satorra-Bentler principle says that the normal theory standard error for $\widehat{\beta}$ should be robust, while Anderson and Amemiya [4] do not make a clear prediction, and the review of simulations by Finney and DiStefano [26] says never to expect robustness. One expects the normal theory standard errors for the other parameters to be too small, leading to under-coverage of the confidence intervals. The sandwich and bootstrap confidence intervals should perform better for these parameters, but it remains to be seen what sample size is required. Table 5.9 shows results for the moderately large sample size of $n = 200$.

The Satorra-Bentler principle is confirmed again. The normal theory confidence interval did well for $\beta$, the only model parameter that is not a variance or covariance. The performance of the normal theory intervals was spotty for the the other parameters, but not really bad. Only for $\phi$ and $\omega_{13}$ was coverage significantly below 0.95 with the Bonferroni correction. Generally, the normal theory method seemed to do better than sandwich and the bootstrap most of the time, though I did not carry out formal tests for all the differences in coverage.

Table 5.9: Coverage of 95% confidence intervals for the double measurement regression model (5.5), $n = 200$, Exponential base distribution, 1,000 simulated data sets

| | $\beta$ | $\phi$ | $\psi$ | $\omega_{11}$ | $\omega_{22}$ | $\omega_{33}$ | $\omega_{44}$ | $\omega_{13}$ | $\omega_{24}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.958 | 0.774 | 0.900 | 0.901 | 0.840 | 0.890 | 0.851 | 0.909 | 0.953 |
| Sandwich | 0.950 | 0.906 | 0.931 | 0.935 | 0.921 | 0.922 | 0.921 | 0.940 | 0.949 |
| Bootstrap | 0.946 | 0.906 | 0.924 | 0.929 | 0.916 | 0.923 | 0.921 | 0.939 | 0.944 |
| | | | | | $z$ Statistics* | | | | |
| Normal Theory | 1.161 | -25.537 | -7.255 | -7.110 | -15.960 | -8.706 | -14.364 | -5.949 | 0.435 |
| Sandwich | 0.000 | -6.384 | -2.757 | -2.176 | -4.208 | -4.063 | -4.208 | -1.451 | -0.145 |
| Bootstrap | -0.580 | -6.384 | -3.772 | -3.047 | -4.933 | -3.918 | -4.208 | -1.596 | -0.871 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

In the simulations using the extra response variable model (5.9), the sandwich and bootstrap required a larger sample to perform adequately for variance and covariance parameters with the exponential base distribution, while the normal theory methods failed. For the present model (double measurement regression), I explored different sample sizes, excluding the bootstrap because it was too time consuming and pretty much guaranteed to be similar to sandwich anyway. $n = 500$ was good enough for all but $\phi = Var(x)$. With $n = 750$, coverage for $\phi$ was 0.925, $z = -3.627$. These simulations were done with different seeds for the random number generator, providing a bit of replication. Replication can be informative in simulation studies, just as in other forms of empirical research.

Coverage for the normal theory confidence intervals was consistently awful, except for $\beta$ (as Satorra and Bentler predict), and also except for $\omega_{24}$. This last parameter is special, because the true value in the simulations happend to be zero. In contrast, $\omega_{13}$ is parallel to $\omega_{24}$ in every way, except that I gave it a non-zero true value. So it's a kind of controlled experiment. The repeated finding of robustness for $\omega_{24}$ is encouraging, because it implies accuracy when the natural null hypothses $\omega_{24} = 0$ is true. It suggests that the Type I error probability would be close to 0.05 for normal theory methods — for example, with a $z-$test. It also teaches the somewhat uncomfortable lesson that robustness could depend on the true parameter values. The full picture might be painted only by a very extensive simulation study, or a more refined theory.

The exploratory simulations were intended to locate a sample size where the sandwich and bootstrap confidence intervals would perform acceptably. With $n = 1,000$ (the same number required for the exponential distribution under the extra response variable model) everything seemed okay, so I produced Table 5.10 — using a fresh seed for the random number generation, of course. Now the coverage sandwich and bootstrap confidence intervals is no longer significantly different from 0.95 with the Bonferroni correction, though the numbers for $\phi$ are flirting with trouble. The normal theory confidence intervals perform well only for $\beta$ and $\omega_{24}$. So again, the robust methods require a large sample size to work well with most variance and covariance parameters. And again, coverage for

Table 5.10: Coverage of 95% confidence intervals for the double measurement regression model (5.5), $n = 1,000$, Exponential base distribution, 1,000 simulated data sets

| | $\beta$ | $\phi$ | $\psi$ | $\omega_{11}$ | $\omega_{22}$ | $\omega_{33}$ | $\omega_{44}$ | $\omega_{13}$ | $\omega_{24}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.957 | 0.752 | 0.904 | 0.879 | 0.831 | 0.892 | 0.845 | 0.912 | 0.952 |
| Sandwich | 0.955 | 0.931 | 0.946 | 0.944 | 0.938 | 0.942 | 0.933 | 0.951 | 0.950 |
| Bootstrap | 0.947 | 0.930 | 0.943 | 0.942 | 0.938 | 0.946 | 0.933 | 0.953 | 0.942 |
| | | | | | $z$ Statistics* | | | | |
| Normal Theory | 1.016 | -28.729 | -6.674 | -10.302 | -17.266 | -8.416 | -15.235 | -5.514 | 0.290 |
| Sandwich | 0.725 | -2.757 | -0.580 | -0.871 | -1.741 | -1.161 | -2.467 | 0.145 | 0.000 |
| Bootstrap | -0.435 | -2.902 | -1.016 | -1.161 | -1.741 | -0.580 | -2.467 | 0.435 | -1.161 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

$\omega_{24} = 0$ is good, while coverage for $\omega_{13} = 1$ is not.

Table 5.11 goes down to the outrageously small sample size of $n = 50$. Performance

Table 5.11: Coverage of 95% confidence intervals for the double measurement regression model (5.5), $n = 50$, Exponential base distribution, 1,000 simulated data sets

| | $\beta$ | $\phi$ | $\psi$ | $\omega_{11}$ | $\omega_{22}$ | $\omega_{33}$ | $\omega_{44}$ | $\omega_{13}$ | $\omega_{24}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.933 | 0.734 | 0.895 | 0.885 | 0.829 | 0.881 | 0.843 | 0.895 | 0.942 |
| Sandwich | 0.902 | 0.817 | 0.910 | 0.889 | 0.862 | 0.890 | 0.883 | 0.903 | 0.940 |
| Bootstrap | 0.930 | 0.830 | 0.883 | 0.887 | 0.867 | 0.893 | 0.883 | 0.906 | 0.929 |
| | | | | | $z$ Statistics* | | | | |
| Normal Theory | -2.467 | -31.341 | -7.980 | -9.431 | -17.557 | -10.012 | -15.525 | -7.980 | -1.161 |
| Sandwich | -6.965 | -19.298 | -5.804 | -8.851 | -12.768 | -8.706 | -9.721 | -6.819 | -1.451 |
| Bootstrap | -2.902 | -17.411 | -9.721 | -9.141 | -12.043 | -8.270 | -9.721 | -6.384 | -3.047 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

of the normal theory intervals was not too bad for $\beta$ or the special parameter $\omega_{24}$; the sandwich and bootstrap intervals were also okay for $\omega_{24}$. Otherwise, substantial under-coverage was the rule, as one would expect for this sample size.

### Scaled beta base distribution

Table 5.12 shows coverage for the light-tailed scaled beta base distribution, with our standard sample size of $n = 200$. Once again, with this distribution the normal theory intervals show only mild and non-significant under-coverage, supporting the idea that when the normal theory methods are not robust, the problem is with heavy tails rather

Table 5.12: Coverage of 95% confidence intervals for the double measurement regression model (5.5), $n = 200$, Scaled beta base distribution, 1,000 simulated data sets

| | $\beta$ | $\phi$ | $\psi$ | $\omega_{11}$ | $\omega_{22}$ | $\omega_{33}$ | $\omega_{44}$ | $\omega_{13}$ | $\omega_{24}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.935 | 0.935 | 0.940 | 0.939 | 0.938 | 0.950 | 0.943 | 0.941 | 0.947 |
| Sandwich | 0.935 | 0.939 | 0.939 | 0.933 | 0.940 | 0.945 | 0.932 | 0.943 | 0.946 |
| Bootstrap | 0.935 | 0.938 | 0.934 | 0.936 | 0.942 | 0.948 | 0.934 | 0.946 | 0.943 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | -2.176 | -2.176 | -1.451 | -1.596 | -1.741 | 0.000 | -1.016 | -1.306 | -0.435 |
| Sandwich | -2.176 | -1.596 | -1.596 | -2.467 | -1.451 | -0.725 | -2.612 | -1.016 | -0.580 |
| Bootstrap | -2.176 | -1.741 | -2.322 | -2.031 | -1.161 | -0.290 | -2.322 | -0.580 | -1.016 |

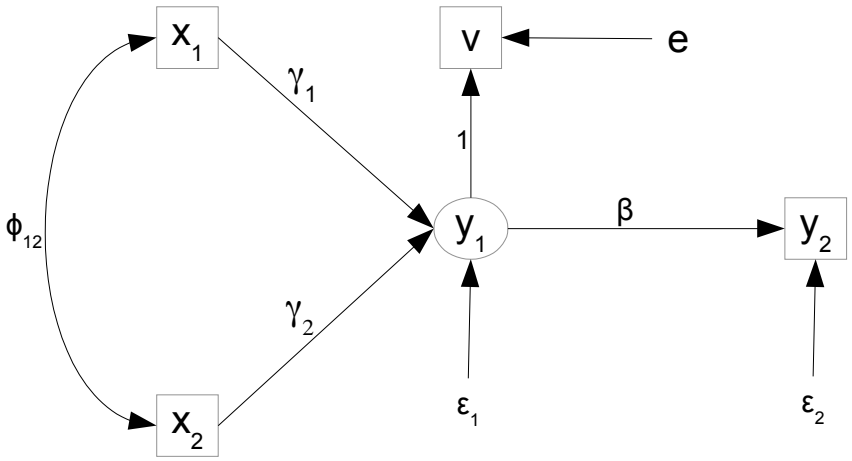
* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

than non-normality per se. The sandwich and bootstrap intervals confer no advantage at this sample size. The numbers are not shown for $n = 50$, but in this case the coverage for $\beta$ and $\omega_{24}$ is acceptable for the normal theory intervals. For most of the other parameters, the normal theory intervals showed a bit less under-coverage than sandwich and the bootstrap, but their coverage was still significantly less than 0.95 in most cases.

### 5.6.3 The "Dip Down" Path Model

For want of a better term, I will call the model of Figure 5.6 the Dip Down Path Model. The observable variables $x_1$ and $x_2$ dip down to influence the sub-surface latent variable

Figure 5.6: The Dip Down Path Model




$y_1$, which re-surfaces to influence the observable variable $y_2$. The latent variable $y_1$ is

measured once, with error. The model equations are

$$\begin{aligned} y_1 &= \gamma_1 x_1 + \gamma_2 x_2 + \epsilon_1 \\ y_2 &= \beta y_1 + \epsilon_2 \\ v &= y_1 + e, \end{aligned} \tag{5.10}$$

where $Var(x_1) = \phi_{11}$, $Var(x_2) = \phi_{22}$, $Cov(x_1, x_2) = \phi_{12}$, $Var(\epsilon_1) = \psi_1$, $Var(\epsilon_2) = \psi_2$, $Var(e) = \omega$, and the error terms are all independent of $x_1$ and $x_2$, and of one another. The variables $x_1$, $x_2$, $v$ and $y_2$ are observable.

For the simulations, the parameter values will be

| $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|
| 0.5 | 0.5 | 0.5 | 2.0 | 2.0 | 1.0 | 2.0 | 3.0 | 1.0 |

Here is a simulation of just one data set from this model, with a very large sample size of $n = 200,000$ and the normal assumption satisfied. Estimates are very close to the truth, as expected. The parameters $\gamma_1$, $\gamma_2$ and $\beta$ were made equal, so a good null hypothesis will be true when we consider hypothesis testing. What makes the null hypothesis good is that all the coefficients involved are on straight arrows, so the Satorra-Bentler principle implies robustness of the normal theory methods. We will get to that later.

```
> # Dip down model in which observable x1 and x2 affect latent y1 (measured once),
which in turn affects observable y2.
>
> rm(list=ls())
>
> # Set true parameter values
> #   Covariance between x1 and x2 will come from x = u + delta,
> #   with Var(u) = v and Var(delta)=phi12. So, for example.
> #   Var(x1) = Var(u1+delta) = v + phi12
> #   Cov(x1,x2) = Cov(u1+delta,u2+delta) = Var(delta) = phi12
> gamma1 = 0.5; gamma2 = 0.5; beta = 0.5 # H0: gamma1=gamma2=beta is true
> v1 = 1; v2 = 1; phi12 = 1
> psi1 = 1; psi2 = 2; omega = 3
> phi11 = v1 + phi12; phi22 = v2 + phi12  # Calculate phi11 and phi22
> k = 1 # Scaling constant, to make the variance of the base distribution equal one
>
> truth = c(gamma1, gamma2, beta, phi11, phi22, psi1, psi2, omega, phi12)
> namz = c('gamma1', 'gamma2', 'beta', 'phi11', 'phi22', 'psi1', 'psi2', 'omega', 'phi12')
> names(truth)=namz; truth
gamma1 gamma2   beta  phi11  phi22   psi1   psi2  omega  phi12
   0.5    0.5    0.5    2.0    2.0    1.0    2.0    3.0    1.0
>
> n = 200000; set.seed(9999)
> delta = sqrt(phi12)*k*rnorm(n); u1 = sqrt(v1)*k*rnorm(n)
> u2 = sqrt(v2)*k*rnorm(n); x1 = u1+delta; x2 = u2+delta
```

```
> e = sqrt(omega)*k*rnorm(n)
> epsilon1 = sqrt(psi1)*k*rnorm(n); epsilon2 = sqrt(psi2)*k*rnorm(n)
> # Model equations
> y1 = gamma1*x1 + gamma2*x2 + epsilon1
> y2 = beta*y1 + epsilon2
> v = y1 + e
> simdat = cbind(x1,x2,v,y2)
>
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
>
> mod =   'y1 ~ gamma1*x1 + gamma2*x2
+          y2 ~ beta*y1
+          y1 =~ 1.0*v    # Measurement model
+          # Variances
+          x1 ~~ phi11*x1 # Var(x1) = phi11
+          x2 ~~ phi22*x2 # Var(x2) = phi22
+          y1 ~~ psi1*y1  # Var(epsilon1) = psi1
+          y2 ~~ psi2*y2  # Var(epsilon2) = psi2
+          v  ~~ omega*v  # Var(e) = omega
+          # Covariance
+          x1 ~~ phi12*x2 # Cov(x1,x2) = phi12
+         '
> fit1 = lavaan(mod,simdat)
> p1 = parameterEstimates(fit1)
> ci1 = p1[-4,9:10] # Upper and lower confidence limits
> hit1 = as.numeric(ci1[,1] < truth & truth < ci1[,2]) # Binary for in ci
> rbind(truth, coef(fit1))
         gamma1    gamma2      beta    phi11    phi22      psi1     psi2    omega     phi12
truth 0.5000000 0.5000000 0.5000000 2.000000 2.000000 1.0000000 2.000000 3.000000 1.0000000
      0.5018487 0.5030646 0.5026666 1.987888 1.987486 0.9777155 1.991012 3.027083 0.9922169
```

As in the earlier simulations, this code was put in a loop for 1,000 simulations using normal, exponential and scaled beta base distributions, with various sample sizes (all a lot less than 200,000). We will watch particularly for the robustness of the confidence intervals for $\gamma_1$, $\gamma_2$ and $\beta$, as predicted by the Satorra-Bentler principle.

### Normal base distribution

Table 5.13 shows coverage for $n = 200$ when the assumption of a normal distribution is satisfied. Technically, the normal theory intervals all perform acceptably in that none of the $z$ statistics exceeds the Bonferroni critical value. Still, the $z$ value for normal theory is uncomfortably close to significance for $\psi_2$, and the sandwich and bootstrap under-coverage is statistically significant. I ran the simulation again with a different random number seed to see if it was a fluke, and there was still trouble for $\psi_2$.

Table 5.13: Coverage of 95% confidence intervals for the "dip down" model (5.10), $n = 200$, Normal base distribution, 1,000 simulated data sets

| | $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.948 | 0.941 | 0.952 | 0.941 | 0.950 | 0.933 | 0.930 | 0.961 | 0.940 |
| Sandwich | 0.948 | 0.933 | 0.954 | 0.935 | 0.940 | 0.931 | 0.924 | 0.960 | 0.938 |
| Bootstrap | 0.956 | 0.946 | 0.954 | 0.935 | 0.945 | 0.942 | 0.917 | 0.947 | 0.936 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | -0.290 | -1.306 | 0.290 | -1.306 | 0.000 | -2.467 | -2.902 | 1.596 | -1.451 |
| Sandwich | -0.290 | -2.467 | 0.580 | -2.176 | -1.451 | -2.757 | -3.772 | 1.451 | -1.741 |
| Bootstrap | 0.871 | -0.580 | 0.580 | -2.176 | -0.725 | -1.161 | -4.788 | -0.435 | -2.031 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

Table 5.14 shows results for $n = 500$. After all, even the normal theory methods are asymptotic. Perhaps the sample size was not big enough for this model. Ah, that's

Table 5.14: Coverage of 95% confidence intervals for the "dip down" model (5.10), $n = 500$, Normal base distribution, 1,000 simulated data sets

| | $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.950 | 0.952 | 0.959 | 0.952 | 0.946 | 0.944 | 0.946 | 0.945 | 0.963 |
| Sandwich | 0.946 | 0.952 | 0.953 | 0.948 | 0.941 | 0.944 | 0.946 | 0.949 | 0.961 |
| Bootstrap | 0.954 | 0.953 | 0.950 | 0.951 | 0.944 | 0.939 | 0.942 | 0.948 | 0.958 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | 0.00 | 0.290 | 1.306 | 0.290 | -0.580 | -0.871 | -0.580 | -0.725 | 1.886 |
| Sandwich | -0.58 | 0.290 | 0.435 | -0.290 | -1.306 | -0.871 | -0.580 | -0.145 | 1.596 |
| Bootstrap | 0.58 | 0.435 | 0.000 | 0.145 | -0.871 | -1.596 | -1.161 | -0.290 | 1.161 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

better. Now everything is definitely okay. This is a reminder that robustness may depend on details of the model, as well as on the true distribution and the sample size.

I tried $n = 50$ just for completeness, even though the results so far are not encouraging about small sample sizes. Table 5.15 shows the results. It's pretty bad. Performance is all right for $\beta$, but that's about it.

In summary, it looks like this model may require a somewhat larger sample size than the others, even when the normal distribution assumption is satisfied.

Table 5.15: Coverage of 95% confidence intervals for the "dip down" model (5.10), $n = 50$, Normal base distribution, 1,000 simulated data sets

| | $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.907 | 0.917 | 0.947 | 0.921 | 0.905 | 0.936 | 0.930 | 0.964 | 0.925 |
| Sandwich | 0.908 | 0.923 | 0.944 | 0.904 | 0.884 | 0.922 | 0.918 | 0.959 | 0.911 |
| Bootstrap | 0.938 | 0.941 | 0.953 | 0.901 | 0.881 | 0.941 | 0.909 | 0.930 | 0.910 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | -6.239 | -4.788 | -0.435 | -4.208 | -6.529 | -2.031 | -2.902 | 2.031 | -3.627 |
| Sandwich | -6.094 | -3.918 | -0.871 | -6.674 | -9.576 | -4.063 | -4.643 | 1.306 | -5.659 |
| Bootstrap | -1.741 | -1.306 | 0.435 | -7.110 | -10.012 | -1.306 | -5.949 | -2.902 | -5.804 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

### Exponential base distribution

Table 5.16 shows confidence interval coverage for $n = 200$ and the heavy-tailed exponential base distribution. This is pretty much what we have grown to expect. Normal theory

Table 5.16: Coverage of 95% confidence intervals for the "dip down" model (5.10), $n = 200$, Exponential base distribution, 1,000 simulated data sets

| | $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.959 | 0.933 | 0.948 | 0.777 | 0.784 | 0.936 | 0.736 | 0.849 | 0.828 |
| Sandwich | 0.950 | 0.938 | 0.935 | 0.901 | 0.898 | 0.925 | 0.889 | 0.936 | 0.910 |
| Bootstrap | 0.959 | 0.946 | 0.940 | 0.898 | 0.904 | 0.939 | 0.895 | 0.922 | 0.906 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | 1.306 | -2.467 | -0.290 | -25.101 | -24.086 | -2.031 | -31.050 | -14.655 | -17.702 |
| Sandwich | 0.000 | -1.741 | -2.176 | -7.110 | -7.545 | -3.627 | -8.851 | -2.031 | -5.804 |
| Bootstrap | 1.306 | -0.580 | -1.451 | -7.545 | -6.674 | -1.596 | -7.980 | -4.063 | -6.384 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

intervals are robust for $\gamma_1$, $\gamma_2$ and $\beta$, as the Satorra-Bentler principle stipulates. For the other parameters (with the exception of $\psi_1$, this time), normal theory methods are awful. sandwich and bootstrap do better, but the sample size of $n = 200$ is likely not large enough for really good performance.

**Robustness for a zero covariance** In Table 5.16, there is substantial under-coverage for the non-zero covariance $\phi_{12} = cov(x_1, x_2)$; the coverage was 0.828, $z = -17.702$. Recall that the double measurement regression model (5.5) included two covariances between

error terms. In the simulations, I arbitrarily made one of these covariances ($\omega_{24}$) equal to zero, and let the other ($\omega_{13}$) be non-zero. For the exponential base distribution, in Tables 5.9, 5.10 and even the $n = 50$ Table 5.11, coverage for $\omega_{24} = 0$ ranged from good to acceptable. To see if something like this this might happen for the present model, I did another run of the exponential with $n = 200$, this time letting the covariance parameter $\phi_{12} = 0$ instead of $\phi_{12} = 1$. Table 5.17 shows the results. Empirical coverage of $\phi_{12}$ is

Table 5.17: Coverage of 95% confidence intervals for the "dip down" model (5.10) with $\phi_{12} = 0$, $n = 200$, Exponential base distribution, and 1,000 simulated data sets. **Alert: The true value of $\phi_{12} = 0$.**

| | $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.951 | 0.929 | 0.935 | 0.661 | 0.675 | 0.930 | 0.743 | 0.906 | 0.951 |
| Sandwich | 0.950 | 0.940 | 0.929 | 0.885 | 0.890 | 0.928 | 0.889 | 0.959 | 0.936 |
| Bootstrap | 0.956 | 0.945 | 0.942 | 0.894 | 0.893 | 0.949 | 0.885 | 0.917 | 0.921 |
| | | | | | $z$ Statistics* | | | | |
| Normal Theory | 0.145 | -3.047 | -2.176 | -41.933 | -39.901 | -2.902 | -30.035 | -6.384 | 0.145 |
| Sandwich | 0.000 | -1.451 | -3.047 | -9.431 | -8.706 | -3.192 | -8.851 | 1.306 | -2.031 |
| Bootstrap | 0.871 | -0.725 | -1.161 | -8.125 | -8.270 | -0.145 | -9.431 | -4.788 | -4.208 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

0.951; one almost cannot do better.

This is interesting. It is tempting to think, like [26] and others, that if the standard errors are too small, then tests will be too likely to reject $H_0$ when $H_0$ is true. That's reasonable, but if the standard errors depend on the value of the true parameter and are okay when $H_0$ is true, then the Type I error probability will not be adversely affected. Moreover, when $H_0$ is false, underestimating the standard deviation of the statistic is not a bad thing, provided you are interested in testing rather than in confidence intervals. In fact, a small standard error will lead to rejection of $H_0$ at a higher rate — that is, it will yield higher statistical power and more correct decisions. This is why we need to consider Type I error probabilities as a separate issue from the standard errors. We'll do so later, in Section 5.8. In the meantime, we have the following.

**Standard errors of covariance parameters are robust when the true value is zero** It's only a conjecture at this point, but I suspect that *in general, normal theory standard errors of covariance parameters are asymptotically correct when the variables involved are independent.* For the current model, this is fairly easy to show, because the variables involved ($x_1$ and $x_2$) are observable, and the MLE of $\phi_{12} = Cov(x_1, x_2)$ is available in closed form.

Let the observable variables $x_1$ and $x_2$ have a joint distribution that is arbitrary

except that the expected values exist up to fourth order. The variables are centered[24], so that $E(x_1) = E(x_2) = 0$. Adopt the notation $Var(x_1) = \sigma_{11}$, $Var(x_2) = \sigma_{22}$, and $Cov(x_1, x_2) = \sigma_{12}$.

The MLE of $\sigma_{12}$ is $\widehat{\sigma}_{12} = \frac{1}{n}\sum_{i=1}^{n} x_{i,1}x_{i,2}$, which is also the most natural method of moments estimator. Its true variance is

$$\begin{aligned} Var(\widehat{\sigma}_{12}) &= Var\left(\frac{1}{n}\sum_{i=1}^{n} x_{i,1}x_{i,2}\right) \\ &= \frac{1}{n^2}\sum_{i=1}^{n} Var(x_{i,1}x_{i,2}) \\ &= \frac{1}{n^2}\sum_{i=1}^{n} \left(E\{(x_{i,1}x_{i,2})^2\} - (E\{x_{i,1}x_{i,2}\})^2\right) \\ &= \frac{1}{n}\left(E\{x_1^2x_2^2\} - (E\{x_1x_2\})^2\right) \\ &= \frac{1}{n}\left(E\{x_1^2x_2^2\} - \sigma_{12}^2\right). \end{aligned} \tag{5.11}$$

To obtain the expression for $Var(\widehat{\sigma}_{12})$ that holds under (bivariate) normality, all we need is $E\{x_1^2x_2^2\}$. This can be obtained directly by integrating, but it's easier to differentiate the moment-generating function $M(\mathbf{t}) = e^{\frac{1}{2}\mathbf{t}^\top\boldsymbol{\Sigma}\mathbf{t}}$ twice with respect to $t_1$ and twice with respect to $t_2$, and then set $t_1$ and $t_2$ to zero. I did it with Sage. The first step is to set up the matrices.

```
# E(x1^2 x2^2) for bivariate normal
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
Sigma = SymmetricMatrix(2,'sigma'); show(Sigma)
t = ZeroMatrix(2,1)
t[0,0] = var('t1'); t[1,0] = var('t2'); show(t)
```

evaluate

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix}$$

$$\begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

Then, define the moment-generating function.

[24] In this theory, they are centered by subtracting off $\mu_1$ and $\mu_2$. In practice, they would be centered by subtracting off the random variables $\overline{x}_1$ and $\overline{x}_2$. Asymptotically, there is no difference, and it makes the calculations simpler to just let $E(x_1) = E(x_2) = 0$.

```
mgf = exp( 1/2 * t.transpose() * Sigma * t) # It's a 1x1 matrix
mgf = mgf[0,0]; mgf
```

evaluate

$$e^{\left(\frac{1}{2}\,\sigma_{11}t_1^2+\sigma_{12}t_1t_2+\frac{1}{2}\,\sigma_{22}t_2^2\right)}$$

Finally, differentiate the moment-generating function and set $\mathbf{t} = \mathbf{0}$.

```
d1 = derivative(mgf,t1,2); d1
d2 = derivative(d1,t2,2); d2
answ = d2(t1=0,t2=0); answ
```

evaluate

$$2\,\sigma_{12}^2 + \sigma_{11}\sigma_{22}$$

That's $E(x_1^2x_2^2)$ for the normal distribution. Using this result, the normal theory variance of the sample covariance is

$$\begin{aligned} Var(\widehat{\sigma}_{12}) &= \frac{1}{n}\left(E\{x_1^2x_2^2\} - \sigma_{12}^2\right) \\ &= \frac{1}{n}\left(\sigma_{11}\sigma_{22} + 2\,\sigma_{12}^2 - \sigma_{12}^2\right) \\ &= \frac{1}{n}\left(\sigma_{11}\sigma_{22} + \sigma_{12}^2\right), \end{aligned} \tag{5.12}$$

compared to the general $\frac{1}{n}\left(E\{x_1^2x_2^2\} - \sigma_{12}^2\right)$ from (5.11). Since normal theory standard errors are often too small with non-normal data, it is natural to suspect that perhaps (5.12) is always less than or equal to (5.11). However,

$$\begin{aligned} & E\{x_1^2x_2^2\} - \sigma_{12}^2 - \left(\sigma_{11}\sigma_{22} + \sigma_{12}^2\right) \\ = \;& E\{x_1^2x_2^2\} - E\{x_1^2\}E\{x_2^2\} - 2\,\sigma_{12}^2 \\ = \;& Cov(x_1^2, x_2^2) - 2\,Cov(x_1, x_2)^2. \end{aligned}$$

This quantity can be negative. Consider jointly distributed $x_1$ and $x_2$ with $P(x_1 = 1, x_2 = 1) = P(x_1 = -1, x_2 = -1) = \frac{1}{2}$. In this case, $Cov(x_1^2, x_2^2) = 0$, and $Cov(x_1, x_2) = 1$.

However, observe what happens when $x_1$ and $x_2$ are independent. Expression (5.11) is

$$\begin{aligned} \frac{1}{n}\left(E\{x_1^2x_2^2\} - \sigma_{12}^2\right) &= \frac{1}{n}\left(E\{x_1^2\}E\{x_2^2\} - 0\right) \\ &= \frac{1}{n}\left(\sigma_{11}\sigma_{22}\right), \end{aligned}$$

while Expression (5.12) is

$$\frac{1}{n}\left(\sigma_{11}\sigma_{22} + \sigma_{12}^2\right) = \frac{1}{n}\left(\sigma_{11}\sigma_{22}\right).$$

That is, when the variables involved are independent, the variance of the sample covariance assuming normality is correct for an arbitrary joint distribution. This makes the normal theory standard error correct, even when the normal assumption is wrong.

That last statement conveys the right idea, but the word "correct" is ambiguous. Here's what I really mean. The robust standard error of $\widehat{\sigma}_{12}$ is

$$SE_{\text{robust}} = \sqrt{\frac{1}{n}\left(\frac{\sum_{i=1}^{n} x_{i,1}^2 x_{i,2}^2}{n} \; - \; \widehat{\sigma}_{12}^2\right)}.$$

$SE_{\text{robust}}$ is "correct," but not in the sense of ordinary consistency. It goes almost surely to zero, and the quantity it is estimating,

$$SD(\widehat{\sigma}_{12}) = \sqrt{\frac{1}{n}\left(E\{x_1^2 x_2^2\} - \sigma_{12}^2\right)},$$

is a moving target that also goes to zero. The fact that they both go to zero is not good enough, because any random variable divided by zero goes to zero in probability. What makes $SE_{\text{robust}}$ good is that it's (almost surely) *asymptotically equivalent* to $SD(\widehat{\sigma}_{12})$. That is, the ratio

$$\begin{aligned}
\frac{SE_{\text{robust}}}{SD(\widehat{\sigma}_{12})} &= \frac{\sqrt{\frac{1}{n}\left(\frac{\sum_{i=1}^{n} x_{i,1}^2 x_{i,2}^2}{n} \; - \; \widehat{\sigma}_{12}^2\right)}}{\sqrt{\frac{1}{n}\left(E\{x_1^2 x_2^2\} - \sigma_{12}^2\right)}} \\
&= \frac{\sqrt{\frac{\sum_{i=1}^{n} x_{i,1}^2 x_{i,2}^2}{n} \; - \; \widehat{\sigma}_{12}^2}}{\sqrt{E\{x_1^2 x_2^2\} - \sigma_{12}^2}} \\
&\xrightarrow{a.s} \frac{\sqrt{E\{x_1^2 x_2^2\} - \sigma_{12}^2}}{\sqrt{E\{x_1^2 x_2^2\} - \sigma_{12}^2}} = 1.
\end{aligned}$$

Clearly, if $x_1$ and $x_2$ are independent, $SE_{\text{normal}} = \sqrt{\frac{1}{n}\left(\widehat{\sigma}_{11}\widehat{\sigma}_{22} + \widehat{\sigma}_{12}^2\right)}$ is equally good, because in that case,

$$\begin{aligned}
\frac{SE_{\text{normal}}}{SD(\widehat{\sigma}_{12})} &= \frac{\sqrt{\widehat{\sigma}_{11}\widehat{\sigma}_{22} + \widehat{\sigma}_{12}^2}}{\sqrt{E\{x_1^2 x_2^2\} - \sigma_{12}^2}} \\
&\xrightarrow{a.s} \frac{\sqrt{\sigma_{11}\sigma_{22} + \sigma_{12}^2}}{\sqrt{E\{x_1^2 x_2^2\} - \sigma_{12}^2}} \\
&\overset{\text{ind.}}{=} \frac{\sqrt{\sigma_{11}\sigma_{22} + 0}}{\sqrt{E\{x_1^2\}E\{x_2^2\} - 0}} \\
&= \frac{\sqrt{\sigma_{11}\sigma_{22}}}{\sqrt{\sigma_{11}\sigma_{22}}} = 1.
\end{aligned}$$

This is the exact way in which $SE_{\text{normal}}$ is robust when $x_1$ and $x_2$ are independent.

Notice that it had to be assumed that $x_1$ and $x_2$ were independent, and not merely uncorrelated[25]. The following example shows that this distinction is not just a computational convenience that allows one to write $E\{x_1^2x_2^2\} = E\{x_1^2\}E\{x_2^2\}$. It can really make a difference.

**Example 5.6.1** *Variables that have zero covariance but are not independent.*

Let the random variable $x_1$ have a density that is symmetric around zero, meaning that its density $f(x) = f(-x)$ for all real $x$. This makes $E\{x_1\} = 0$, and it will be shown in an exercise that $E\{x_1^3\} = 0$ as well. Let $x_2 = \beta_0 + x_1^2 + \epsilon$, where $\epsilon$ has expected value zero and is independent of $x_1$. The intercept $\beta_0 = -Var(x_1)$, so that $E\{x_2\} = 0$. Thus, $x_1$ and $x_2$ are both centered. We have

$$\begin{aligned} Cov(x_1, x_2) &= E\{x_1x_2\} - E\{x_1\}E\{x_2\} \\ &= E\{x_1\left(\beta_0 + x_1^2 + \epsilon\right)\} \\ &= \beta_0\, E\{x_1\} + E\{x_1^3\} + E\{x_1\}E\{\epsilon\} \\ &= 0. \end{aligned} \tag{5.13}$$

What makes this work the symmetric relationship between $x_1$ and $x_2$, as in footnote 25.

Though $x_1$ and $x_2$ have zero covariance, they are not independent. Intuitively, this is because $x_2$ depends on $x_1$ in a systematic way (plus error). I deleted a formal proof, because it was a distraction from the main message. That main message is conveyed better by a specific example.

Let $x_1$ and $\epsilon$ both have the double exponential (Laplace) density $f(x) = \frac{1}{2}e^{-|x|}$, which is symmetric around zero. The moments are given by

$$E(x^k) = \begin{cases} 0 & \text{for } k \text{ odd} \\ k! & \text{for } k \text{ even.} \end{cases}$$

Thus, $Var(x_1) = Var(\epsilon) = 2$. Let $x_2 = -2 + x_1^2 + \epsilon$, so that $Var(x_2) = 22$, and $Cov(x_1, x_2) = 0$ as in (5.13). Using (5.12), the normal theory variance of the sample covariance is

$$\begin{aligned} Var(\widehat{\sigma}_{12}) &= \frac{1}{n}\left(\sigma_{11}\sigma_{22} + \sigma_{12}^2\right) \\ &= \frac{1}{n}\left(2 \cdot 22 + 0\right) = \frac{1}{n}(44), \end{aligned}$$

[25] To review, independent random variables always have zero covariance, and if the variables are multivariate normal, then zero covariance also implies independence. For other distributions, it's not true. For example, suppose the discrete random variables variables $x$ and $y$ have joint distribution

| | $x = 1$ | $x = 2$ | $x = 3$ |
|---|---|---|---|
| $y = 1$ | 3/12 | 1/12 | 3/12 |
| $y = 2$ | 1/12 | 3/12 | 1/12 |

It is easy to verify that $Cov(x, y) = 0$, but they are not independent because $P(x = 1, y = 1) \neq P(x = 1)P(y = 1)$.

while from (5.11), the true variance of the sample covariance is

$$
\begin{aligned}
Var(\widehat{\sigma}_{12}) &= \frac{1}{n}\left(E\{x_1^2 x_2^2\} - \sigma_{12}^2\right) \\
&= \frac{1}{n}\left(E\{x_1^2(-2 + x_1^2 + \epsilon)^2\} - 0\right) \\
&= \frac{1}{n}\left(E\{x_1^2(x_1^4 + 2\,\epsilon x_1^2 + \epsilon^2 - 4\,x_1^2 - 4\,\epsilon + 4)\}\right) \\
&= \frac{1}{n}\left(E\{x_1^6 + 2\,\epsilon x_1^4 + \epsilon^2 x_1^2 - 4\,x_1^4 - 4\,\epsilon x_1^2 + 4\,x_1^2\}\right) \\
&= \frac{1}{n}\left(E\{x_1^6\} + 2\,E\{\epsilon\}E\{x_1^4\} + E\{\epsilon^2\}E\{x_1^2\} - 4\,E\{x_1^4\} - 4\,E\{\epsilon\}E\{x_1^2\} + 4\,E\{x_1^2\}\right) \\
&= \frac{1}{n}\left(E\{x_1^6\} + E\{\epsilon^2\}E\{x_1^2\} - 4\,E\{x_1^4\} + 4\,E\{x_1^2\}\right) \\
&= \frac{1}{n}\left(6! + 2!2! - 4\cdot 4! + 4\cdot 2!\right) \\
&= \frac{1}{n}\left(720 + 4 - 96 + 8\right) \\
&= \frac{1}{n}\left(636\right)
\end{aligned}
$$

There are two things to notice here. First, since $E\{x_1^2 x_2^2\} \neq E\{x_1^2\}E\{x_2^2\}$, the variables $x_1$ and $x_2$ cannot be independent. Second, 636 is a lot bigger than 44. This means that the normal theory standard error $\sqrt{\frac{1}{n}\left(\widehat{\sigma}_{11}\widehat{\sigma}_{22} + \widehat{\sigma}_{12}^2\right)}$ will radically under-estimate the true standard deviation of $\widehat{\sigma}_{12}$. It is *not* robust, and the coverage of the confidence interval will be very poor. This will now be verified with a quick simulation.

**Simulation with variables uncorrelated but not independent** As in the numerical example just given, $x_1$ and $\epsilon$ will have independent double exponential distributions, and $x_2 = -2 + x_1^2 + \epsilon$. Here's the simulation of a single data set.

It is easy enough to gererate double exponential random deviates, but we'll use the `rLaplace()` function from the `ExtDist` package, which needs to be installed and loaded. It has a `BIC` function that covers up the usual version, but that does not matter to us.

```
> rm(list=ls()); options(scipen=999)
> # install.packages('ExtDist') # Only need to do this once
> library(ExtDist)

Attaching package: ExtDist

The following object is masked from package:stats:

    BIC


> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
```

```
> library(lavaan)

This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
R output in black by default
```

Now we enter the true parameter values $\sigma_{11}$, $\sigma_{12}$ and $\sigma_{22}$ calculated earlier, and the lavaan model string.

```
> sigma11 = 2; sigma12=0; sigma22 = 22
> truth = c(sigma11, sigma12, sigma22)
> names(truth) = c('sigma11', 'sigma12', 'sigma22')
> truth
sigma11 sigma12 sigma22
      2       0      22
> # Model has variances and covariance only
> mod = 'x1 ~~ sigma11*x1; x1 ~~ sigma12*x2
+                          x2 ~~ sigma22*x2'
```

Now simulate a data set and fit the model. In the simulation study, this will be inside a loop.

```
> # Simulate one data set
> set.seed(9999); n = 200
> x1 = rLaplace(n); epsilon = rLaplace(n)
> x2 = x1^2 + epsilon - 2
> simdat = cbind(x1,x2)
>
> fit1 = lavaan(mod,simdat)
> p1 = parameterEstimates(fit1); p1

  lhs op rhs   label    est    se      z pvalue ci.lower ci.upper
1  x1 ~~  x1 sigma11  2.429 0.243 10.000  0.000    1.953    2.905
2  x1 ~~  x2 sigma12  1.993 0.652  3.058  0.002    0.715    3.270
3  x2 ~~  x2 sigma22 33.337 3.334 10.000  0.000   26.803   39.871
> ci1 = p1[,9:10] # Upper and lower confidence limits
> hit1 = as.numeric(ci1[,1] < truth & truth < ci1[,2]) # Binary for in ci
> cbind(ci1,truth,hit1)
          ci.lower  ci.upper truth hit1
sigma11  1.9526673  2.904692     2    1
sigma12  0.7154044  3.269924     0    0
sigma22 26.8032346 39.871180    22    0
```

The parameter of interest is the covariance $\sigma_{12}$. Notice that the standard error of $\widehat{\sigma}_{12}$ is 0.652. Just to verify that this is indeed equal to $\sqrt{\frac{1}{n}\left(\widehat{\sigma}_{11}\widehat{\sigma}_{22} + \widehat{\sigma}_{12}^2\right)}$,

```
> Sigmahat = var(simdat) * (n-1) / n; Sigmahat
         x1        x2
x1 2.428680  1.992664
x2 1.992664 33.337207
> # Normal theory standard error of the sample covariance
> sqrt(1/n * (Sigmahat[1,1]*Sigmahat[2,2] + Sigmahat[1,2]^2))
[1] 0.6516752
> # Bingo.
```

This was put in a simulation loop with 1,000 iterations, along with the usual

```
> fit2 = lavaan(mod, data = simdat, se='robust.huber.white')
> fit3 = lavaan(mod, data = simdat, se='bootstrap')
```

Coverage of the normal theory confidence interval should be poor, and coverage of the sandwich and bootstrap intervals should be better. The results are shown in Table 5.18.

Table 5.18: Coverage of 95% confidence intervals for the double exponential model of Example 5.6.1 with $x_1$ and $x_2$ uncorrelated but not independent, $n = 200$, one thousand simulated data sets.

| | $\sigma_{11}$ | $\sigma_{12}$ | $\sigma_{22}$ |
|---|---|---|---|
| Normal Theory | 0.773 | 0.427 | 0.291 |
| Sandwich | 0.917 | 0.948 | 0.748 |
| Bootstrap | 0.919 | 0.911 | 0.772 |
| $z$ Statistics* | | | |
| Normal Theory | -25.682 | -75.885 | -95.618 |
| Sandwich | -4.788 | -0.290 | -29.309 |
| Bootstrap | -4.498 | -5.659 | -25.827 |

* Bonferroni critical value for 9 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 2.77.

The main hypothesis is confirmed. Coverage of the normal theory confidence interval for $\sigma_{12}$ is terrible, while for the sandwich method it's excellent. It's a little odd that coverage of the bootstrap interval, while far better than normal theory, is substandard. I expected sandwich and the bootstrap to yield very similar results. To see if it was a coincidence, I re-ran the code and got a coverage of 0.908, $z = -6.094$ for the bootstrap confidence interval. Coverage was excellent again for the sandwich confidence interval, and of course very bad for the normal theory interval. I then tried $n = 1,000$. The result for the bootstrap was better, with a coverage of 0.932 and a $z$ statistic of $-2.612$. This is technically okay because it does not quite exceed the Bonferroni critical value, but I tried $n = 2,000$ anyway. The bootstrap results were good this time, with coverage = 0.943 and $z = -1.016$. So while the sandwich and the bootstrap often give nearly the same results, it does not always happen. Here, the bootstrap appears to require a much bigger sample size than the sandwich.

**The conjecture** I feel that this has been a valuable digression[26]. Here's the main point. It has been shown that when two observable random variables are independent, the normal theory standard error for their covariance is robust. *I strongly suspect that this is also true for latent variables, including error terms.* I am not able to supply a proof at this point, but we can and will check the hypothesis in further simulations.

It is important to clarify that the claim above does not contradict the Satorra-Bentler principle. That principle says that normal theory standard errors for straight-arrow parmeters are robust, not that the standard errors for parameters on curved, double-headed arrows (that is, covariance parameters) are always non-robust.

**Back to the Dip Down model with an exponential base distribution** Let us return to gathering evidence about robustness. Before I took us off on a side trip, we saw in Table 5.16 that normal theory standard errors for the straight-arrow parameters $\gamma_1$, $\gamma_2$ and $\beta$ were robust as expected. Performance of the other standard errors, including the sandwich and bootstrap, were generally poor. Suspecting that $n = 200$ was too small, I tried $n = 1,000$. Table 5.19 shows the result. Note that we have returned to $\phi_{12} \neq 0$, so for this parameter, we don't expect robustness for the normal theory interval.

Table 5.19: Coverage of 95% confidence intervals for the "dip down" model (5.10), $n = 1,000$, Exponential base distribution, 1,000 simulated data sets

| | $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.955 | 0.962 | 0.963 | 0.756 | 0.798 | 0.933 | 0.732 | 0.834 | 0.789 |
| Sandwich | 0.958 | 0.960 | 0.959 | 0.934 | 0.942 | 0.947 | 0.934 | 0.947 | 0.931 |
| Bootstrap | 0.957 | 0.963 | 0.957 | 0.925 | 0.942 | 0.947 | 0.935 | 0.941 | 0.934 |
| | | | | | $z$ Statistics* | | | | |
| Normal Theory | 0.725 | 1.741 | 1.886 | -28.148 | -22.054 | -2.467 | -31.631 | -16.831 | -23.360 |
| Sandwich | 1.161 | 1.451 | 1.306 | -2.322 | -1.161 | -0.435 | -2.322 | -0.435 | -2.757 |
| Bootstrap | 1.016 | 1.886 | 1.016 | -3.627 | -1.161 | -0.435 | -2.176 | -1.306 | -2.322 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

With $n = 1,000$, the sandwich and bootstrap intervals now perform well enough, except for the the bootstrap interval for $\phi_{11}$. For completeness, I also tried $n = 50$. Results are not shown. The conclusion is that if $n = 200$ is too small, one should not expect good things to happen with $n = 50$.

[26]The online Merriam-Webster dictionary defines a digression as "the act or an instance of leaving the main subject in an extended written or verbal expression of thought." One of my colleagues was once known to his students as the Doctor of Digression. Maybe my students say the same thing about me.

**Scaled beta base distribution, Dip down model**

Recall that for the non-normal but light-tailed scaled beta base distribution, the normal theory intervals performed well at $n = 200$ for the extra response variable regression model (5.9) and the double measurement regression model (5.5). This applied to the variance and covariance parameters as well as the straight-arrow parameters whose robustness is guaranteed by the Satorra-Bentler principle. As mentioned previously, this supports the idea [13] that lack of robustness for normal theory methods comes specifically from heavy tails than from just departure from normality.

In Table 5.20, we try the scaled beta distribution with the dip down model and $n = 200$. Coverage of the normal theory confidence intervals is good for the straight arrow parameters $\gamma_1$, $\gamma_2$ and $\beta$, but not for $\phi_{11} = Var(x_1)$ or $\phi_{22} = Var(x_2)$. The sandwich and the bootstrap also perform badly for $\phi_{11}$ and $\phi_{22}$.

Table 5.20: Coverage of 95% confidence intervals for the "dip down" model (5.10), $n = 200$, Scaled beta base distribution, 1,000 simulated data sets

| | $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.942 | 0.956 | 0.947 | 0.923 | 0.925 | 0.946 | 0.931 | 0.955 | 0.944 |
| Sandwich | 0.941 | 0.958 | 0.941 | 0.910 | 0.921 | 0.938 | 0.936 | 0.951 | 0.936 |
| Bootstrap | 0.951 | 0.958 | 0.946 | 0.909 | 0.923 | 0.942 | 0.938 | 0.946 | 0.933 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | -1.161 | 0.871 | -0.435 | -3.918 | -3.627 | -0.580 | -2.757 | 0.725 | -0.871 |
| Sandwich | -1.306 | 1.161 | -1.306 | -5.804 | -4.208 | -1.741 | -2.031 | 0.145 | -2.031 |
| Bootstrap | 0.145 | 1.161 | -0.580 | -5.949 | -3.918 | -1.161 | -1.741 | -0.580 | -2.467 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

Before getting depressed about this, recall that for $n = 200$ and normal data (Table 5.13), the normal theory intervals also did not perform very well for parameters other than $\gamma_1$, $\gamma_2$ and $\beta$. It's true that coverage did not quite reach a significant departure from 95%, but still it was bad enough for us to go to $n = 500$, where everything was fine. Table 5.21 shows $n = 500$ for the scaled beta distribution.

Now the coverage of the normal theory confidence intervals never differs significantly from 95%, and it's actually good except for $\phi_{22}$. In a slightly shaky voice, we maintain the conclusion that in the absence of outliers, normal theory standard errors are okay for non-normal data, even for the variance and covariance parameters excluded from the Satorra-Bentler principle.

### 5.6.4 The Standardized Two-factor Model

We have collected a heavy load of evidence already, but I decided to include one more model. This one is a basic confirmatory factor analysis model (see Chapter 3) with two

Table 5.21: Coverage of 95% confidence intervals for the "dip down" model (5.10), $n = 500$, Scaled beta base distribution, 1,000 simulated data sets

| | $\gamma_1$ | $\gamma_2$ | $\beta$ | $\phi_{11}$ | $\phi_{22}$ | $\psi_1$ | $\psi_2$ | $\omega$ | $\phi_{12}$ |
|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.956 | 0.937 | 0.943 | 0.945 | 0.935 | 0.950 | 0.949 | 0.957 | 0.943 |
| Sandwich | 0.959 | 0.942 | 0.946 | 0.942 | 0.938 | 0.947 | 0.951 | 0.953 | 0.944 |
| Bootstrap | 0.961 | 0.943 | 0.944 | 0.940 | 0.941 | 0.949 | 0.953 | 0.957 | 0.947 |
| | | | | $z$ Statistics* | | | | | |
| Normal Theory | 0.871 | -1.886 | -1.016 | -0.725 | -2.176 | 0.000 | -0.145 | 1.016 | -1.016 |
| Sandwich | 1.306 | -1.161 | -0.580 | -1.161 | -1.741 | -0.435 | 0.145 | 0.435 | -0.871 |
| Bootstrap | 1.596 | -1.016 | -0.871 | -1.451 | -1.306 | -0.145 | 0.435 | 1.016 | -0.435 |

* Bonferroni critical value for 27 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.11.

latent variables called factors, and three observable variables per factor. For identifiability, the variances of the factors are set to one, so that the covariance between them is a correlation. Three features make the model attractive enough to include in the simulations. First, earlier results lead one to expect that the normal theory standard error of the correlation will be robust when the factors are independent. Second, because correlations between factors in this surrogate model equal the correlations between factors for the original model (!), parameter estimation is of interest in its own right for a change. In particular, the confidence interval for the correlation is something one would want to know, and not just a convenient metric for judging whether the confidence interval is "too small." Third, the variances of the factors are constrained to equal one, and a careful reading of the Satorra-Bentler principle tells us it does not apply in this case. Therefore, normal theory standard erros for the factor loadings (the straight-arrow parameters) might not be robust. It will be interesting to see.

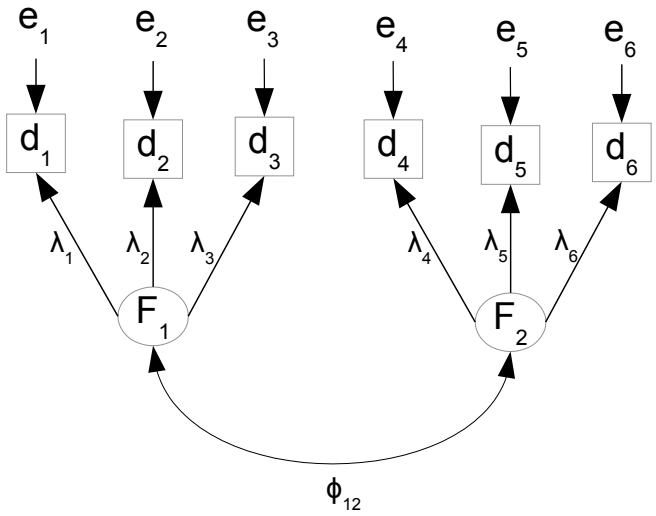
Figure 5.7 is a reproduction of Figure 3.4 from Chapter 3. The model equations are

$$\begin{aligned} d_1 &= \lambda_1 F_1 + e_1 \\ d_2 &= \lambda_2 F_1 + e_2 \\ d_3 &= \lambda_3 F_1 + e_3 \\ d_4 &= \lambda_4 F_2 + e_4 \\ d_5 &= \lambda_5 F_2 + e_5 \\ d_6 &= \lambda_6 F_2 + e_6, \end{aligned} \tag{5.14}$$

with $Var(F_1) = Var(F_2) = 1$, $Cov(F_1, F_2) = \phi_{12}$ (a correlation), and $Var(e_j) = \omega_j$ for $j = 1, \ldots, 6$. As indicated on the path diagram, the error terms $e_1, \ldots, e_6$ are independent of one another and of the factors. For the simulations, the parameter values will be

| $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\phi_{12}$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.0 | 2.0 | 3.0 | 1.0 | 2.0 | 3.0 | 0.5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Figure 5.7: Two Standardized Factors




### Normal base distribution

Here is an R session showing the simulation of a single data set. First comes the setup.

```
> rm(list=ls()); options(scipen=999)
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
> # Set true parameter values
> # Covariance between exogenous variables (factors) will come from adding delta
> # to both factors, with Var(delta) = phi12, x1 = t1 + delta and x2 = t2 + delta
> phi12 = 0.5 # phi11 = 1; phi22 = 1
> lambda1 = 1; lambda2 = 2; lambda3 = 3; lambda4 = 1; lambda5 = 2; lambda6 = 3
> omega1  = 1; omega2  = 1; omega3  = 1; omega4  = 1; omega5  = 1; omega6  = 1
> k = 1 # Scaling constant to make variance of base distribution = one
> # Calculate variances of t1 and t2
> v1 = 1-phi12; v2=1-phi12
> truth = c(lambda1,lambda2,lambda3,lambda4,lambda5,lambda6, phi12,
+           omega1,omega2,omega3,omega4,omega5,omega6)
> namz = c('lambda1','lambda2','lambda3','lambda4','lambda5','lambda6',
+          'phi12', 'omega1','omega2','omega3','omega4','omega5','omega6')
> names(truth)=namz; truth
lambda1 lambda2 lambda3 lambda4 lambda5 lambda6   phi12  omega1  omega2  omega3  omega4  omega5
    1.0     2.0     3.0     1.0     2.0     3.0     0.5     1.0     1.0     1.0     1.0     1.0
 omega6
    1.0
> # Here are 2 good true null hypotheses.
```

```
> # H0: lambda1=lambda4, lambda2=lambda4, lambda3=lambda6     and
> # H0: omega1=omega2=omega3=omega4=omega5=omega6
```

Now we simulate one data set, define the lavaan model, fit it, and display the results.

```
> n = 200; set.seed(9999)
> delta = sqrt(phi12)*k*rnorm(n); t1 = sqrt(v1)*k*rnorm(n)
> t2 = sqrt(v2)*k*rnorm(n)
> F1 = t1 + delta; F2 = t2 + delta # Making Corr(F1,F2) = phi12
> e1 = sqrt(omega1)*k*rnorm(n); e2 = sqrt(omega2)*k*rnorm(n)
> e3 = sqrt(omega3)*k*rnorm(n); e4 = sqrt(omega4)*k*rnorm(n)
> e5 = sqrt(omega5)*k*rnorm(n); e6 = sqrt(omega6)*k*rnorm(n)
> d1 = lambda1*F1 + e1; d2 = lambda2*F1 + e2; d3 = lambda3*F1 + e3
> d4 = lambda4*F2 + e4; d5 = lambda5*F2 + e5; d6 = lambda6*F2 + e6
> simdat = cbind(d1,d2,d3,d4,d5,d6)
>
> mod = ' # Measurement model
+         F1 =~ lambda1*d1 + lambda2*d2 + lambda3*d3
+         F2 =~ lambda4*d4 + lambda5*d5 + lambda6*d6
+         # Variances and covariances
+         F1 ~~ 1*F1; F1 ~~ phi12*F2; F2 ~~ 1*F2
+         d1 ~~ omega1*d1; d2 ~~ omega2*d2; d3 ~~ omega3*d3
+         d4 ~~ omega4*d4; d5 ~~ omega5*d5; d6 ~~ omega6*d6
+         # Constraints for identifiability
+         lambda1 > 0; lambda4 > 0
+       '
> fit1 = lavaan(mod,data=simdat)
>
> p1 = parameterEstimates(fit1); p1
```

| | lhs | op | rhs | label | est | se | z | pvalue | ci.lower | ci.upper |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | F1 | =~ | d1 | lambda1 | 1.010 | 0.084 | 12.069 | 0.000 | 0.846 | 1.174 |
| 2 | F1 | =~ | d2 | lambda2 | 2.007 | 0.130 | 15.383 | 0.000 | 1.752 | 2.263 |
| 3 | F1 | =~ | d3 | lambda3 | 3.178 | 0.184 | 17.245 | 0.000 | 2.817 | 3.539 |
| 4 | F2 | =~ | d4 | lambda4 | 1.074 | 0.090 | 11.995 | 0.000 | 0.899 | 1.250 |
| 5 | F2 | =~ | d5 | lambda5 | 1.838 | 0.127 | 14.436 | 0.000 | 1.588 | 2.087 |
| 6 | F2 | =~ | d6 | lambda6 | 2.837 | 0.171 | 16.614 | 0.000 | 2.502 | 3.172 |
| 7 | F1 | ~~ | F1 | | 1.000 | 0.000 | NA | NA | 1.000 | 1.000 |
| 8 | F1 | ~~ | F2 | phi12 | 0.612 | 0.051 | 12.095 | 0.000 | 0.513 | 0.711 |
| 9 | F2 | ~~ | F2 | | 1.000 | 0.000 | NA | NA | 1.000 | 1.000 |
| 10 | d1 | ~~ | d1 | omega1 | 0.813 | 0.091 | 8.977 | 0.000 | 0.635 | 0.990 |
| 11 | d2 | ~~ | d2 | omega2 | 1.138 | 0.183 | 6.236 | 0.000 | 0.780 | 1.496 |
| 12 | d3 | ~~ | d3 | omega3 | 1.126 | 0.370 | 3.039 | 0.002 | 0.400 | 1.852 |
| 13 | d4 | ~~ | d4 | omega4 | 0.912 | 0.105 | 8.707 | 0.000 | 0.707 | 1.118 |
| 14 | d5 | ~~ | d5 | omega5 | 1.267 | 0.188 | 6.735 | 0.000 | 0.898 | 1.635 |
| 15 | d6 | ~~ | d6 | omega6 | 1.156 | 0.344 | 3.361 | 0.001 | 0.482 | 1.830 |

```
> ci1 = p1[ -c(7,9), 9:10] # Upper and lower confidence limits
> hit1 = as.numeric(ci1[,1] < truth & truth < ci1[,2]) # Binary for in ci
> cbind(namz,ci1,truth,hit1)
      namz  ci.lower  ci.upper truth hit1
1  lambda1 0.8460581 1.1741316   1.0    1
2  lambda2 1.7515160 2.2630119   2.0    1
3  lambda3 2.8170547 3.5394860   3.0    1
4  lambda4 0.8987614 1.2498563   1.0    1
5  lambda5 1.5883070 2.0873371   2.0    1
6  lambda6 2.5021979 3.1715405   3.0    1
8    phi12 0.5127342 0.7110450   0.5    0
10  omega1 0.6353314 0.9902481   1.0    0
11  omega2 0.7803893 1.4958409   1.0    1
12  omega3 0.3997169 1.8518908   1.0    1
13  omega4 0.7069451 1.1176770   1.0    1
14  omega5 0.8979403 1.6350932   1.0    1
15  omega6 0.4818202 1.8297737   1.0    1
```

Table 5.22 shows the results for 1,000 simulated data sets with a normal base distribution and $n = 200$.

Table 5.22: Coverage of 95% confidence intervals for the two-factor model (5.14), $n = 200$, Normal base distribution, 1,000 simulated data sets

| | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\phi_{12}$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.950 | 0.941 | 0.944 | 0.945 | 0.951 | 0.942 | 0.940 | 0.939 | 0.924 | 0.942 | 0.943 | 0.949 | 0.948 |
| Sandwich | 0.942 | 0.941 | 0.942 | 0.940 | 0.943 | 0.939 | 0.939 | 0.928 | 0.909 | 0.943 | 0.938 | 0.944 | 0.950 |
| Bootstrap | 0.949 | 0.934 | 0.942 | 0.943 | 0.940 | 0.940 | 0.946 | 0.932 | 0.905 | 0.937 | 0.932 | 0.942 | 0.944 |
| | | | | | | | $z$ Statistics* | | | | | | |
| Normal Theory | 0.000 | -1.306 | -0.871 | -0.725 | 0.145 | -1.161 | -1.451 | -1.596 | -3.772 | -1.161 | -1.016 | -0.145 | -0.290 |
| Sandwich | -1.161 | -1.306 | -1.161 | -1.451 | -1.016 | -1.596 | -1.596 | -3.192 | -5.949 | -1.016 | -1.741 | -0.871 | 0.000 |
| Bootstrap | -0.145 | -2.322 | -1.161 | -1.016 | -1.451 | -1.451 | -0.580 | -2.612 | -6.529 | -1.886 | -2.612 | -1.161 | -0.871 |

* Bonferroni critical value for 39 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.22.

The normal theory coverage is fine except for $\omega_2$. There is nothing special to distinguish $\omega_2$ from the other error variances, and they are all fine. I ran another simulation (omitting the very time-consuming bootstrap), and this time $\omega_2$ was okay but $\omega_4$ was significantly under-covered. I tried it again with $n = 500$ (no bootstrap), and everything was good. It was probably just sample size. Accordingly, I ran a full simulation with $n = 500$. The results are in Table 5.23.

All is well. The necessary sample size is a bit larger than one might expect for such a simple model, but we live and learn.

Table 5.23: Coverage of 95% confidence intervals for the two-factor model (5.14), $n = 500$, Normal base distribution, 1,000 simulated data sets

| | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\phi_{12}$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.955 | 0.952 | 0.950 | 0.943 | 0.949 | 0.940 | 0.952 | 0.944 | 0.96 | 0.960 | 0.946 | 0.953 | 0.958 |
| Sandwich | 0.956 | 0.956 | 0.946 | 0.945 | 0.947 | 0.939 | 0.943 | 0.943 | 0.96 | 0.958 | 0.947 | 0.949 | 0.952 |
| Bootstrap | 0.955 | 0.956 | 0.944 | 0.942 | 0.948 | 0.943 | 0.950 | 0.939 | 0.96 | 0.959 | 0.944 | 0.948 | 0.954 |
| | | | | | | | $z$ Statistics* | | | | | | |
| Normal Theory | 0.725 | 0.290 | 0.000 | -1.016 | -0.145 | -1.451 | 0.290 | -0.871 | 1.451 | 1.451 | -0.580 | 0.435 | 1.161 |
| Sandwich | 0.871 | 0.871 | -0.580 | -0.725 | -0.435 | -1.596 | -1.016 | -1.016 | 1.451 | 1.161 | -0.435 | -0.145 | 0.290 |
| Bootstrap | 0.725 | 0.871 | -0.871 | -1.161 | -0.290 | -1.016 | 0.000 | -1.596 | 1.451 | 1.306 | -0.871 | -0.290 | 0.580 |

* Bonferroni critical value for 39 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.22.

### Exponential base distribution

Table 5.24 shows confidence interval coverage for $n = 200$ and the exponential base distribution. Now we are seeing something different. For the first time, coverage for the straight-line parameters (in this case, the factor loadings $\lambda_1, \ldots, \lambda_6$) is really bad. As mentioned earlier, this does not contradict the Satorra-Bentler principle, because $Var(F_1) = Var(F_2) = 1$ represents a constraint on covariance matrix of the exogenous variables. On the other hand, Table 5.24 does strongly contradict Anderson and Amemiya's 1988 paper [4], which claims robustness for the factor loadings in a general factor analysis model.

Table 5.24: Coverage of 95% confidence intervals for the two-factor model (5.14), $n = 200$, Exponential base distribution, 1,000 simulated data sets

| | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\phi_{12}$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.889 | 0.832 | 0.811 | 0.889 | 0.851 | 0.821 | 0.824 | 0.710 | 0.834 | 0.908 | 0.705 | 0.859 | 0.927 |
| Sandwich | 0.921 | 0.919 | 0.922 | 0.923 | 0.927 | 0.916 | 0.912 | 0.891 | 0.917 | 0.937 | 0.894 | 0.927 | 0.947 |
| Bootstrap | 0.923 | 0.915 | 0.916 | 0.921 | 0.918 | 0.915 | 0.928 | 0.893 | 0.923 | 0.931 | 0.898 | 0.929 | 0.943 |
| | | | | | | | $z$ Statistics* | | | | | | |
| Normal Theory | -8.851 | -17.121 | -20.168 | -8.851 | -14.364 | -18.717 | -18.282 | -34.823 | -16.831 | -6.094 | -35.548 | -13.204 | -3.337 |
| Sandwich | -4.208 | -4.498 | -4.063 | -3.918 | -3.337 | -4.933 | -5.514 | -8.561 | -4.788 | -1.886 | -8.125 | -3.337 | -0.435 |
| Bootstrap | -3.918 | -5.078 | -4.933 | -4.208 | -4.643 | -5.078 | -3.192 | -8.270 | -3.918 | -2.757 | -7.545 | -3.047 | -1.016 |

* Bonferroni critical value for 39 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.22.

Returning to Table 5.24, the sandwich and bootstrap confidence intervals do a lot better than normal theory, but they are still not acceptable. A larger sample size is required. When the normal assumption was satisfied, this model required $n = 500$ for good performance, so this is no surprise. I tried $n = 500$ for the exponential base distribution (with no bootstrap), and the sandwich's performance was still substandard, with several $z$ values in the $-4$ to $-5$ range. The sandwich looked okay in another trial run with $n = 1,000$ (normal theory was still a disaster) except for $\omega_4$, so I produced Table 5.25.

As you can see, now it's okay except that the sandwich and bootstrap $z$ values for $\omega_1$

Table 5.25: Coverage of 95% confidence intervals for the two-factor model (5.14), $n = 1,000$, Exponential base distribution, 1,000 simulated data sets

| | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\phi_{12}$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.899 | 0.850 | 0.812 | 0.896 | 0.858 | 0.856 | 0.842 | 0.698 | 0.840 | 0.925 | 0.706 | 0.837 | 0.925 |
| Sandwich | 0.946 | 0.944 | 0.940 | 0.944 | 0.946 | 0.953 | 0.940 | 0.922 | 0.940 | 0.951 | 0.935 | 0.937 | 0.956 |
| Bootstrap | 0.941 | 0.943 | 0.940 | 0.939 | 0.945 | 0.948 | 0.943 | 0.919 | 0.941 | 0.948 | 0.937 | 0.939 | 0.950 |
| | | | | | | $z$ Statistics* | | | | | | | |
| Normal Theory | -7.400 | -14.510 | -20.023 | -7.835 | -13.349 | -13.639 | -15.670 | -36.564 | -15.960 | -3.627 | -35.403 | -16.396 | -3.627 |
| Sandwich | -0.580 | -0.871 | -1.451 | -0.871 | -0.580 | 0.435 | -1.451 | -4.063 | -1.451 | 0.145 | -2.176 | -1.886 | 0.871 |
| Bootstrap | -1.306 | -1.016 | -1.451 | -1.596 | -0.725 | -0.290 | -1.016 | -4.498 | -1.306 | -0.290 | -1.886 | -1.596 | 0.000 |

* Bonferroni critical value for 39 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.22.

are over the Bonferroni line. It's like Whack-a-mole. Most likely an even larger sample size is required before things truly settle down, but we will let it go now. Just note, though, that it's easy to increase the sample size in a simulation. In a real study, a few hundred more subjects could easily cost another several hundred person hours to collect, enter and clean the data — or more, if participants in the study are not just filling out questionnaires.

It's unfortunate that a factor analysis model with standardized factors seems to require such a large sample size when the data are not normal. As mentioned earlier, for a model with standardized factors, the covariances between factors under the surrogate model are exactly the correlations between factors under the original model; se Chapter 3. Correlations are very interpretable, and this is one of the few cases I know where confidence intervals for the parameters of a surrogate model are of real interest. It seems that for such confidence intervals to be meaningful for non-normal data, the sample necessary sample size will be inconveniently large. For most data sets, checking normality is probably a good idea.

**Zero correlation between factors** The model under consideration provides an opportunity to check the performance of the normal theory standard error when the factors are independent. The reader may recall that we have a running hypothesis here. The hypothesis is that when two exogenous variables (including error terms) are independent and not merely uncorrelated, the normal theory standard error is a good estimate of the true standard deviation of the covariance. We know this to be true when the variables in question are observable, and from simulation results it seems to be true of at least some error terms. Now we'll check an example of latent exogenous variables that are not errors.

Table 5.26 shows simulation results for an exponential base distribution and $n = 200$ when the true value of $\phi_{12} = 0$ because $F_1$ and $F_2$ are independent. All the other parameter values are the same as in earlier simulations. The normal theory coverage of $\phi_{12}$ is 0.933, with $z = -2.467$. This is not significantly different from 0.95 with the Bonferroni correction, but it's not stellar either. I ran a replication with a different random number seed and no bootstrap, and the empirical coverage was 0.942 ($z = -1.161$) for the normal theory interval, and 0.928 ($z = -3.192$) for the sandwich. My conclusion is that the normal theory standard error for $\phi_{12}$ is good, even at $n = 200$ and a heavy-tailed base

Table 5.26: Coverage of 95% confidence intervals for the two-factor model (5.14) **with independent factors** , $n = 200$, Exponential base distribution, 1,000 simulated data sets

| | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\phi_{12}$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.843 | 0.780 | 0.722 | 0.832 | 0.757 | 0.747 | 0.933 | 0.712 | 0.867 | 0.941 | 0.706 | 0.862 | 0.927 |
| Sandwich | 0.922 | 0.914 | 0.903 | 0.921 | 0.918 | 0.907 | 0.917 | 0.881 | 0.937 | 0.939 | 0.867 | 0.926 | 0.948 |
| Bootstrap | 0.920 | 0.912 | 0.899 | 0.919 | 0.911 | 0.904 | 0.933 | 0.880 | 0.932 | 0.933 | 0.870 | 0.914 | 0.941 |
| | | | | | | $z$ Statistics* | | | | | | | |
| Normal Theory | -15.525 | -24.666 | -33.082 | -17.121 | -28.003 | -29.454 | -2.467 | -34.533 | -12.043 | -1.306 | -35.403 | -12.768 | -3.337 |
| Sandwich | -4.063 | -5.223 | -6.819 | -4.208 | -4.643 | -6.239 | -4.788 | -10.012 | -1.886 | -1.596 | -12.043 | -3.482 | -0.290 |
| Bootstrap | -4.353 | -5.514 | -7.400 | -4.498 | -5.659 | -6.674 | -2.467 | -10.157 | -2.612 | -2.467 | -11.608 | -5.223 | -1.306 |

* Bonferroni critical value for 39 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.22.

distribution. I'm now convinced that the phenomenon is quite general. When exogenous variables, including error terms, are independent, normal theory standard errors are good.

### Scaled beta base distribution

Another running hypothesis is that normal theory standard errors work well with non-normal data, provided the distribution is not heavy tailed; that is, there is minimal excess kurtosis. Our example is a scaled version of the beta distribution with $\alpha = 3$ and $\beta = 1$, so that its density increases like $y = x^2$.

Table 5.27 shows simulation results for $n = 200$. Coverage of the normal theory intervals is within acceptable limits for all the parameters except $\omega_1$. Also, coverage $\omega_5$ is a bit low, though not quite significantly different from 0.05 with the Bonferroni correction. It is worth noting that in contrast to what happened with the exponential

Table 5.27: Coverage of 95% confidence intervals for the two-factor model (5.14), $n = 200$, Scaled beta base distribution, 1,000 simulated data sets

| | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\phi_{12}$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.939 | 0.949 | 0.943 | 0.958 | 0.939 | 0.949 | 0.939 | 0.921 | 0.947 | 0.948 | 0.938 | 0.928 | 0.947 |
| Sandwich | 0.937 | 0.943 | 0.939 | 0.950 | 0.935 | 0.935 | 0.931 | 0.915 | 0.942 | 0.946 | 0.933 | 0.928 | 0.949 |
| Bootstrap | 0.931 | 0.940 | 0.940 | 0.952 | 0.936 | 0.940 | 0.939 | 0.918 | 0.947 | 0.944 | 0.933 | 0.927 | 0.951 |
| | | | | | | $z$ Statistics* | | | | | | | |
| Normal Theory | -1.596 | -0.145 | -1.016 | 1.161 | -1.596 | -0.145 | -1.596 | -4.208 | -0.435 | -0.290 | -1.741 | -3.192 | -0.435 |
| Sandwich | -1.886 | -1.016 | -1.596 | 0.000 | -2.176 | -2.176 | -2.757 | -5.078 | -1.161 | -0.580 | -2.467 | -3.192 | -0.145 |
| Bootstrap | -2.757 | -1.451 | -1.451 | 0.290 | -2.031 | -1.451 | -1.596 | -4.643 | -0.435 | -0.871 | -2.467 | -3.337 | 0.145 |

* Bonferroni critical value for 39 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.22.

base distribution, coverage was respectable for the straight-arrow parameters $\lambda_1, \ldots, \lambda_6$.

In Table 5.28, the sample size is increased to $n = 500$.

Table 5.28: Coverage of 95% confidence intervals for the two-factor model (5.14), $n = 500$, Scaled beta base distribution, 1,000 simulated data sets

| | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda_6$ | $\phi_{12}$ | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Normal Theory | 0.967 | 0.962 | 0.948 | 0.952 | 0.953 | 0.947 | 0.950 | 0.956 | 0.952 | 0.940 | 0.940 | 0.948 | 0.959 |
| Sandwich | 0.963 | 0.957 | 0.946 | 0.951 | 0.952 | 0.948 | 0.953 | 0.954 | 0.949 | 0.942 | 0.949 | 0.952 | 0.954 |
| Bootstrap | 0.965 | 0.955 | 0.943 | 0.949 | 0.946 | 0.947 | 0.956 | 0.955 | 0.952 | 0.941 | 0.947 | 0.952 | 0.952 |
| | | | | | | | $z$ Statistics* | | | | | | |
| Normal Theory | 2.467 | 1.741 | -0.290 | 0.290 | 0.435 | -0.435 | 0.000 | 0.871 | 0.290 | -1.451 | -1.451 | -0.29 | 1.306 |
| Sandwich | 1.886 | 1.016 | -0.580 | 0.145 | 0.290 | -0.290 | 0.435 | 0.580 | -0.145 | -1.161 | -0.145 | 0.29 | 0.580 |
| Bootstrap | 2.176 | 0.725 | -1.016 | -0.145 | -0.580 | -0.435 | 0.871 | 0.725 | 0.290 | -1.306 | -0.435 | 0.29 | 0.290 |

* Bonferroni critical value for 39 two-sided $z$-tests of $H_0$: Coverage = 0.95 is 3.22.

Ah, that is satisfying! Everything is okay. Note that it also took $n = 500$ with normal data for the normal theory intervals to perform well. This is more evidence of robustness for normal theory standard errors when the non-normal distribution is not excessively heavy-tailed.

### 5.6.5 Big Data: One factor and 50 observed variables

So far, the simulations have been based on models with fairly small numbers of parameters and observed variables. Such models are good for developing understanding and often reveal the true nature of what is going on. It must be admitted, though, that structural equation models for real research data sets are often much larger. It is legitimate to wonder how well robustness extends to big models. Perhaps numerical problems emerge, or the required sample size increases rapidly with the size of the problem. The following is only one example, but the result are encouraging.

The "Big Data" simulations will be based on a confirmatory factor analysis model (see Chapter 3) with a single factor[27] and fifty observed variables. It also may be viewed as an expanded version of the Extra Response Variable Regression model (5.9) used in our first set of simulations — except that this one has 48 "extra" variables.

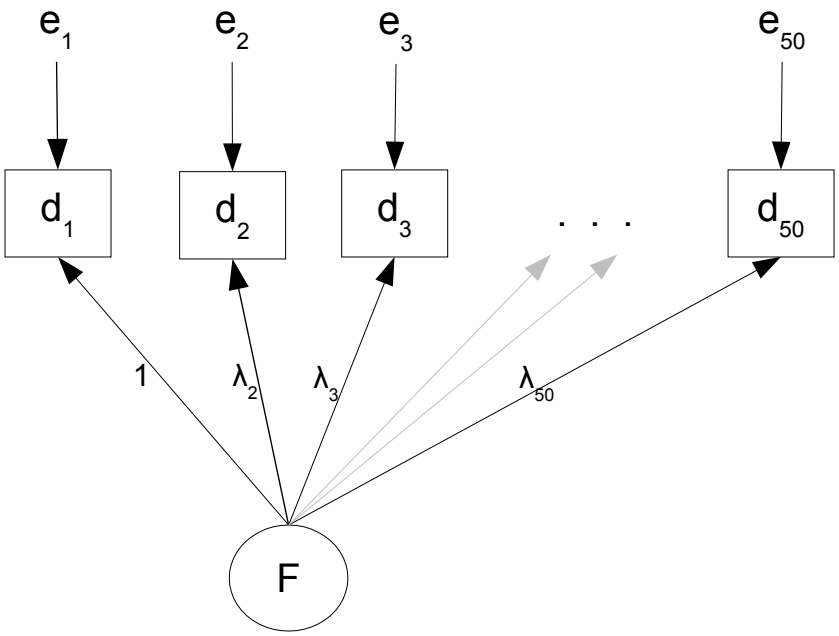
Independently for $i = 1, \ldots, n$ and $j = 1, \ldots, 50$, let

$$d_{i,j} = \lambda_j F_i + e_{i,j}, \tag{5.15}$$

where $Var(F_i) = \phi$, $Var(e_{i,j}) = \omega_j$, and all expected values equal zero. The parameter $\lambda_1$ is fixed to the known value of one, for parameter identifiability. Figure 5.8 shows a path diagram.

[27] Does this make it a toy model? Well, maybe, but if so it's a toy that presents a choking hazard. Also, the historical origin of factor analysis was Spearman's (1904) treatise on the general intelligence factor [60], and that single-factor model is not a toy.

Figure 5.8: Path diagram of Big Data model (5.15)




In the simulations, all the true parameter values were set to one. As you will see, this makes it easy to write efficient code. First, I will show you an R trick. Suppose you have an $n \times 1$ vector $\mathbf{a}$ and a square matrix $\mathbf{B}$. According to the rules of matrix algebra, you can't add these two quantities. However, R will give it a try. If you type `a + B`, then R will start adding the elements of $\mathbf{a}$ to the elements of $\mathbf{B}$, going down the columns of $\mathbf{B}$. If it runs out of $\mathbf{a}$ elements (or $\mathbf{B}$ elements), then it just starts over with another copy of the smaller object. If it does not come out even in the end, R issues a warning. If it does come out even, then R assumes that's what you intended and is silent. Here is an example.

```
> f = 1:10; eek = runif(50); dim(eek) = c(10,5)
> f+eek # This adds f to each column of eek.
          [,1]      [,2]      [,3]      [,4]      [,5]
 [1,]  1.690330  1.081871  1.747586  1.366547  1.690870
 [2,]  2.750996  2.731330  2.496223  2.443276  2.967938
 [3,]  3.886479  3.508633  3.769627  3.463070  3.069670
 [4,]  4.062570  4.942303  4.666011  4.448823  4.694002
 [5,]  5.595410  5.680228  5.648273  5.419548  5.458454
 [6,]  6.422430  6.617219  6.634463  6.998574  6.485297
 [7,]  7.202352  7.108045  7.723994  7.453080  7.655071
 [8,]  8.574795  8.033842  8.053910  8.159130  8.966246
 [9,]  9.683287  9.234638  9.301378  9.875964  9.416682
[10,] 10.198118 10.703119 10.135289 10.491988 10.915884
```

Because the numbers in `eek` are between zero and one, you can see that the number one has been added to all the numbers in the first row, the number two has been added to all the numbers in the second row, and so on. This could be done with a loop, but it would be a lot slower, which matters in simulations. It could also be accomplished with matrix multiplication, but the matrices can become quite large.

Here's the code for simulating one data set and fitting the model.

```
rm(list=ls()); options(scipen=999)
# install.packages("lavaan", dependencies = TRUE) # Only need to do this once
library(lavaan)

nvars = 50; n = 200
# All parameter values equal one.
truth = numeric(2*nvars)+1
k = 1 # Scaling constant to make variance of base distribution = one
# Labels for the columns of the data file
namz = character(nvars)
for(j in 1:nvars) namz[j] = paste("d",as.character(j),sep='')
# colnames(bigdata) = namz
# With the cfa function, only need to specify the measurement model
mod = 'F =~ 1.0*d1+d2+d3+d4+d5+d6+d7+d8+d9+d10+
                d11+d12+d13+d14+d15+d16+d17+d18+d19+d20+
                d21+d22+d23+d24+d25+d26+d27+d28+d29+d30+
                d31+d32+d33+d34+d35+d36+d37+d38+d39+d40+
                d41+d42+d43+d44+d45+d46+d47+d48+d49+d50'

# Simulate a data set
set.seed(9999)
F = k*rnorm(n); e = k*rnorm(n*nvars); dim(e) = c(n,nvars)
simdat = F + e; colnames(simdat) = namz
# Fit the model
fit1 = cfa(mod,data=simdat)

p1 = parameterEstimates(fit1); p1
ci1 = p1[2:(2*nvars+1), 8:9] # Upper and lower confidence limits
hit1 = as.numeric(ci1[,1] < truth & truth < ci1[,2]) # Binary for in ci
cbind(ci1,truth,hit1)
```

The cute part is `simdat = F + e`. The object `F` (the factor) is an $n \times 1$ random vector, and `e` is an $n \times 50$ matrix of independent error terms. Because all the factor loadings equal one, row $i$ in the data file is obtained by adding $F_i$ to each element in row $i$ of the error matrix `e`. That's what the statement does.

As in all the simulations, this code is put in a simulation loop, along with a few more lines that generate sandwich and bootstrap confidence intervals. The first thing to note is that while one might anticipate numerical problems for such a large model, it turned out that there were almost none. Well, I did try $n = 50$ (half the number of parameters) and

it crashed, while $n = 100$ frequently produced estimates that were outside the parameter space. Starting with $n = 200$, everything was fine.

### Normal base distribution

It is a bit challenging to look at the results of a simulation, because there are so many parameters. In Table 5.35, $n = 200$ and the base distribution is normal. The column `sig` contains an asterisk ($*$) if any of the $z$ statistics exceeds the Bonferroni critical value of `3.76` for 300 tests.

Table 5.29: Coverage of 95% confidence intervals for the "Big Data" model (5.15), $n = 200$, Normal base distribution, 1,000 simulated data sets

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda2 | 0.960 | 0.958 | 0.953 | 1.451 | 1.161 | 0.435 | |
| lambda3 | 0.954 | 0.953 | 0.955 | 0.580 | 0.435 | 0.725 | |
| lambda4 | 0.944 | 0.941 | 0.945 | -0.871 | -1.306 | -0.725 | |
| lambda5 | 0.951 | 0.951 | 0.953 | 0.145 | 0.145 | 0.435 | |
| lambda6 | 0.955 | 0.949 | 0.944 | 0.725 | -0.145 | -0.871 | |
| lambda7 | 0.951 | 0.950 | 0.954 | 0.145 | 0.000 | 0.580 | |
| lambda8 | 0.955 | 0.951 | 0.953 | 0.725 | 0.145 | 0.435 | |
| lambda9 | 0.955 | 0.960 | 0.956 | 0.725 | 1.451 | 0.871 | |
| lambda10 | 0.947 | 0.943 | 0.948 | -0.435 | -1.016 | -0.290 | |
| lambda11 | 0.949 | 0.948 | 0.944 | -0.145 | -0.290 | -0.871 | |
| lambda12 | 0.955 | 0.951 | 0.951 | 0.725 | 0.145 | 0.145 | |
| lambda13 | 0.963 | 0.959 | 0.953 | 1.886 | 1.306 | 0.435 | |
| lambda14 | 0.947 | 0.944 | 0.950 | -0.435 | -0.871 | 0.000 | |
| lambda15 | 0.950 | 0.944 | 0.947 | 0.000 | -0.871 | -0.435 | |
| lambda16 | 0.941 | 0.934 | 0.937 | -1.306 | -2.322 | -1.886 | |
| lambda17 | 0.955 | 0.954 | 0.952 | 0.725 | 0.580 | 0.290 | |
| lambda18 | 0.940 | 0.937 | 0.935 | -1.451 | -1.886 | -2.176 | |
| lambda19 | 0.960 | 0.958 | 0.956 | 1.451 | 1.161 | 0.871 | |
| lambda20 | 0.950 | 0.950 | 0.950 | 0.000 | 0.000 | 0.000 | |
| lambda21 | 0.951 | 0.946 | 0.946 | 0.145 | -0.580 | -0.580 | |
| lambda22 | 0.960 | 0.948 | 0.950 | 1.451 | -0.290 | 0.000 | |
| lambda23 | 0.941 | 0.939 | 0.936 | -1.306 | -1.596 | -2.031 | |
| lambda24 | 0.938 | 0.936 | 0.946 | -1.741 | -2.031 | -0.580 | |
| lambda25 | 0.959 | 0.956 | 0.950 | 1.306 | 0.871 | 0.000 | |
| lambda26 | 0.953 | 0.944 | 0.946 | 0.435 | -0.871 | -0.580 | |
| lambda27 | 0.953 | 0.955 | 0.953 | 0.435 | 0.725 | 0.435 | |
| lambda28 | 0.944 | 0.944 | 0.943 | -0.871 | -0.871 | -1.016 | |
| lambda29 | 0.946 | 0.943 | 0.942 | -0.580 | -1.016 | -1.161 | |
| lambda30 | 0.952 | 0.946 | 0.954 | 0.290 | -0.580 | 0.580 | |
| lambda31 | 0.952 | 0.953 | 0.950 | 0.290 | 0.435 | 0.000 | |
| lambda32 | 0.953 | 0.948 | 0.951 | 0.435 | -0.290 | 0.145 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda33 | 0.949 | 0.948 | 0.952 | -0.145 | -0.290 | 0.290 | |
| lambda34 | 0.961 | 0.956 | 0.951 | 1.596 | 0.871 | 0.145 | |
| lambda35 | 0.959 | 0.954 | 0.954 | 1.306 | 0.580 | 0.580 | |
| lambda36 | 0.950 | 0.949 | 0.954 | 0.000 | -0.145 | 0.580 | |
| lambda37 | 0.951 | 0.950 | 0.947 | 0.145 | 0.000 | -0.435 | |
| lambda38 | 0.953 | 0.954 | 0.945 | 0.435 | 0.580 | -0.725 | |
| lambda39 | 0.950 | 0.946 | 0.952 | 0.000 | -0.580 | 0.290 | |
| lambda40 | 0.962 | 0.953 | 0.960 | 1.741 | 0.435 | 1.451 | |
| lambda41 | 0.945 | 0.939 | 0.942 | -0.725 | -1.596 | -1.161 | |
| lambda42 | 0.958 | 0.960 | 0.955 | 1.161 | 1.451 | 0.725 | |
| lambda43 | 0.960 | 0.957 | 0.955 | 1.451 | 1.016 | 0.725 | |
| lambda44 | 0.943 | 0.941 | 0.943 | -1.016 | -1.306 | -1.016 | |
| lambda45 | 0.956 | 0.949 | 0.952 | 0.871 | -0.145 | 0.290 | |
| lambda46 | 0.958 | 0.956 | 0.960 | 1.161 | 0.871 | 1.451 | |
| lambda47 | 0.946 | 0.942 | 0.951 | -0.580 | -1.161 | 0.145 | |
| lambda48 | 0.952 | 0.951 | 0.946 | 0.290 | 0.145 | -0.580 | |
| lambda49 | 0.961 | 0.960 | 0.955 | 1.596 | 1.451 | 0.725 | |
| lambda50 | 0.946 | 0.950 | 0.947 | -0.580 | 0.000 | -0.435 | |
| omega1 | 0.936 | 0.928 | 0.924 | -2.031 | -3.192 | -3.772 | * |
| omega2 | 0.946 | 0.940 | 0.937 | -0.580 | -1.451 | -1.886 | |
| omega3 | 0.933 | 0.930 | 0.928 | -2.467 | -2.902 | -3.192 | |
| omega4 | 0.944 | 0.942 | 0.944 | -0.871 | -1.161 | -0.871 | |
| omega5 | 0.935 | 0.929 | 0.928 | -2.176 | -3.047 | -3.192 | |
| omega6 | 0.934 | 0.924 | 0.922 | -2.322 | -3.772 | -4.063 | * |
| omega7 | 0.941 | 0.934 | 0.931 | -1.306 | -2.322 | -2.757 | |
| omega8 | 0.951 | 0.946 | 0.944 | 0.145 | -0.580 | -0.871 | |
| omega9 | 0.944 | 0.945 | 0.940 | -0.871 | -0.725 | -1.451 | |
| omega10 | 0.924 | 0.918 | 0.913 | -3.772 | -4.643 | -5.369 | * |
| omega11 | 0.935 | 0.934 | 0.935 | -2.176 | -2.322 | -2.176 | |
| omega12 | 0.954 | 0.947 | 0.948 | 0.580 | -0.435 | -0.290 | |
| omega13 | 0.939 | 0.929 | 0.930 | -1.596 | -3.047 | -2.902 | |
| omega14 | 0.944 | 0.938 | 0.938 | -0.871 | -1.741 | -1.741 | |
| omega15 | 0.939 | 0.934 | 0.938 | -1.596 | -2.322 | -1.741 | |
| omega16 | 0.929 | 0.924 | 0.928 | -3.047 | -3.772 | -3.192 | * |
| omega17 | 0.932 | 0.930 | 0.923 | -2.612 | -2.902 | -3.918 | * |
| omega18 | 0.936 | 0.938 | 0.935 | -2.031 | -1.741 | -2.176 | |
| omega19 | 0.936 | 0.928 | 0.931 | -2.031 | -3.192 | -2.757 | |
| omega20 | 0.923 | 0.918 | 0.919 | -3.918 | -4.643 | -4.498 | * |
| omega21 | 0.944 | 0.936 | 0.937 | -0.871 | -2.031 | -1.886 | |
| omega22 | 0.945 | 0.943 | 0.939 | -0.725 | -1.016 | -1.596 | |
| omega23 | 0.937 | 0.934 | 0.933 | -1.886 | -2.322 | -2.467 | |
| omega24 | 0.929 | 0.926 | 0.922 | -3.047 | -3.482 | -4.063 | * |
| omega25 | 0.956 | 0.947 | 0.943 | 0.871 | -0.435 | -1.016 | |
| omega26 | 0.937 | 0.928 | 0.930 | -1.886 | -3.192 | -2.902 | |
| omega27 | 0.934 | 0.933 | 0.929 | -2.322 | -2.467 | -3.047 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega28 | 0.929 | 0.931 | 0.932 | -3.047 | -2.757 | -2.612 | |
| omega29 | 0.940 | 0.934 | 0.934 | -1.451 | -2.322 | -2.322 | |
| omega30 | 0.929 | 0.920 | 0.923 | -3.047 | -4.353 | -3.918 | * |
| omega31 | 0.943 | 0.943 | 0.941 | -1.016 | -1.016 | -1.306 | |
| omega32 | 0.928 | 0.925 | 0.926 | -3.192 | -3.627 | -3.482 | |
| omega33 | 0.942 | 0.938 | 0.935 | -1.161 | -1.741 | -2.176 | |
| omega34 | 0.943 | 0.937 | 0.937 | -1.016 | -1.886 | -1.886 | |
| omega35 | 0.935 | 0.927 | 0.929 | -2.176 | -3.337 | -3.047 | |
| omega36 | 0.940 | 0.936 | 0.932 | -1.451 | -2.031 | -2.612 | |
| omega37 | 0.926 | 0.914 | 0.917 | -3.482 | -5.223 | -4.788 | * |
| omega38 | 0.942 | 0.938 | 0.940 | -1.161 | -1.741 | -1.451 | |
| omega39 | 0.935 | 0.931 | 0.931 | -2.176 | -2.757 | -2.757 | |
| omega40 | 0.940 | 0.938 | 0.936 | -1.451 | -1.741 | -2.031 | |
| omega41 | 0.934 | 0.929 | 0.928 | -2.322 | -3.047 | -3.192 | |
| omega42 | 0.943 | 0.937 | 0.936 | -1.016 | -1.886 | -2.031 | |
| omega43 | 0.947 | 0.938 | 0.935 | -0.435 | -1.741 | -2.176 | |
| omega44 | 0.926 | 0.926 | 0.930 | -3.482 | -3.482 | -2.902 | |
| omega45 | 0.937 | 0.935 | 0.934 | -1.886 | -2.176 | -2.322 | |
| omega46 | 0.938 | 0.935 | 0.932 | -1.741 | -2.176 | -2.612 | |
| omega47 | 0.944 | 0.940 | 0.933 | -0.871 | -1.451 | -2.467 | |
| omega48 | 0.918 | 0.919 | 0.917 | -4.643 | -4.498 | -4.788 | * |
| omega49 | 0.949 | 0.941 | 0.940 | -0.145 | -1.306 | -1.451 | |
| omega50 | 0.933 | 0.928 | 0.928 | -2.467 | -3.192 | -3.192 | |
| phi | 0.956 | 0.948 | 0.948 | 0.871 | -0.290 | -0.290 | |

* At least one $z$ statistic exceeds the Bonferroni critical value of 3.76 for 300 two-sided $z$-tests of $H_0$: Coverage = 0.95.

All the confidence intervals for the straight-arrow parameters $\lambda_2, \ldots, \lambda_{50}$ have acceptable coverage, while nine of the confidence intervals for the variance parameters fail the test, with coverage that is significantly lower than 0.95. Table 5.30 shows the same experiment for $n = 500$.

Table 5.30: Coverage of 95% confidence intervals for the "Big Data" model (5.15), $n = 500$, Normal base distribution, 1,000 simulated data sets

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda2 | 0.949 | 0.946 | 0.945 | -0.145 | -0.580 | -0.725 | |
| lambda3 | 0.934 | 0.932 | 0.926 | -2.322 | -2.612 | -3.482 | |
| lambda4 | 0.943 | 0.943 | 0.945 | -1.016 | -1.016 | -0.725 | |
| lambda5 | 0.940 | 0.941 | 0.940 | -1.451 | -1.306 | -1.451 | |
| lambda6 | 0.941 | 0.939 | 0.939 | -1.306 | -1.596 | -1.596 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda7 | 0.946 | 0.944 | 0.944 | -0.580 | -0.871 | -0.871 | |
| lambda8 | 0.949 | 0.943 | 0.942 | -0.145 | -1.016 | -1.161 | |
| lambda9 | 0.939 | 0.939 | 0.938 | -1.596 | -1.596 | -1.741 | |
| lambda10 | 0.945 | 0.947 | 0.947 | -0.725 | -0.435 | -0.435 | |
| lambda11 | 0.934 | 0.933 | 0.933 | -2.322 | -2.467 | -2.467 | |
| lambda12 | 0.947 | 0.950 | 0.939 | -0.435 | 0.000 | -1.596 | |
| lambda13 | 0.946 | 0.942 | 0.936 | -0.580 | -1.161 | -2.031 | |
| lambda14 | 0.940 | 0.937 | 0.934 | -1.451 | -1.886 | -2.322 | |
| lambda15 | 0.942 | 0.948 | 0.947 | -1.161 | -0.290 | -0.435 | |
| lambda16 | 0.932 | 0.934 | 0.939 | -2.612 | -2.322 | -1.596 | |
| lambda17 | 0.942 | 0.948 | 0.943 | -1.161 | -0.290 | -1.016 | |
| lambda18 | 0.944 | 0.944 | 0.943 | -0.871 | -0.871 | -1.016 | |
| lambda19 | 0.954 | 0.951 | 0.948 | 0.580 | 0.145 | -0.290 | |
| lambda20 | 0.936 | 0.933 | 0.934 | -2.031 | -2.467 | -2.322 | |
| lambda21 | 0.942 | 0.942 | 0.937 | -1.161 | -1.161 | -1.886 | |
| lambda22 | 0.940 | 0.936 | 0.937 | -1.451 | -2.031 | -1.886 | |
| lambda23 | 0.945 | 0.945 | 0.945 | -0.725 | -0.725 | -0.725 | |
| lambda24 | 0.954 | 0.951 | 0.951 | 0.580 | 0.145 | 0.145 | |
| lambda25 | 0.948 | 0.946 | 0.950 | -0.290 | -0.580 | 0.000 | |
| lambda26 | 0.942 | 0.942 | 0.939 | -1.161 | -1.161 | -1.596 | |
| lambda27 | 0.953 | 0.951 | 0.955 | 0.435 | 0.145 | 0.725 | |
| lambda28 | 0.956 | 0.956 | 0.954 | 0.871 | 0.871 | 0.580 | |
| lambda29 | 0.955 | 0.945 | 0.949 | 0.725 | -0.725 | -0.145 | |
| lambda30 | 0.948 | 0.947 | 0.950 | -0.290 | -0.435 | 0.000 | |
| lambda31 | 0.935 | 0.934 | 0.936 | -2.176 | -2.322 | -2.031 | |
| lambda32 | 0.946 | 0.940 | 0.943 | -0.580 | -1.451 | -1.016 | |
| lambda33 | 0.937 | 0.936 | 0.930 | -1.886 | -2.031 | -2.902 | |
| lambda34 | 0.951 | 0.949 | 0.945 | 0.145 | -0.145 | -0.725 | |
| lambda35 | 0.943 | 0.942 | 0.944 | -1.016 | -1.161 | -0.871 | |
| lambda36 | 0.941 | 0.938 | 0.941 | -1.306 | -1.741 | -1.306 | |
| lambda37 | 0.955 | 0.954 | 0.955 | 0.725 | 0.580 | 0.725 | |
| lambda38 | 0.949 | 0.946 | 0.959 | -0.145 | -0.580 | 1.306 | |
| lambda39 | 0.953 | 0.953 | 0.954 | 0.435 | 0.435 | 0.580 | |
| lambda40 | 0.934 | 0.938 | 0.934 | -2.322 | -1.741 | -2.322 | |
| lambda41 | 0.948 | 0.947 | 0.945 | -0.290 | -0.435 | -0.725 | |
| lambda42 | 0.949 | 0.949 | 0.945 | -0.145 | -0.145 | -0.725 | |
| lambda43 | 0.945 | 0.941 | 0.944 | -0.725 | -1.306 | -0.871 | |
| lambda44 | 0.959 | 0.958 | 0.955 | 1.306 | 1.161 | 0.725 | |
| lambda45 | 0.946 | 0.941 | 0.947 | -0.580 | -1.306 | -0.435 | |
| lambda46 | 0.956 | 0.949 | 0.954 | 0.871 | -0.145 | 0.580 | |
| lambda47 | 0.947 | 0.942 | 0.942 | -0.435 | -1.161 | -1.161 | |
| lambda48 | 0.939 | 0.939 | 0.936 | -1.596 | -1.596 | -2.031 | |
| lambda49 | 0.944 | 0.940 | 0.936 | -0.871 | -1.451 | -2.031 | |
| lambda50 | 0.950 | 0.946 | 0.961 | 0.000 | -0.580 | 1.596 | |
| omega1 | 0.940 | 0.938 | 0.936 | -1.451 | -1.741 | -2.031 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega2 | 0.941 | 0.937 | 0.937 | -1.306 | -1.886 | -1.886 | |
| omega3 | 0.937 | 0.939 | 0.937 | -1.886 | -1.596 | -1.886 | |
| omega4 | 0.950 | 0.947 | 0.946 | 0.000 | -0.435 | -0.580 | |
| omega5 | 0.928 | 0.929 | 0.933 | -3.192 | -3.047 | -2.467 | |
| omega6 | 0.946 | 0.942 | 0.941 | -0.580 | -1.161 | -1.306 | |
| omega7 | 0.945 | 0.945 | 0.941 | -0.725 | -0.725 | -1.306 | |
| omega8 | 0.939 | 0.936 | 0.940 | -1.596 | -2.031 | -1.451 | |
| omega9 | 0.949 | 0.945 | 0.943 | -0.145 | -0.725 | -1.016 | |
| omega10 | 0.955 | 0.948 | 0.951 | 0.725 | -0.290 | 0.145 | |
| omega11 | 0.953 | 0.950 | 0.956 | 0.435 | 0.000 | 0.871 | |
| omega12 | 0.948 | 0.949 | 0.947 | -0.290 | -0.145 | -0.435 | |
| omega13 | 0.940 | 0.934 | 0.935 | -1.451 | -2.322 | -2.176 | |
| omega14 | 0.942 | 0.942 | 0.941 | -1.161 | -1.161 | -1.306 | |
| omega15 | 0.942 | 0.938 | 0.933 | -1.161 | -1.741 | -2.467 | |
| omega16 | 0.954 | 0.954 | 0.951 | 0.580 | 0.580 | 0.145 | |
| omega17 | 0.936 | 0.939 | 0.934 | -2.031 | -1.596 | -2.322 | |
| omega18 | 0.941 | 0.939 | 0.944 | -1.306 | -1.596 | -0.871 | |
| omega19 | 0.953 | 0.947 | 0.947 | 0.435 | -0.435 | -0.435 | |
| omega20 | 0.941 | 0.941 | 0.937 | -1.306 | -1.306 | -1.886 | |
| omega21 | 0.940 | 0.940 | 0.937 | -1.451 | -1.451 | -1.886 | |
| omega22 | 0.950 | 0.955 | 0.955 | 0.000 | 0.725 | 0.725 | |
| omega23 | 0.943 | 0.942 | 0.943 | -1.016 | -1.161 | -1.016 | |
| omega24 | 0.946 | 0.944 | 0.947 | -0.580 | -0.871 | -0.435 | |
| omega25 | 0.943 | 0.945 | 0.943 | -1.016 | -0.725 | -1.016 | |
| omega26 | 0.945 | 0.944 | 0.941 | -0.725 | -0.871 | -1.306 | |
| omega27 | 0.957 | 0.956 | 0.951 | 1.016 | 0.871 | 0.145 | |
| omega28 | 0.941 | 0.941 | 0.950 | -1.306 | -1.306 | 0.000 | |
| omega29 | 0.922 | 0.921 | 0.921 | -4.063 | -4.208 | -4.208 | * |
| omega30 | 0.949 | 0.944 | 0.940 | -0.145 | -0.871 | -1.451 | |
| omega31 | 0.953 | 0.953 | 0.950 | 0.435 | 0.435 | 0.000 | |
| omega32 | 0.955 | 0.949 | 0.949 | 0.725 | -0.145 | -0.145 | |
| omega33 | 0.957 | 0.948 | 0.943 | 1.016 | -0.290 | -1.016 | |
| omega34 | 0.930 | 0.929 | 0.928 | -2.902 | -3.047 | -3.192 | |
| omega35 | 0.946 | 0.945 | 0.944 | -0.580 | -0.725 | -0.871 | |
| omega36 | 0.936 | 0.937 | 0.938 | -2.031 | -1.886 | -1.741 | |
| omega37 | 0.940 | 0.937 | 0.937 | -1.451 | -1.886 | -1.886 | |
| omega38 | 0.948 | 0.949 | 0.946 | -0.290 | -0.145 | -0.580 | |
| omega39 | 0.952 | 0.949 | 0.948 | 0.290 | -0.145 | -0.290 | |
| omega40 | 0.940 | 0.938 | 0.939 | -1.451 | -1.741 | -1.596 | |
| omega41 | 0.961 | 0.955 | 0.955 | 1.596 | 0.725 | 0.725 | |
| omega42 | 0.934 | 0.929 | 0.929 | -2.322 | -3.047 | -3.047 | |
| omega43 | 0.948 | 0.943 | 0.943 | -0.290 | -1.016 | -1.016 | |
| omega44 | 0.951 | 0.947 | 0.948 | 0.145 | -0.435 | -0.290 | |
| omega45 | 0.929 | 0.930 | 0.927 | -3.047 | -2.902 | -3.337 | |
| omega46 | 0.938 | 0.934 | 0.937 | -1.741 | -2.322 | -1.886 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega47 | 0.938 | 0.936 | 0.936 | -1.741 | -2.031 | -2.031 | |
| omega48 | 0.953 | 0.946 | 0.946 | 0.435 | -0.580 | -0.580 | |
| omega49 | 0.943 | 0.940 | 0.940 | -1.016 | -1.451 | -1.451 | |
| omega50 | 0.924 | 0.923 | 0.923 | -3.772 | -3.918 | -3.918 | * |
| phi | 0.950 | 0.949 | 0.953 | 0.000 | -0.145 | 0.435 | |

* At least one $z$ statistic exceeds the Bonferroni critical value of 3.76 for 300 two-sided $z$-tests of $H_0$: Coverage = 0.95.

This time, coverage for the $\lambda_j$ factor loadings is okay again, and only two of the variance parameters suffer from significant under-coverage. It is quite clear that all methods are working acceptably for normal data, with a sample size of $n = 500$ or perhaps a bit above required for really excellent performance.

### Exponential base distribution

Table 5.31 shows results for $n = 200$ with the heavy-tailed exponential distribution. Looking at the last column (labelled Sig) observe the strong support for the Satorra-Bentler principle. Problems are indicated for only two of the 48 straight-arrow factor loadings, and in both cases, the culprit is under-coverage by the sandwich interval. The normal-theory (and bootstrap) intervals are okay in every case. For the variance parameters, all three methods fail in every case, but coverage of the normal theory intervals is much worse.

Table 5.31: Coverage of 95% confidence intervals for the "Big Data" model (5.15), $n = 200$, Exponential base distribution, 1,000 simulated data sets

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda2 | 0.952 | 0.951 | 0.945 | 0.290 | 0.145 | -0.725 | |
| lambda3 | 0.954 | 0.937 | 0.940 | 0.580 | -1.886 | -1.451 | |
| lambda4 | 0.947 | 0.936 | 0.942 | -0.435 | -2.031 | -1.161 | |
| lambda5 | 0.952 | 0.946 | 0.947 | 0.290 | -0.580 | -0.435 | |
| lambda6 | 0.953 | 0.946 | 0.944 | 0.435 | -0.580 | -0.871 | |
| lambda7 | 0.945 | 0.931 | 0.940 | -0.725 | -2.757 | -1.451 | |
| lambda8 | 0.947 | 0.937 | 0.933 | -0.435 | -1.886 | -2.467 | |
| lambda9 | 0.947 | 0.938 | 0.941 | -0.435 | -1.741 | -1.306 | |
| lambda10 | 0.945 | 0.930 | 0.934 | -0.725 | -2.902 | -2.322 | |
| lambda11 | 0.961 | 0.944 | 0.944 | 1.596 | -0.871 | -0.871 | |
| lambda12 | 0.945 | 0.933 | 0.937 | -0.725 | -2.467 | -1.886 | |
| lambda13 | 0.952 | 0.940 | 0.945 | 0.290 | -1.451 | -0.725 | |
| lambda14 | 0.951 | 0.933 | 0.945 | 0.145 | -2.467 | -0.725 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda15 | 0.944 | 0.934 | 0.936 | -0.871 | -2.322 | -2.031 | |
| lambda16 | 0.947 | 0.935 | 0.937 | -0.435 | -2.176 | -1.886 | |
| lambda17 | 0.946 | 0.938 | 0.945 | -0.580 | -1.741 | -0.725 | |
| lambda18 | 0.947 | 0.942 | 0.937 | -0.435 | -1.161 | -1.886 | |
| lambda19 | 0.949 | 0.934 | 0.935 | -0.145 | -2.322 | -2.176 | |
| lambda20 | 0.947 | 0.933 | 0.945 | -0.435 | -2.467 | -0.725 | |
| lambda21 | 0.955 | 0.939 | 0.941 | 0.725 | -1.596 | -1.306 | |
| lambda22 | 0.942 | 0.930 | 0.939 | -1.161 | -2.902 | -1.596 | |
| lambda23 | 0.946 | 0.932 | 0.942 | -0.580 | -2.612 | -1.161 | |
| lambda24 | 0.943 | 0.928 | 0.939 | -1.016 | -3.192 | -1.596 | |
| lambda25 | 0.946 | 0.936 | 0.946 | -0.580 | -2.031 | -0.580 | |
| lambda26 | 0.946 | 0.927 | 0.933 | -0.580 | -3.337 | -2.467 | |
| lambda27 | 0.955 | 0.943 | 0.942 | 0.725 | -1.016 | -1.161 | |
| lambda28 | 0.945 | 0.927 | 0.933 | -0.725 | -3.337 | -2.467 | |
| lambda29 | 0.946 | 0.939 | 0.943 | -0.580 | -1.596 | -1.016 | |
| lambda30 | 0.943 | 0.918 | 0.930 | -1.016 | -4.643 | -2.902 | * |
| lambda31 | 0.943 | 0.931 | 0.934 | -1.016 | -2.757 | -2.322 | |
| lambda32 | 0.950 | 0.941 | 0.934 | 0.000 | -1.306 | -2.322 | |
| lambda33 | 0.946 | 0.927 | 0.932 | -0.580 | -3.337 | -2.612 | |
| lambda34 | 0.937 | 0.930 | 0.935 | -1.886 | -2.902 | -2.176 | |
| lambda35 | 0.948 | 0.929 | 0.927 | -0.290 | -3.047 | -3.337 | |
| lambda36 | 0.950 | 0.935 | 0.946 | 0.000 | -2.176 | -0.580 | |
| lambda37 | 0.945 | 0.935 | 0.945 | -0.725 | -2.176 | -0.725 | |
| lambda38 | 0.948 | 0.944 | 0.947 | -0.290 | -0.871 | -0.435 | |
| lambda39 | 0.946 | 0.936 | 0.944 | -0.580 | -2.031 | -0.871 | |
| lambda40 | 0.948 | 0.931 | 0.942 | -0.290 | -2.757 | -1.161 | |
| lambda41 | 0.942 | 0.931 | 0.940 | -1.161 | -2.757 | -1.451 | |
| lambda42 | 0.941 | 0.924 | 0.932 | -1.306 | -3.772 | -2.612 | * |
| lambda43 | 0.955 | 0.939 | 0.945 | 0.725 | -1.596 | -0.725 | |
| lambda44 | 0.944 | 0.942 | 0.948 | -0.871 | -1.161 | -0.290 | |
| lambda45 | 0.939 | 0.925 | 0.932 | -1.596 | -3.627 | -2.612 | |
| lambda46 | 0.958 | 0.946 | 0.951 | 1.161 | -0.580 | 0.145 | |
| lambda47 | 0.949 | 0.931 | 0.936 | -0.145 | -2.757 | -2.031 | |
| lambda48 | 0.956 | 0.943 | 0.942 | 0.871 | -1.016 | -1.161 | |
| lambda49 | 0.952 | 0.939 | 0.940 | 0.290 | -1.596 | -1.451 | |
| lambda50 | 0.954 | 0.937 | 0.942 | 0.580 | -1.886 | -1.161 | |
| omega1 | 0.664 | 0.887 | 0.886 | -41.497 | -9.141 | -9.286 | * |
| omega2 | 0.679 | 0.889 | 0.892 | -39.321 | -8.851 | -8.416 | * |
| omega3 | 0.691 | 0.907 | 0.908 | -37.580 | -6.239 | -6.094 | * |
| omega4 | 0.685 | 0.887 | 0.891 | -38.450 | -9.141 | -8.561 | * |
| omega5 | 0.693 | 0.881 | 0.883 | -37.289 | -10.012 | -9.721 | * |
| omega6 | 0.645 | 0.869 | 0.872 | -44.254 | -11.753 | -11.317 | * |
| omega7 | 0.657 | 0.862 | 0.868 | -42.513 | -12.768 | -11.898 | * |
| omega8 | 0.676 | 0.884 | 0.880 | -39.756 | -9.576 | -10.157 | * |
| omega9 | 0.678 | 0.894 | 0.898 | -39.466 | -8.125 | -7.545 | * |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega10 | 0.657 | 0.878 | 0.879 | -42.513 | -10.447 | -10.302 | * |
| omega11 | 0.712 | 0.907 | 0.906 | -34.533 | -6.239 | -6.384 | * |
| omega12 | 0.684 | 0.882 | 0.887 | -38.595 | -9.866 | -9.141 | * |
| omega13 | 0.684 | 0.886 | 0.886 | -38.595 | -9.286 | -9.286 | * |
| omega14 | 0.681 | 0.880 | 0.886 | -39.031 | -10.157 | -9.286 | * |
| omega15 | 0.685 | 0.884 | 0.885 | -38.450 | -9.576 | -9.431 | * |
| omega16 | 0.714 | 0.875 | 0.881 | -34.242 | -10.882 | -10.012 | * |
| omega17 | 0.645 | 0.874 | 0.875 | -44.254 | -11.027 | -10.882 | * |
| omega18 | 0.672 | 0.873 | 0.875 | -40.336 | -11.172 | -10.882 | * |
| omega19 | 0.668 | 0.865 | 0.874 | -40.917 | -12.333 | -11.027 | * |
| omega20 | 0.701 | 0.890 | 0.894 | -36.129 | -8.706 | -8.125 | * |
| omega21 | 0.671 | 0.893 | 0.898 | -40.482 | -8.270 | -7.545 | * |
| omega22 | 0.664 | 0.870 | 0.873 | -41.497 | -11.608 | -11.172 | * |
| omega23 | 0.679 | 0.879 | 0.882 | -39.321 | -10.302 | -9.866 | * |
| omega24 | 0.669 | 0.881 | 0.882 | -40.772 | -10.012 | -9.866 | * |
| omega25 | 0.660 | 0.885 | 0.885 | -42.078 | -9.431 | -9.431 | * |
| omega26 | 0.694 | 0.892 | 0.900 | -37.144 | -8.416 | -7.255 | * |
| omega27 | 0.677 | 0.874 | 0.883 | -39.611 | -11.027 | -9.721 | * |
| omega28 | 0.663 | 0.864 | 0.872 | -41.642 | -12.478 | -11.317 | * |
| omega29 | 0.666 | 0.869 | 0.871 | -41.207 | -11.753 | -11.463 | * |
| omega30 | 0.695 | 0.887 | 0.893 | -36.999 | -9.141 | -8.270 | * |
| omega31 | 0.683 | 0.868 | 0.872 | -38.740 | -11.898 | -11.317 | * |
| omega32 | 0.671 | 0.864 | 0.867 | -40.482 | -12.478 | -12.043 | * |
| omega33 | 0.683 | 0.894 | 0.896 | -38.740 | -8.125 | -7.835 | * |
| omega34 | 0.669 | 0.885 | 0.888 | -40.772 | -9.431 | -8.996 | * |
| omega35 | 0.665 | 0.891 | 0.893 | -41.352 | -8.561 | -8.270 | * |
| omega36 | 0.670 | 0.879 | 0.882 | -40.627 | -10.302 | -9.866 | * |
| omega37 | 0.647 | 0.880 | 0.881 | -43.964 | -10.157 | -10.012 | * |
| omega38 | 0.694 | 0.895 | 0.894 | -37.144 | -7.980 | -8.125 | * |
| omega39 | 0.677 | 0.885 | 0.892 | -39.611 | -9.431 | -8.416 | * |
| omega40 | 0.690 | 0.875 | 0.880 | -37.725 | -10.882 | -10.157 | * |
| omega41 | 0.681 | 0.888 | 0.892 | -39.031 | -8.996 | -8.416 | * |
| omega42 | 0.703 | 0.875 | 0.880 | -35.839 | -10.882 | -10.157 | * |
| omega43 | 0.691 | 0.878 | 0.883 | -37.580 | -10.447 | -9.721 | * |
| omega44 | 0.666 | 0.882 | 0.887 | -41.207 | -9.866 | -9.141 | * |
| omega45 | 0.676 | 0.898 | 0.902 | -39.756 | -7.545 | -6.965 | * |
| omega46 | 0.672 | 0.872 | 0.878 | -40.336 | -11.317 | -10.447 | * |
| omega47 | 0.649 | 0.881 | 0.883 | -43.674 | -10.012 | -9.721 | * |
| omega48 | 0.691 | 0.894 | 0.892 | -37.580 | -8.125 | -8.416 | * |
| omega49 | 0.702 | 0.884 | 0.886 | -35.984 | -9.576 | -9.286 | * |
| omega50 | 0.679 | 0.885 | 0.887 | -39.321 | -9.431 | -9.141 | * |
| phi | 0.842 | 0.909 | 0.912 | -15.670 | -5.949 | -5.514 | * |

* At least one $z$ statistic exceeds the Bonferroni critical value of 3.76 for 300 two-sided $z$-tests of

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |

$H_0$: Coverage = 0.95.

The normal theory confidence intervals for the variance parameters are doomed, but we anticipate that a larger sample size will help for the sandwich and Bootstrap intervals. The numbers in Table 5.31 are comparable to those in Table 5.3 for the "Extra response variables" model, which is really just a smaller version of this one. It took $n = 1,000$ to achieve good performance for the little extra variables model. For the Big Data model, I tried $n = 500$ with no bootstrap, and the sandwich was better but still inadequate. With $n = 1,000$, twelve of the sandwich $z$ statistics exceeded the Bonferroni critical value. That's still too many. In an experiment with $n = 1,500$, only two of the sandwich $z$ statistics were greater than the Bonferroni critical value. Encouraged, I tried the complete job (including bootstrap) for $n = 1,500$. The results are shown in Table 5.32.

Table 5.32: Coverage of 95% confidence intervals for the "Big Data" model (5.15), $n = 1,500$, Exponential base distribution, 1,000 simulated data sets

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda2 | 0.948 | 0.949 | 0.948 | -0.290 | -0.145 | -0.290 | |
| lambda3 | 0.950 | 0.948 | 0.949 | 0.000 | -0.290 | -0.145 | |
| lambda4 | 0.955 | 0.951 | 0.951 | 0.725 | 0.145 | 0.145 | |
| lambda5 | 0.939 | 0.939 | 0.939 | -1.596 | -1.596 | -1.596 | |
| lambda6 | 0.946 | 0.946 | 0.946 | -0.580 | -0.580 | -0.580 | |
| lambda7 | 0.955 | 0.952 | 0.956 | 0.725 | 0.290 | 0.871 | |
| lambda8 | 0.943 | 0.944 | 0.944 | -1.016 | -0.871 | -0.871 | |
| lambda9 | 0.960 | 0.955 | 0.958 | 1.451 | 0.725 | 1.161 | |
| lambda10 | 0.947 | 0.944 | 0.944 | -0.435 | -0.871 | -0.871 | |
| lambda11 | 0.934 | 0.931 | 0.939 | -2.322 | -2.757 | -1.596 | |
| lambda12 | 0.945 | 0.945 | 0.939 | -0.725 | -0.725 | -1.596 | |
| lambda13 | 0.946 | 0.945 | 0.947 | -0.580 | -0.725 | -0.435 | |
| lambda14 | 0.950 | 0.944 | 0.950 | 0.000 | -0.871 | 0.000 | |
| lambda15 | 0.954 | 0.951 | 0.951 | 0.580 | 0.145 | 0.145 | |
| lambda16 | 0.954 | 0.952 | 0.954 | 0.580 | 0.290 | 0.580 | |
| lambda17 | 0.941 | 0.937 | 0.937 | -1.306 | -1.886 | -1.886 | |
| lambda18 | 0.943 | 0.937 | 0.941 | -1.016 | -1.886 | -1.306 | |
| lambda19 | 0.955 | 0.949 | 0.946 | 0.725 | -0.145 | -0.580 | |
| lambda20 | 0.950 | 0.950 | 0.958 | 0.000 | 0.000 | 1.161 | |
| lambda21 | 0.948 | 0.951 | 0.949 | -0.290 | 0.145 | -0.145 | |
| lambda22 | 0.968 | 0.968 | 0.961 | 2.612 | 2.612 | 1.596 | |
| lambda23 | 0.944 | 0.951 | 0.943 | -0.871 | 0.145 | -1.016 | |
| lambda24 | 0.933 | 0.933 | 0.933 | -2.467 | -2.467 | -2.467 | |
| lambda25 | 0.950 | 0.956 | 0.954 | 0.000 | 0.871 | 0.580 | |
| lambda26 | 0.936 | 0.935 | 0.940 | -2.031 | -2.176 | -1.451 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda27 | 0.954 | 0.954 | 0.957 | 0.580 | 0.580 | 1.016 | |
| lambda28 | 0.946 | 0.948 | 0.947 | -0.580 | -0.290 | -0.435 | |
| lambda29 | 0.948 | 0.945 | 0.944 | -0.290 | -0.725 | -0.871 | |
| lambda30 | 0.946 | 0.940 | 0.943 | -0.580 | -1.451 | -1.016 | |
| lambda31 | 0.953 | 0.953 | 0.951 | 0.435 | 0.435 | 0.145 | |
| lambda32 | 0.944 | 0.946 | 0.942 | -0.871 | -0.580 | -1.161 | |
| lambda33 | 0.946 | 0.948 | 0.950 | -0.580 | -0.290 | 0.000 | |
| lambda34 | 0.945 | 0.952 | 0.948 | -0.725 | 0.290 | -0.290 | |
| lambda35 | 0.945 | 0.940 | 0.942 | -0.725 | -1.451 | -1.161 | |
| lambda36 | 0.947 | 0.940 | 0.940 | -0.435 | -1.451 | -1.451 | |
| lambda37 | 0.946 | 0.936 | 0.936 | -0.580 | -2.031 | -2.031 | |
| lambda38 | 0.940 | 0.938 | 0.934 | -1.451 | -1.741 | -2.322 | |
| lambda39 | 0.941 | 0.935 | 0.938 | -1.306 | -2.176 | -1.741 | |
| lambda40 | 0.934 | 0.934 | 0.934 | -2.322 | -2.322 | -2.322 | |
| lambda41 | 0.949 | 0.945 | 0.950 | -0.145 | -0.725 | 0.000 | |
| lambda42 | 0.944 | 0.941 | 0.943 | -0.871 | -1.306 | -1.016 | |
| lambda43 | 0.949 | 0.949 | 0.950 | -0.145 | -0.145 | 0.000 | |
| lambda44 | 0.954 | 0.949 | 0.946 | 0.580 | -0.145 | -0.580 | |
| lambda45 | 0.956 | 0.949 | 0.951 | 0.871 | -0.145 | 0.145 | |
| lambda46 | 0.950 | 0.952 | 0.948 | 0.000 | 0.290 | -0.290 | |
| lambda47 | 0.951 | 0.946 | 0.948 | 0.145 | -0.580 | -0.290 | |
| lambda48 | 0.934 | 0.931 | 0.934 | -2.322 | -2.757 | -2.322 | |
| lambda49 | 0.943 | 0.944 | 0.944 | -1.016 | -0.871 | -0.871 | |
| lambda50 | 0.956 | 0.954 | 0.956 | 0.871 | 0.580 | 0.871 | |
| omega1 | 0.687 | 0.934 | 0.938 | -38.160 | -2.322 | -1.741 | * |
| omega2 | 0.668 | 0.939 | 0.938 | -40.917 | -1.596 | -1.741 | * |
| omega3 | 0.684 | 0.937 | 0.938 | -38.595 | -1.886 | -1.741 | * |
| omega4 | 0.697 | 0.931 | 0.928 | -36.709 | -2.757 | -3.192 | * |
| omega5 | 0.696 | 0.945 | 0.946 | -36.854 | -0.725 | -0.580 | * |
| omega6 | 0.683 | 0.943 | 0.942 | -38.740 | -1.016 | -1.161 | * |
| omega7 | 0.691 | 0.940 | 0.941 | -37.580 | -1.451 | -1.306 | * |
| omega8 | 0.655 | 0.933 | 0.942 | -42.803 | -2.467 | -1.161 | * |
| omega9 | 0.659 | 0.921 | 0.921 | -42.223 | -4.208 | -4.208 | * |
| omega10 | 0.683 | 0.942 | 0.937 | -38.740 | -1.161 | -1.886 | * |
| omega11 | 0.709 | 0.949 | 0.948 | -34.968 | -0.145 | -0.290 | * |
| omega12 | 0.689 | 0.928 | 0.929 | -37.870 | -3.192 | -3.047 | * |
| omega13 | 0.663 | 0.924 | 0.928 | -41.642 | -3.772 | -3.192 | * |
| omega14 | 0.695 | 0.943 | 0.942 | -36.999 | -1.016 | -1.161 | * |
| omega15 | 0.686 | 0.931 | 0.934 | -38.305 | -2.757 | -2.322 | * |
| omega16 | 0.671 | 0.939 | 0.946 | -40.482 | -1.596 | -0.580 | * |
| omega17 | 0.667 | 0.925 | 0.927 | -41.062 | -3.627 | -3.337 | * |
| omega18 | 0.701 | 0.940 | 0.943 | -36.129 | -1.451 | -1.016 | * |
| omega19 | 0.661 | 0.935 | 0.934 | -41.933 | -2.176 | -2.322 | * |
| omega20 | 0.681 | 0.928 | 0.925 | -39.031 | -3.192 | -3.627 | * |
| omega21 | 0.686 | 0.951 | 0.952 | -38.305 | 0.145 | 0.290 | * |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega22 | 0.682 | 0.947 | 0.949 | -38.886 | -0.435 | -0.145 | * |
| omega23 | 0.691 | 0.946 | 0.943 | -37.580 | -0.580 | -1.016 | * |
| omega24 | 0.683 | 0.947 | 0.946 | -38.740 | -0.435 | -0.580 | * |
| omega25 | 0.666 | 0.932 | 0.932 | -41.207 | -2.612 | -2.612 | * |
| omega26 | 0.659 | 0.921 | 0.920 | -42.223 | -4.208 | -4.353 | * |
| omega27 | 0.671 | 0.929 | 0.929 | -40.482 | -3.047 | -3.047 | * |
| omega28 | 0.704 | 0.959 | 0.957 | -35.693 | 1.306 | 1.016 | * |
| omega29 | 0.688 | 0.939 | 0.937 | -38.015 | -1.596 | -1.886 | * |
| omega30 | 0.694 | 0.942 | 0.942 | -37.144 | -1.161 | -1.161 | * |
| omega31 | 0.682 | 0.938 | 0.938 | -38.886 | -1.741 | -1.741 | * |
| omega32 | 0.698 | 0.945 | 0.946 | -36.564 | -0.725 | -0.580 | * |
| omega33 | 0.688 | 0.940 | 0.947 | -38.015 | -1.451 | -0.435 | * |
| omega34 | 0.687 | 0.932 | 0.933 | -38.160 | -2.612 | -2.467 | * |
| omega35 | 0.682 | 0.932 | 0.934 | -38.886 | -2.612 | -2.322 | * |
| omega36 | 0.679 | 0.941 | 0.943 | -39.321 | -1.306 | -1.016 | * |
| omega37 | 0.687 | 0.923 | 0.925 | -38.160 | -3.918 | -3.627 | * |
| omega38 | 0.689 | 0.937 | 0.940 | -37.870 | -1.886 | -1.451 | * |
| omega39 | 0.702 | 0.936 | 0.935 | -35.984 | -2.031 | -2.176 | * |
| omega40 | 0.695 | 0.941 | 0.944 | -36.999 | -1.306 | -0.871 | * |
| omega41 | 0.684 | 0.937 | 0.935 | -38.595 | -1.886 | -2.176 | * |
| omega42 | 0.689 | 0.945 | 0.943 | -37.870 | -0.725 | -1.016 | * |
| omega43 | 0.680 | 0.931 | 0.935 | -39.176 | -2.757 | -2.176 | * |
| omega44 | 0.662 | 0.933 | 0.933 | -41.787 | -2.467 | -2.467 | * |
| omega45 | 0.663 | 0.938 | 0.938 | -41.642 | -1.741 | -1.741 | * |
| omega46 | 0.669 | 0.923 | 0.928 | -40.772 | -3.918 | -3.192 | * |
| omega47 | 0.688 | 0.928 | 0.929 | -38.015 | -3.192 | -3.047 | * |
| omega48 | 0.689 | 0.939 | 0.935 | -37.870 | -1.596 | -2.176 | * |
| omega49 | 0.670 | 0.941 | 0.941 | -40.627 | -1.306 | -1.306 | * |
| omega50 | 0.704 | 0.943 | 0.945 | -35.693 | -1.016 | -0.725 | * |
| phi | 0.822 | 0.934 | 0.933 | -18.572 | -2.322 | -2.467 | * |

* At least one $z$ statistic exceeds the Bonferroni critical value of 3.76 for 300 two-sided $z$-tests of $H_0$: Coverage = 0.95.

In Table 5.32, all of the variance parameters are flagged with an $*$, because of the huge $z$ values for the normal theory intervals. This obscures the fact that we are not there yet. The sample size is still not big enough. I tried $n = 3,000$ with no bootstrap. The sandwich had good coverage for all parameters, so presumably $n = 3,000$ was too much. In a trial no-bootstrap run with $n = 2,000$, all the sandwich intervals had adequate coverage. So, I ran the full job including the bootstrap. The results are shown in Table 5.33.

Table 5.33: Coverage of 95% confidence intervals for the "Big Data" model (5.15), $n = 2,000$, Exponential base distribution, 1,000 simulated data sets

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda2 | 0.942 | 0.940 | 0.935 | -1.161 | -1.451 | -2.176 | |
| lambda3 | 0.937 | 0.934 | 0.938 | -1.886 | -2.322 | -1.741 | |
| lambda4 | 0.960 | 0.962 | 0.961 | 1.451 | 1.741 | 1.596 | |
| lambda5 | 0.951 | 0.946 | 0.946 | 0.145 | -0.580 | -0.580 | |
| lambda6 | 0.944 | 0.943 | 0.940 | -0.871 | -1.016 | -1.451 | |
| lambda7 | 0.950 | 0.950 | 0.947 | 0.000 | 0.000 | -0.435 | |
| lambda8 | 0.958 | 0.950 | 0.948 | 1.161 | 0.000 | -0.290 | |
| lambda9 | 0.959 | 0.960 | 0.964 | 1.306 | 1.451 | 2.031 | |
| lambda10 | 0.954 | 0.947 | 0.948 | 0.580 | -0.435 | -0.290 | |
| lambda11 | 0.950 | 0.947 | 0.946 | 0.000 | -0.435 | -0.580 | |
| lambda12 | 0.946 | 0.943 | 0.947 | -0.580 | -1.016 | -0.435 | |
| lambda13 | 0.941 | 0.936 | 0.934 | -1.306 | -2.031 | -2.322 | |
| lambda14 | 0.958 | 0.951 | 0.960 | 1.161 | 0.145 | 1.451 | |
| lambda15 | 0.942 | 0.943 | 0.939 | -1.161 | -1.016 | -1.596 | |
| lambda16 | 0.950 | 0.949 | 0.949 | 0.000 | -0.145 | -0.145 | |
| lambda17 | 0.944 | 0.943 | 0.941 | -0.871 | -1.016 | -1.306 | |
| lambda18 | 0.949 | 0.950 | 0.951 | -0.145 | 0.000 | 0.145 | |
| lambda19 | 0.947 | 0.946 | 0.948 | -0.435 | -0.580 | -0.290 | |
| lambda20 | 0.963 | 0.957 | 0.962 | 1.886 | 1.016 | 1.741 | |
| lambda21 | 0.955 | 0.950 | 0.951 | 0.725 | 0.000 | 0.145 | |
| lambda22 | 0.954 | 0.951 | 0.950 | 0.580 | 0.145 | 0.000 | |
| lambda23 | 0.945 | 0.940 | 0.944 | -0.725 | -1.451 | -0.871 | |
| lambda24 | 0.946 | 0.946 | 0.944 | -0.580 | -0.580 | -0.871 | |
| lambda25 | 0.948 | 0.940 | 0.937 | -0.290 | -1.451 | -1.886 | |
| lambda26 | 0.951 | 0.954 | 0.953 | 0.145 | 0.580 | 0.435 | |
| lambda27 | 0.947 | 0.945 | 0.946 | -0.435 | -0.725 | -0.580 | |
| lambda28 | 0.947 | 0.946 | 0.942 | -0.435 | -0.580 | -1.161 | |
| lambda29 | 0.957 | 0.953 | 0.952 | 1.016 | 0.435 | 0.290 | |
| lambda30 | 0.948 | 0.947 | 0.944 | -0.290 | -0.435 | -0.871 | |
| lambda31 | 0.953 | 0.948 | 0.951 | 0.435 | -0.290 | 0.145 | |
| lambda32 | 0.944 | 0.941 | 0.946 | -0.871 | -1.306 | -0.580 | |
| lambda33 | 0.958 | 0.954 | 0.955 | 1.161 | 0.580 | 0.725 | |
| lambda34 | 0.947 | 0.948 | 0.945 | -0.435 | -0.290 | -0.725 | |
| lambda35 | 0.959 | 0.954 | 0.956 | 1.306 | 0.580 | 0.871 | |
| lambda36 | 0.953 | 0.953 | 0.951 | 0.435 | 0.435 | 0.145 | |
| lambda37 | 0.942 | 0.943 | 0.942 | -1.161 | -1.016 | -1.161 | |
| lambda38 | 0.951 | 0.949 | 0.946 | 0.145 | -0.145 | -0.580 | |
| lambda39 | 0.963 | 0.959 | 0.958 | 1.886 | 1.306 | 1.161 | |
| lambda40 | 0.958 | 0.954 | 0.954 | 1.161 | 0.580 | 0.580 | |
| lambda41 | 0.956 | 0.950 | 0.949 | 0.871 | 0.000 | -0.145 | |
| lambda42 | 0.954 | 0.951 | 0.952 | 0.580 | 0.145 | 0.290 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda43 | 0.948 | 0.948 | 0.945 | -0.290 | -0.290 | -0.725 | |
| lambda44 | 0.957 | 0.953 | 0.952 | 1.016 | 0.435 | 0.290 | |
| lambda45 | 0.951 | 0.948 | 0.947 | 0.145 | -0.290 | -0.435 | |
| lambda46 | 0.951 | 0.948 | 0.947 | 0.145 | -0.290 | -0.435 | |
| lambda47 | 0.948 | 0.943 | 0.945 | -0.290 | -1.016 | -0.725 | |
| lambda48 | 0.949 | 0.940 | 0.948 | -0.145 | -1.451 | -0.290 | |
| lambda49 | 0.946 | 0.945 | 0.946 | -0.580 | -0.725 | -0.580 | |
| lambda50 | 0.956 | 0.957 | 0.957 | 0.871 | 1.016 | 1.016 | |
| omega1 | 0.673 | 0.932 | 0.935 | -40.191 | -2.612 | -2.176 | * |
| omega2 | 0.673 | 0.945 | 0.945 | -40.191 | -0.725 | -0.725 | * |
| omega3 | 0.718 | 0.941 | 0.942 | -33.662 | -1.306 | -1.161 | * |
| omega4 | 0.663 | 0.925 | 0.929 | -41.642 | -3.627 | -3.047 | * |
| omega5 | 0.670 | 0.946 | 0.943 | -40.627 | -0.580 | -1.016 | * |
| omega6 | 0.684 | 0.948 | 0.952 | -38.595 | -0.290 | 0.290 | * |
| omega7 | 0.685 | 0.944 | 0.943 | -38.450 | -0.871 | -1.016 | * |
| omega8 | 0.685 | 0.941 | 0.939 | -38.450 | -1.306 | -1.596 | * |
| omega9 | 0.701 | 0.953 | 0.949 | -36.129 | 0.435 | -0.145 | * |
| omega10 | 0.690 | 0.947 | 0.943 | -37.725 | -0.435 | -1.016 | * |
| omega11 | 0.670 | 0.939 | 0.938 | -40.627 | -1.596 | -1.741 | * |
| omega12 | 0.686 | 0.935 | 0.941 | -38.305 | -2.176 | -1.306 | * |
| omega13 | 0.697 | 0.940 | 0.938 | -36.709 | -1.451 | -1.741 | * |
| omega14 | 0.674 | 0.937 | 0.938 | -40.046 | -1.886 | -1.741 | * |
| omega15 | 0.697 | 0.928 | 0.928 | -36.709 | -3.192 | -3.192 | * |
| omega16 | 0.658 | 0.928 | 0.928 | -42.368 | -3.192 | -3.192 | * |
| omega17 | 0.691 | 0.937 | 0.936 | -37.580 | -1.886 | -2.031 | * |
| omega18 | 0.680 | 0.944 | 0.943 | -39.176 | -0.871 | -1.016 | * |
| omega19 | 0.674 | 0.937 | 0.939 | -40.046 | -1.886 | -1.596 | * |
| omega20 | 0.690 | 0.931 | 0.935 | -37.725 | -2.757 | -2.176 | * |
| omega21 | 0.699 | 0.939 | 0.939 | -36.419 | -1.596 | -1.596 | * |
| omega22 | 0.680 | 0.938 | 0.943 | -39.176 | -1.741 | -1.016 | * |
| omega23 | 0.655 | 0.929 | 0.933 | -42.803 | -3.047 | -2.467 | * |
| omega24 | 0.687 | 0.939 | 0.941 | -38.160 | -1.596 | -1.306 | * |
| omega25 | 0.664 | 0.944 | 0.942 | -41.497 | -0.871 | -1.161 | * |
| omega26 | 0.652 | 0.935 | 0.933 | -43.238 | -2.176 | -2.467 | * |
| omega27 | 0.678 | 0.926 | 0.930 | -39.466 | -3.482 | -2.902 | * |
| omega28 | 0.665 | 0.935 | 0.934 | -41.352 | -2.176 | -2.322 | * |
| omega29 | 0.676 | 0.927 | 0.924 | -39.756 | -3.337 | -3.772 | * |
| omega30 | 0.672 | 0.943 | 0.940 | -40.336 | -1.016 | -1.451 | * |
| omega31 | 0.673 | 0.941 | 0.941 | -40.191 | -1.306 | -1.306 | * |
| omega32 | 0.695 | 0.951 | 0.947 | -36.999 | 0.145 | -0.435 | * |
| omega33 | 0.641 | 0.926 | 0.928 | -44.834 | -3.482 | -3.192 | * |
| omega34 | 0.693 | 0.948 | 0.941 | -37.289 | -0.290 | -1.306 | * |
| omega35 | 0.678 | 0.945 | 0.945 | -39.466 | -0.725 | -0.725 | * |
| omega36 | 0.683 | 0.947 | 0.942 | -38.740 | -0.435 | -1.161 | * |
| omega37 | 0.701 | 0.944 | 0.942 | -36.129 | -0.871 | -1.161 | * |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega38 | 0.673 | 0.935 | 0.939 | -40.191 | -2.176 | -1.596 | * |
| omega39 | 0.680 | 0.941 | 0.939 | -39.176 | -1.306 | -1.596 | * |
| omega40 | 0.673 | 0.945 | 0.943 | -40.191 | -0.725 | -1.016 | * |
| omega41 | 0.689 | 0.941 | 0.938 | -37.870 | -1.306 | -1.741 | * |
| omega42 | 0.695 | 0.938 | 0.936 | -36.999 | -1.741 | -2.031 | * |
| omega43 | 0.641 | 0.943 | 0.942 | -44.834 | -1.016 | -1.161 | * |
| omega44 | 0.661 | 0.931 | 0.932 | -41.933 | -2.757 | -2.612 | * |
| omega45 | 0.695 | 0.952 | 0.953 | -36.999 | 0.290 | 0.435 | * |
| omega46 | 0.691 | 0.947 | 0.950 | -37.580 | -0.435 | 0.000 | * |
| omega47 | 0.678 | 0.944 | 0.943 | -39.466 | -0.871 | -1.016 | * |
| omega48 | 0.690 | 0.940 | 0.940 | -37.725 | -1.451 | -1.451 | * |
| omega49 | 0.685 | 0.950 | 0.951 | -38.450 | 0.000 | 0.145 | * |
| omega50 | 0.687 | 0.940 | 0.941 | -38.160 | -1.451 | -1.306 | * |
| phi | 0.832 | 0.941 | 0.941 | -17.121 | -1.306 | -1.306 | * |

* At least one $z$ statistic exceeds the Bonferroni critical value of 3.76 for 300 two-sided $z$-tests of $H_0$: Coverage = 0.95.

This a success; all the sandwich and bootstrap intervals had acceptable coverage, and the sample size of $2,000$ is only 500 more than was needed for the comparable but much smaller Extra Response Variables model (5.9).

### Scaled beta base distribution

At this point, we have stopped worrying that the large number of variables in this example is going to present special problems. For the non-normal but light-tailed beta base distribution, we expect normal theory intervals to perform fairly well. When the data were actually normal, we had good results for the straight-arrow parameters (the factor loadings) with $n = 200$, but an $n$ of at least 500 was required for all the variance parameters to have acceptable coverage. Table 5.34 shows results for the scaled beta distribution with $n = 200$.

Table 5.34: Coverage of 95% confidence intervals for the "Big Data" model (5.15), $n = 200$, Scaled beta base distribution, 1,000 simulated data sets

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda2 | 0.956 | 0.951 | 0.950 | 0.871 | 0.145 | 0.000 | |
| lambda3 | 0.967 | 0.952 | 0.957 | 2.467 | 0.290 | 1.016 | |
| lambda4 | 0.957 | 0.953 | 0.959 | 1.016 | 0.435 | 1.306 | |
| lambda5 | 0.953 | 0.950 | 0.951 | 0.435 | 0.000 | 0.145 | |
| lambda6 | 0.947 | 0.944 | 0.941 | -0.435 | -0.871 | -1.306 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda7 | 0.940 | 0.935 | 0.935 | -1.451 | -2.176 | -2.176 | |
| lambda8 | 0.943 | 0.934 | 0.940 | -1.016 | -2.322 | -1.451 | |
| lambda9 | 0.959 | 0.951 | 0.944 | 1.306 | 0.145 | -0.871 | |
| lambda10 | 0.956 | 0.953 | 0.952 | 0.871 | 0.435 | 0.290 | |
| lambda11 | 0.955 | 0.950 | 0.945 | 0.725 | 0.000 | -0.725 | |
| lambda12 | 0.956 | 0.950 | 0.949 | 0.871 | 0.000 | -0.145 | |
| lambda13 | 0.954 | 0.949 | 0.943 | 0.580 | -0.145 | -1.016 | |
| lambda14 | 0.951 | 0.949 | 0.956 | 0.145 | -0.145 | 0.871 | |
| lambda15 | 0.941 | 0.941 | 0.944 | -1.306 | -1.306 | -0.871 | |
| lambda16 | 0.955 | 0.947 | 0.954 | 0.725 | -0.435 | 0.580 | |
| lambda17 | 0.958 | 0.952 | 0.953 | 1.161 | 0.290 | 0.435 | |
| lambda18 | 0.951 | 0.945 | 0.949 | 0.145 | -0.725 | -0.145 | |
| lambda19 | 0.948 | 0.948 | 0.947 | -0.290 | -0.290 | -0.435 | |
| lambda20 | 0.945 | 0.946 | 0.944 | -0.725 | -0.580 | -0.871 | |
| lambda21 | 0.947 | 0.942 | 0.946 | -0.435 | -1.161 | -0.580 | |
| lambda22 | 0.946 | 0.952 | 0.952 | -0.580 | 0.290 | 0.290 | |
| lambda23 | 0.967 | 0.955 | 0.953 | 2.467 | 0.725 | 0.435 | |
| lambda24 | 0.955 | 0.947 | 0.945 | 0.725 | -0.435 | -0.725 | |
| lambda25 | 0.947 | 0.943 | 0.944 | -0.435 | -1.016 | -0.871 | |
| lambda26 | 0.947 | 0.939 | 0.943 | -0.435 | -1.596 | -1.016 | |
| lambda27 | 0.956 | 0.953 | 0.952 | 0.871 | 0.435 | 0.290 | |
| lambda28 | 0.954 | 0.946 | 0.952 | 0.580 | -0.580 | 0.290 | |
| lambda29 | 0.951 | 0.945 | 0.951 | 0.145 | -0.725 | 0.145 | |
| lambda30 | 0.950 | 0.942 | 0.940 | 0.000 | -1.161 | -1.451 | |
| lambda31 | 0.953 | 0.946 | 0.949 | 0.435 | -0.580 | -0.145 | |
| lambda32 | 0.943 | 0.940 | 0.950 | -1.016 | -1.451 | 0.000 | |
| lambda33 | 0.943 | 0.936 | 0.940 | -1.016 | -2.031 | -1.451 | |
| lambda34 | 0.954 | 0.950 | 0.953 | 0.580 | 0.000 | 0.435 | |
| lambda35 | 0.944 | 0.944 | 0.944 | -0.871 | -0.871 | -0.871 | |
| lambda36 | 0.953 | 0.951 | 0.955 | 0.435 | 0.145 | 0.725 | |
| lambda37 | 0.952 | 0.945 | 0.949 | 0.290 | -0.725 | -0.145 | |
| lambda38 | 0.948 | 0.943 | 0.941 | -0.290 | -1.016 | -1.306 | |
| lambda39 | 0.955 | 0.952 | 0.952 | 0.725 | 0.290 | 0.290 | |
| lambda40 | 0.965 | 0.963 | 0.960 | 2.176 | 1.886 | 1.451 | |
| lambda41 | 0.946 | 0.942 | 0.945 | -0.580 | -1.161 | -0.725 | |
| lambda42 | 0.949 | 0.942 | 0.951 | -0.145 | -1.161 | 0.145 | |
| lambda43 | 0.966 | 0.962 | 0.965 | 2.322 | 1.741 | 2.176 | |
| lambda44 | 0.959 | 0.950 | 0.958 | 1.306 | 0.000 | 1.161 | |
| lambda45 | 0.938 | 0.929 | 0.943 | -1.741 | -3.047 | -1.016 | |
| lambda46 | 0.952 | 0.949 | 0.946 | 0.290 | -0.145 | -0.580 | |
| lambda47 | 0.951 | 0.942 | 0.947 | 0.145 | -1.161 | -0.435 | |
| lambda48 | 0.960 | 0.950 | 0.956 | 1.451 | 0.000 | 0.871 | |
| lambda49 | 0.964 | 0.959 | 0.957 | 2.031 | 1.306 | 1.016 | |
| lambda50 | 0.948 | 0.946 | 0.944 | -0.290 | -0.580 | -0.871 | |
| omega1 | 0.930 | 0.927 | 0.928 | -2.902 | -3.337 | -3.192 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega2 | 0.919 | 0.921 | 0.916 | -4.498 | -4.208 | -4.933 | * |
| omega3 | 0.917 | 0.920 | 0.911 | -4.788 | -4.353 | -5.659 | * |
| omega4 | 0.930 | 0.932 | 0.932 | -2.902 | -2.612 | -2.612 | |
| omega5 | 0.921 | 0.923 | 0.921 | -4.208 | -3.918 | -4.208 | * |
| omega6 | 0.928 | 0.932 | 0.934 | -3.192 | -2.612 | -2.322 | |
| omega7 | 0.935 | 0.928 | 0.927 | -2.176 | -3.192 | -3.337 | |
| omega8 | 0.925 | 0.924 | 0.926 | -3.627 | -3.772 | -3.482 | * |
| omega9 | 0.928 | 0.926 | 0.926 | -3.192 | -3.482 | -3.482 | |
| omega10 | 0.926 | 0.933 | 0.928 | -3.482 | -2.467 | -3.192 | |
| omega11 | 0.937 | 0.939 | 0.937 | -1.886 | -1.596 | -1.886 | |
| omega12 | 0.941 | 0.942 | 0.941 | -1.306 | -1.161 | -1.306 | |
| omega13 | 0.933 | 0.923 | 0.922 | -2.467 | -3.918 | -4.063 | * |
| omega14 | 0.930 | 0.931 | 0.926 | -2.902 | -2.757 | -3.482 | |
| omega15 | 0.931 | 0.931 | 0.934 | -2.757 | -2.757 | -2.322 | |
| omega16 | 0.938 | 0.936 | 0.936 | -1.741 | -2.031 | -2.031 | |
| omega17 | 0.929 | 0.935 | 0.931 | -3.047 | -2.176 | -2.757 | |
| omega18 | 0.944 | 0.944 | 0.938 | -0.871 | -0.871 | -1.741 | |
| omega19 | 0.934 | 0.942 | 0.939 | -2.322 | -1.161 | -1.596 | |
| omega20 | 0.946 | 0.940 | 0.938 | -0.580 | -1.451 | -1.741 | |
| omega21 | 0.932 | 0.924 | 0.928 | -2.612 | -3.772 | -3.192 | * |
| omega22 | 0.913 | 0.914 | 0.911 | -5.369 | -5.223 | -5.659 | * |
| omega23 | 0.930 | 0.932 | 0.934 | -2.902 | -2.612 | -2.322 | |
| omega24 | 0.937 | 0.940 | 0.935 | -1.886 | -1.451 | -2.176 | |
| omega25 | 0.941 | 0.939 | 0.934 | -1.306 | -1.596 | -2.322 | |
| omega26 | 0.927 | 0.923 | 0.913 | -3.337 | -3.918 | -5.369 | * |
| omega27 | 0.940 | 0.932 | 0.933 | -1.451 | -2.612 | -2.467 | |
| omega28 | 0.934 | 0.935 | 0.931 | -2.322 | -2.176 | -2.757 | |
| omega29 | 0.938 | 0.947 | 0.946 | -1.741 | -0.435 | -0.580 | |
| omega30 | 0.941 | 0.945 | 0.944 | -1.306 | -0.725 | -0.871 | |
| omega31 | 0.951 | 0.952 | 0.954 | 0.145 | 0.290 | 0.580 | |
| omega32 | 0.926 | 0.928 | 0.926 | -3.482 | -3.192 | -3.482 | |
| omega33 | 0.936 | 0.931 | 0.927 | -2.031 | -2.757 | -3.337 | |
| omega34 | 0.941 | 0.939 | 0.938 | -1.306 | -1.596 | -1.741 | |
| omega35 | 0.926 | 0.932 | 0.936 | -3.482 | -2.612 | -2.031 | |
| omega36 | 0.920 | 0.927 | 0.926 | -4.353 | -3.337 | -3.482 | * |
| omega37 | 0.928 | 0.935 | 0.935 | -3.192 | -2.176 | -2.176 | |
| omega38 | 0.929 | 0.923 | 0.920 | -3.047 | -3.918 | -4.353 | * |
| omega39 | 0.919 | 0.927 | 0.928 | -4.498 | -3.337 | -3.192 | * |
| omega40 | 0.932 | 0.930 | 0.925 | -2.612 | -2.902 | -3.627 | |
| omega41 | 0.941 | 0.939 | 0.941 | -1.306 | -1.596 | -1.306 | |
| omega42 | 0.924 | 0.920 | 0.918 | -3.772 | -4.353 | -4.643 | * |
| omega43 | 0.939 | 0.937 | 0.938 | -1.596 | -1.886 | -1.741 | |
| omega44 | 0.935 | 0.933 | 0.931 | -2.176 | -2.467 | -2.757 | |
| omega45 | 0.916 | 0.918 | 0.914 | -4.933 | -4.643 | -5.223 | * |
| omega46 | 0.922 | 0.924 | 0.918 | -4.063 | -3.772 | -4.643 | * |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega47 | 0.932 | 0.930 | 0.926 | -2.612 | -2.902 | -3.482 | |
| omega48 | 0.915 | 0.920 | 0.915 | -5.078 | -4.353 | -5.078 | * |
| omega49 | 0.936 | 0.933 | 0.932 | -2.031 | -2.467 | -2.612 | |
| omega50 | 0.937 | 0.931 | 0.928 | -1.886 | -2.757 | -3.192 | |
| phi | 0.956 | 0.947 | 0.949 | 0.871 | -0.435 | -0.145 | |

* At least one $z$ statistic exceeds the Bonferroni critical value of 3.76 for 300 two-sided $z$-tests of $H_0$: Coverage = 0.95.

Coverage is okay for all the straight-arrow factor loadings, while the $z$ statistics for fifteen variance parameters are over the Bonferroni limit. This is similar to what happened with the normal base distribution; in that case, $n = 500$ was adequate to take care of the problem. Table 5.35 contains results for $n = 500$ with the scaled beta base distribution.

Table 5.35: Coverage of 95% confidence intervals for the "Big Data" model (5.15), $n = 500$, Scaled beta base distribution, 1,000 simulated data sets

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda2 | 0.952 | 0.949 | 0.952 | 0.290 | -0.145 | 0.290 | |
| lambda3 | 0.948 | 0.948 | 0.949 | -0.290 | -0.290 | -0.145 | |
| lambda4 | 0.942 | 0.942 | 0.953 | -1.161 | -1.161 | 0.435 | |
| lambda5 | 0.945 | 0.945 | 0.944 | -0.725 | -0.725 | -0.871 | |
| lambda6 | 0.949 | 0.945 | 0.944 | -0.145 | -0.725 | -0.871 | |
| lambda7 | 0.951 | 0.950 | 0.950 | 0.145 | 0.000 | 0.000 | |
| lambda8 | 0.955 | 0.951 | 0.946 | 0.725 | 0.145 | -0.580 | |
| lambda9 | 0.934 | 0.937 | 0.935 | -2.322 | -1.886 | -2.176 | |
| lambda10 | 0.942 | 0.944 | 0.942 | -1.161 | -0.871 | -1.161 | |
| lambda11 | 0.935 | 0.933 | 0.939 | -2.176 | -2.467 | -1.596 | |
| lambda12 | 0.955 | 0.955 | 0.945 | 0.725 | 0.725 | -0.725 | |
| lambda13 | 0.958 | 0.958 | 0.954 | 1.161 | 1.161 | 0.580 | |
| lambda14 | 0.960 | 0.958 | 0.957 | 1.451 | 1.161 | 1.016 | |
| lambda15 | 0.942 | 0.941 | 0.937 | -1.161 | -1.306 | -1.886 | |
| lambda16 | 0.947 | 0.948 | 0.947 | -0.435 | -0.290 | -0.435 | |
| lambda17 | 0.947 | 0.941 | 0.940 | -0.435 | -1.306 | -1.451 | |
| lambda18 | 0.957 | 0.952 | 0.950 | 1.016 | 0.290 | 0.000 | |
| lambda19 | 0.950 | 0.949 | 0.951 | 0.000 | -0.145 | 0.145 | |
| lambda20 | 0.948 | 0.946 | 0.945 | -0.290 | -0.580 | -0.725 | |
| lambda21 | 0.948 | 0.944 | 0.945 | -0.290 | -0.871 | -0.725 | |
| lambda22 | 0.952 | 0.952 | 0.951 | 0.290 | 0.290 | 0.145 | |
| lambda23 | 0.947 | 0.948 | 0.937 | -0.435 | -0.290 | -1.886 | |
| lambda24 | 0.952 | 0.956 | 0.953 | 0.290 | 0.871 | 0.435 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| lambda25 | 0.944 | 0.945 | 0.942 | -0.871 | -0.725 | -1.161 | |
| lambda26 | 0.944 | 0.942 | 0.941 | -0.871 | -1.161 | -1.306 | |
| lambda27 | 0.956 | 0.957 | 0.955 | 0.871 | 1.016 | 0.725 | |
| lambda28 | 0.958 | 0.952 | 0.953 | 1.161 | 0.290 | 0.435 | |
| lambda29 | 0.952 | 0.952 | 0.946 | 0.290 | 0.290 | -0.580 | |
| lambda30 | 0.957 | 0.954 | 0.958 | 1.016 | 0.580 | 1.161 | |
| lambda31 | 0.952 | 0.948 | 0.955 | 0.290 | -0.290 | 0.725 | |
| lambda32 | 0.945 | 0.945 | 0.937 | -0.725 | -0.725 | -1.886 | |
| lambda33 | 0.960 | 0.958 | 0.951 | 1.451 | 1.161 | 0.145 | |
| lambda34 | 0.958 | 0.956 | 0.955 | 1.161 | 0.871 | 0.725 | |
| lambda35 | 0.965 | 0.963 | 0.967 | 2.176 | 1.886 | 2.467 | |
| lambda36 | 0.958 | 0.951 | 0.951 | 1.161 | 0.145 | 0.145 | |
| lambda37 | 0.950 | 0.941 | 0.941 | 0.000 | -1.306 | -1.306 | |
| lambda38 | 0.966 | 0.958 | 0.959 | 2.322 | 1.161 | 1.306 | |
| lambda39 | 0.950 | 0.947 | 0.950 | 0.000 | -0.435 | 0.000 | |
| lambda40 | 0.958 | 0.953 | 0.957 | 1.161 | 0.435 | 1.016 | |
| lambda41 | 0.953 | 0.947 | 0.945 | 0.435 | -0.435 | -0.725 | |
| lambda42 | 0.945 | 0.942 | 0.946 | -0.725 | -1.161 | -0.580 | |
| lambda43 | 0.946 | 0.944 | 0.944 | -0.580 | -0.871 | -0.871 | |
| lambda44 | 0.961 | 0.958 | 0.959 | 1.596 | 1.161 | 1.306 | |
| lambda45 | 0.949 | 0.944 | 0.944 | -0.145 | -0.871 | -0.871 | |
| lambda46 | 0.963 | 0.956 | 0.957 | 1.886 | 0.871 | 1.016 | |
| lambda47 | 0.936 | 0.938 | 0.936 | -2.031 | -1.741 | -2.031 | |
| lambda48 | 0.953 | 0.949 | 0.948 | 0.435 | -0.145 | -0.290 | |
| lambda49 | 0.955 | 0.951 | 0.948 | 0.725 | 0.145 | -0.290 | |
| lambda50 | 0.950 | 0.943 | 0.950 | 0.000 | -1.016 | 0.000 | |
| omega1 | 0.950 | 0.953 | 0.950 | 0.000 | 0.435 | 0.000 | |
| omega2 | 0.941 | 0.942 | 0.939 | -1.306 | -1.161 | -1.596 | |
| omega3 | 0.937 | 0.944 | 0.941 | -1.886 | -0.871 | -1.306 | |
| omega4 | 0.948 | 0.951 | 0.952 | -0.290 | 0.145 | 0.290 | |
| omega5 | 0.936 | 0.941 | 0.937 | -2.031 | -1.306 | -1.886 | |
| omega6 | 0.938 | 0.939 | 0.938 | -1.741 | -1.596 | -1.741 | |
| omega7 | 0.937 | 0.936 | 0.934 | -1.886 | -2.031 | -2.322 | |
| omega8 | 0.937 | 0.943 | 0.941 | -1.886 | -1.016 | -1.306 | |
| omega9 | 0.945 | 0.947 | 0.952 | -0.725 | -0.435 | 0.290 | |
| omega10 | 0.940 | 0.946 | 0.947 | -1.451 | -0.580 | -0.435 | |
| omega11 | 0.943 | 0.947 | 0.946 | -1.016 | -0.435 | -0.580 | |
| omega12 | 0.933 | 0.936 | 0.932 | -2.467 | -2.031 | -2.612 | |
| omega13 | 0.945 | 0.948 | 0.951 | -0.725 | -0.290 | 0.145 | |
| omega14 | 0.933 | 0.936 | 0.940 | -2.467 | -2.031 | -1.451 | |
| omega15 | 0.931 | 0.937 | 0.937 | -2.757 | -1.886 | -1.886 | |
| omega16 | 0.935 | 0.937 | 0.933 | -2.176 | -1.886 | -2.467 | |
| omega17 | 0.942 | 0.940 | 0.938 | -1.161 | -1.451 | -1.741 | |
| omega18 | 0.944 | 0.950 | 0.949 | -0.871 | 0.000 | -0.145 | |
| omega19 | 0.941 | 0.947 | 0.945 | -1.306 | -0.435 | -0.725 | |

| | Coverage | | | Z-tests | | | |
|---|---|---|---|---|---|---|---|
| | Normal.Theory | Sand. | Bootstrap | Normal.Theory | Sand. | Bootstrap | Sig |
| omega20 | 0.939 | 0.945 | 0.945 | -1.596 | -0.725 | -0.725 | |
| omega21 | 0.948 | 0.955 | 0.955 | -0.290 | 0.725 | 0.725 | |
| omega22 | 0.936 | 0.942 | 0.943 | -2.031 | -1.161 | -1.016 | |
| omega23 | 0.939 | 0.947 | 0.947 | -1.596 | -0.435 | -0.435 | |
| omega24 | 0.931 | 0.940 | 0.942 | -2.757 | -1.451 | -1.161 | |
| omega25 | 0.944 | 0.945 | 0.949 | -0.871 | -0.725 | -0.145 | |
| omega26 | 0.947 | 0.951 | 0.947 | -0.435 | 0.145 | -0.435 | |
| omega27 | 0.936 | 0.946 | 0.943 | -2.031 | -0.580 | -1.016 | |
| omega28 | 0.928 | 0.932 | 0.932 | -3.192 | -2.612 | -2.612 | |
| omega29 | 0.937 | 0.945 | 0.947 | -1.886 | -0.725 | -0.435 | |
| omega30 | 0.937 | 0.938 | 0.937 | -1.886 | -1.741 | -1.886 | |
| omega31 | 0.936 | 0.938 | 0.940 | -2.031 | -1.741 | -1.451 | |
| omega32 | 0.943 | 0.942 | 0.940 | -1.016 | -1.161 | -1.451 | |
| omega33 | 0.924 | 0.930 | 0.927 | -3.772 | -2.902 | -3.337 | * |
| omega34 | 0.935 | 0.946 | 0.943 | -2.176 | -0.580 | -1.016 | |
| omega35 | 0.925 | 0.933 | 0.934 | -3.627 | -2.467 | -2.322 | |
| omega36 | 0.941 | 0.941 | 0.946 | -1.306 | -1.306 | -0.580 | |
| omega37 | 0.944 | 0.945 | 0.945 | -0.871 | -0.725 | -0.725 | |
| omega38 | 0.938 | 0.943 | 0.940 | -1.741 | -1.016 | -1.451 | |
| omega39 | 0.945 | 0.944 | 0.941 | -0.725 | -0.871 | -1.306 | |
| omega40 | 0.934 | 0.940 | 0.944 | -2.322 | -1.451 | -0.871 | |
| omega41 | 0.940 | 0.937 | 0.936 | -1.451 | -1.886 | -2.031 | |
| omega42 | 0.929 | 0.934 | 0.934 | -3.047 | -2.322 | -2.322 | |
| omega43 | 0.941 | 0.948 | 0.946 | -1.306 | -0.290 | -0.580 | |
| omega44 | 0.940 | 0.943 | 0.939 | -1.451 | -1.016 | -1.596 | |
| omega45 | 0.931 | 0.936 | 0.929 | -2.757 | -2.031 | -3.047 | |
| omega46 | 0.951 | 0.950 | 0.950 | 0.145 | 0.000 | 0.000 | |
| omega47 | 0.937 | 0.941 | 0.937 | -1.886 | -1.306 | -1.886 | |
| omega48 | 0.935 | 0.943 | 0.940 | -2.176 | -1.016 | -1.451 | |
| omega49 | 0.933 | 0.939 | 0.938 | -2.467 | -1.596 | -1.741 | |
| omega50 | 0.931 | 0.928 | 0.928 | -2.757 | -3.192 | -3.192 | |
| phi | 0.942 | 0.939 | 0.935 | -1.161 | -1.596 | -2.176 | |

* At least one $z$ statistic exceeds the Bonferroni critical value of 3.76 for 300 two-sided $z$-tests of $H_0$: Coverage = 0.95.

Again, the coverage for all the factor loadings is fine. Only a single interval for a variance parameter (normal theory) fails the Bonferroni test. This counts as a success. Robustness holds for the normal theory standard errors, even for non-normal data and even for variance parameters, provided the non-normal data do not have much excess kurtosis. The large number of observed variables has little effect on this finding.

### 5.6.6 Secondary analyses and conclusions

We have some conclusions that hold across a good variety of models. Contrary to blanket claims on both sides [4, 26, 48], normal theory standard errors are sometimes robust, but not always. When robustness fails, it is for distributions with excess kurtosis — that is, with heavy tails. In this case, the sandwich and bootstrap standard errors are robust, but unexpectedly large sample sizes may be needed for the robustness to fully kick in. Some models require larger sample sizes than others, and it's hard to predict when this will happen by just looking at the path diagram. In the simulations of this chapter, normal theory standard errors performed well for non-normal data, provided the non-normal distribution was not heavy-tailed.

The Satorra-Bentler principle was consistently supported. Even with heavy-tailed data, normal theory intervals are robust for straight-arrow parameters as long as the variance parameters are unrestricted by the model (see the Satorra-Bentler principle for exact details). A notable example of variance parameters that *are* restricted by the model is in a factor analysis model where identifiability is purchased by standardizing the factors. In this important case, the normal theory standard errors for the factor loadings are not robust.

The Satorra-Bentler principle does not say anything about robustness for variance and covariance parameters. Generally, when the distributions are heavy-tailed, normal theory standard errors are not robust for the variance and covariance parameters. An exception is for covariances of exogenous variables (including error terms), when the variables in question are independent, and not just uncorrelated. In this case, the normal theory standard error is correct regardless of the distribution of the data.

**Choosing between robust methods** If more than one method is known to be robust in a particular setting, which one should you use? It is definitely tempting to note that while normal theory standard errors are robust for some parameters under some circumstances, the sandwich and bootstrap standard errors are always robust (given the existence of fourth moments). Is it a good idea to "play it safe," and just use a method that's guaranteed to be robust? Do we ever need the normal theory standard errors?

Since robustness is an $n \to \infty$ property, it is clear that for a given finite sample size, one robust method could easily perform better than another one. Fortunately, the simulation studies were designed so that it's easy to go back and test whether apparent differences in performance are statistically significant. The code for a simulation produces a large data file, with one line for each simulation. On each line, there is a zero or a one for each parameter and each standard error, with one indicating that the confidence interval covered the true parameter value. The tables of results shown in this chapter are generated by processing these data files.

The expected values of the zero-one indicators are exactly the coverage probabilities. Suppose $I_{i,1}$ and $I_{i,2}$ are indicators for coverage of two competing intervals for simulation $i$, with respective true coverage probabilities $\theta_1$ and $\theta_2$. The difference between the indicators is a strange discrete random variable taking values $-1, 0, 1$, but

$$E\left(I_{i,1} - I_{i,2}\right) = E\left(I_{i,1}\right) - E\left(I_{i,2}\right) = \theta_1 - \theta_2.$$

The Central Limit Theorem assures us that the sample mean of these differences has a distribution that's asymptotically normal[28], with expected value $\theta_1 - \theta_2$, and a variance that can easily be estimated from the data without bothering to work out exactly what it is. The easiest way to proceed is to use R's `t.test` function, relying on the robustness of the one-sample $t$-test described at the beginning of this chapter. That is, one carries out elementary matched $t$-tests on binary data, confidently describing the result as a $z$ statistic.

**The "Big Data" data** The testing strategy just described will be the workhorse of this section, but it would be tedious to apply to the so-called Big Data model, because there are so many parameters. Fortunately, the lazy choice of a single true value of the factor loadings and a single true value of the error variances means that there is only one true coverage probability for the factor loadings $\lambda_j$, and one true coverage probability for the error variances $\omega_j$. Thus, we can take sample means of indicators for the coverage of all the factor loadings in a given simulation for a given method, and use that number in a matched $t$-test.

To show how the analysis goes, consider the data used to generate Table 5.31. That's the Big Data model, exponential base distribution and $n = 200$. Surprisingly to me, *all* 49 factor loadings have acceptable coverage using normal theory and bootstrap, while the sandwich missed on only 2/49. It is surprising because the number of parameters is so large relative to the sample size. The factor loadings are the straight-arrow parameters of the Satorra-Bentler principle, which is being confirmed here in a big way. However, is it needed? Are the normal theory intervals *significantly* better? Note that in the code that follows, the sandwich estimators are called "Huber." This is the way I did it initially, and I did not go back and fix the vocabulary in my code.

First, we read the data.

```
> rm(list=ls())
> results = read.table("BigDataExpo200-output.txt")
> dim(results)
[1] 1000  300
```

That's right. There were one thousand simulated data sets, with one hundred unknown parameters. For each data set, three confidence intervals were calculated for each parameter: normal theory, sandwich and bootstrap. A one was recorded if the interval covered the parameter, and a zero otherwise. The columns of the `results` data frame correspond to 49 $\lambda_j$ factor loadings, followed by 50 $\omega_j$ error variances, and then a single column for $\phi = Var(F)$. This is repeated three times, one for normal theory, one for Huber (the sandwich estimator), and one for the bootstrap.

For each method, we extract the results for the factor loadings, obtaining three $1000 \times 49$ data frames. The `apply` function is used to calculate the sample mean for each row.

[28]The "sample size" here is the number of simulations, and it's guaranteed to be large enough for the Central Limit Theorem to apply.

```
> N = results[,1:49]; H = results[,101:149]; B = results[,201:249]
> Normal = apply(N,1,FUN=mean); Huber = apply(H,1,FUN=mean)
> Boot = apply(B,1,FUN=mean)
```

Calculating the sample means (of means) yields estimated coverages for the three methods.

```
> round(apply(cbind(Normal,Huber,Boot),2,FUN=mean),4)
Normal  Huber   Boot
0.9479 0.9354 0.9399
```

Coverage of the normal theory intervals looks clearly better. Now carry out three pairwise tests. First comes normal versus sandwich, and then the other two comparisons.

```
> options(scipen=999) # To suppress scientific notation for the p-values
> t.test(Normal, Huber , paired=TRUE)

Paired t-test

data:  Normal and Huber
t = 9.1919, df = 999, p-value < 0.00000000000000022
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 0.009887619 0.015255238
sample estimates:
mean difference
     0.01257143

> t.test(Normal, Boot , paired=TRUE)
Paired t-test

data:  Normal and Boot
t = 3.9275, df = 999, p-value = 0.00009174
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 0.004033466 0.012088983
sample estimates:
mean difference
    0.008061224

> t.test(Huber, Boot , paired=TRUE)

Paired t-test

data:  Huber and Boot
t = -3.1902, df = 999, p-value = 0.001466
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
```

```
 -0.007284495 -0.001735913
sample estimates:
mean difference
   -0.004510204
```

This is clear evidence in favour of the normal theory standard errors, at least for this model, an exponential base distribution and $n = 200$. Also, the bootstrap does significantly better than the sandwich. That is, the difference in performance is *statistically* significant. The actual difference in empirical coverage, 0.9399 versus 0.9354, is tiny.

There are four more relevant tables for this comparison: Exponential base distribution with $n = 1,500$ (Table 5.32), exponential base distribution with $n = 2,000$ (Table 5.33), Scaled beta with $n = 200$ (Table 5.34), and scaled beta with $n = 500$ (Table 5.34). There is no point in checking simulations where the base distribution really is normal, because in that case the normal theory methods are guaranteed to be unbeatable.

One might anticipate that at least for the exponential base distribution, the advantage of the normal theory standard errors would disappear or diminish with larger sample sizes. Results are summarized in Table 5.36.

Table 5.36: Testing differences in coverage for $\lambda_j$ parameters in the "Big Data" model (5.15)

| | Base | | Estimated Coverage | | | Normal vs. | Normal vs. | Sandwich vs. |
|---|---|---|---|---|---|---|---|---|
| Table | Distribution | $n$ | Normal | Sandwich | Bootstrap | Sandwich | Bootstrap | Bootstrap |
| 5.31 | Exponential | 200 | 0.9479 | 0.9354 | 0.9399 | $z = 9.19*$ | $z = 3.93^*$ | $z = -3.19^*$ |
| 5.32 | Exponential | 1500 | 0.9472 | 0.9456 | 0.9460 | $z = 2.06$ | $z = 1.18$ | $z = -0.66$ |
| 5.33 | Exponential | 2000 | 0.9508 | 0.9481 | 0.9481 | $z = 3.87^*$ | $z = 2.89$ | $z = -0.14$ |
| 5.34 | Beta | 200 | 0.9521 | 0.9470 | 0.9489 | $z = 8.26^*$ | $z = 2.37$ | $z = -1.49$ |
| 5.35 | Beta | 500 | 0.9510 | 0.9487 | 0.9482 | $z = 4.49$ | $z = 2.69$ | $z = 0.57$ |

$*$ Greater than Bonferroni critical value of 2.93, correcting for 15 tests.

In Table 5.36, the normal confidence interval has slightly greater coverage than the sandwich for exponential with $n = 2,000$, beta with $n = 200$ and beta with $n = 500$. The differences are statistically significant, but very small, with the estimated normal coverage slightly above 0.95, whicle the estimated coverage of the sandwich interval is slightly below 0.95. The largest difference is 0.952 for the normal compared to 0.947 for the sandwich. Furthermore, none of the estimated coverages involved is significantly different from 0.95, so my conclusion is that for the straight arrow (factor loading) parameters in the Big Data model, with $n = 200$ and an exponential base distribution, the normal theory standard errors performed better than the sandwich and somewhat better than the bootstrap. For the beta base distribution and larger sample sizes, the three methods performed about equally well.

Based on just these analyses, it appears that we might want to rely on the normal theory standard errors for straight-arrow parameters, especially for small to moderate sample sizes. However, the simulations contain a lot more information on this issue, and we need to check it before drawing firm conclusions.

**Straight arrow parameters for the other models** Our interest here is in whether the normal theory confidence intervals for parameters covered by the Satorra-Bentler principle might perform better than the sandwich and bootstrap alternatives. For this reason, we temporarily set aside the standardized 2-factor model (5.14), where the Satorra-Bentler principle does not apply because of restrictions on the variances of the factors. We are left with the following, which is still plenty.

- Extra response variable Model 5.9, parameters $\beta_1$ and $\beta_2$
  - Exponential base distribution, $n = 200$ (Table 5.3)
  - Exponential base distribution, $n = 1,000$ (Table 5.4)
  - Exponential base distribution, $n = 50$ (Table 5.5)
  - Beta base distribution, $n = 200$ (Table 5.6)
  - Beta base distribution, $n = 50$ (Table 5.7)
- Double measurement regression Model 5.5, parameter $\beta$
  - Exponential base distribution, $n = 200$ (Table 5.9)
  - Exponential base distribution, $n = 1,000$ (Table 5.10)
  - Exponential base distribution, $n = 50$ (Table 5.11)
  - Beta base distribution, $n = 200$ (Table 5.12)
- Dip Down path Model 5.10, parameters $\gamma_1$, $\gamma_2$ and $\beta$
  - Exponential base distribution, $n = 200$ (Table 5.16)
  - Exponential base distribution with $\phi_{12} = 0$, $n = 200$ (Table 5.17)
  - Exponential base distribution, $n = 1,000$ (Table 5.19)
  - Beta base distribution, $n = 200$ (Table 5.20)
  - Beta base distribution, $n = 500$ (Table 5.21)

Just counting which empirical coverage comes closest to 0.95 and disregarding ties, normal theory won 13, the sandwich won 5 and the bootstrap won 9. Be reminded that in all these comparisons, the data were not normally distributed.

In terms of formal testing, there are 29 target parameters, and three pairwise comparisons of methods for each one. That's a total of 87 non-independent tests. With a Bonferroni correction, only four comparisons are statistically significant.

- Extra response variable Model 5.9, parameter $\beta_2$, Exponential base distribution, and $n = 50$: Normal was more accurate than Sandwich.

- Extra response variable Model 5.9, parameter $\beta_2$, Beta base distribution, and $n = 50$: Normal coverage was greater than Sandwich, but there is some ambiguity. The normal empirical coverage was 0.956, while coverage of the sandwich interval was 0.933. So while the normal coverage is closer to 0.95 than the sandwich coverage and they are different, the test is *not* saying that normal is more accurate. Also, none of the three empirical coverages differed significantly from 0.95 with the Bonferroni correction for tests on this table.

- Double measurement regression Model 5.5, parameter $\beta$, Exponential base distribution, and $n = 50$.

  - Normal was more accurate than sandwich.
  - Bootstrap was more accurate than sandwich.

There is not a lot of evidence for difference in coverage. Where there is evidence, it favours the normal theory standard errors over the sandwich, for small sample sizes.

It is also instructive to look at the tests that were significant without a Bonferroni correction, corresponding to $z$ values greater than 1.96 in absolute value. This allows the examination of trends, without necessarily taking the individual tests too seriously. Here's the outcome. Note that all the results are for $n \leq 200$.

- Normal beat sandwich four times with $n = 50$ and three times with $n = 200$. Sandwich never beat normal.

- Normal beat bootstrap once with $n = 50$. Bootstrap beat normal twice with $n = 200$.

- Sandwich beat bootstrap once with $n = 200$. Bootstrap beat the sandwich twice with $n = 50$ and three times with $n = 200$.

The conclusion does not change. When the Satorra-Bentler principle applies, normal theory standard errors are somewhat preferred over the sandwich for smaller sample sizes. Bootstrap may have a slight edge over the sandwich for small sample sizes, and there is no convincing evidence of a difference between normal and bootstrap. There is no support for using classical robust standard errors by default. The use of normal theory standard errors for regression with measurement error (Chapter 0 is fully justified, since the Satorra-Bentler principle applies and the tests of interest are all about the regression coefficients, which are straight-arrow parameters.

**Factor loadings in the Standardized Two-factor Model 5.14** This is a special case, because the factor loadings are straight-arrow parameters, but the Satorra-Bentler principle does not apply because the variances of the factors are restricted (to equal one). There are siz x relevant parameters, $\lambda_1, \ldots, \lambda_6$, assessed in five tables where the bases distribution was not normal. Within each table, I'll conduct $6 \times 3 = 18$ pairwise tests and apply a Bonferroni correction. Here are the results.

- Table 5.24 is for an exponential base distribution with $n = 200$. There is significant under-coverage for all six factor loadings with all three methods. For each parameter, coverage of the sandwich and bootstrap intervals was significantly better (less bad) than coverage of the normal theory interval.

- Table 5.26 is another table for the exponential base distribution with $n = 200$. In this run, the two factors were independent. This did not affect the results. Again, all the methods suffered from significant under-coverage; the sandwich and bootstrap intervals were more accurate than normal theory for all the factor loadings, and not significantly different from each other.

- Table 5.25 is for an exponential base distribution with $n = 1,000$. With this sample size, only the normal theory intervals had significant under-coverage. The sandwich and bootstrap intervals were more accurate than normal theory for all factor loadings, and not significantly different from each other.

- Table 5.27 is for a scaled beta base distribution with $n = 200$. For this distribution, coverage was acceptable for all the factor loadings, for all three confidence intervals. With the Bonferroni correction for 18 tests, the normal theory interval had better coverage than the sandwich interval for one of the (indistinguishable) parameters: $\lambda_6$.

- Table 5.28 is for a scaled beta base distribution with $n = 500$. Coverage for all the factor loadings was acceptable, and there were no significant differences in coverage between methods.

The conclusion is that for a heavy-tailed distribution when the the Satorra-Bentler principle does not apply, normal theory standard errors are clearly too small. The sandwich and bootstrap standard errors are definitely better, and neither of them is better than the other. For a light-tailed non-normal distribution, all three methods are okay and there is no clear evidence of any difference in quality.

**Another special case: Independent exogenous variables** When exogenous variables are independent, it appears that normal theory standard errors perform well regardless of the distribution of the data. Is it actually *better* to use the normal theory standard errors in this case, or would the robust and bootstrap standard errors work just as well? The simulations include several examples where significance tests should be informative.

- For the double measurement regression model (5.5), there is a covariance of $\omega_{13}$ between the measurement errors $e_1$ and $e_3$, and a covariance of $\omega_{24}$ between the measurement errors $e_2$ and $e_4$. In the simulations, $\omega_{13} = 1$, while $\omega_{24}$ was set to zero for no particular reason — and the zero covariance was implemented the easy way, by making $e_2$ and $e_4$ independent. The surprisingly good performance of the normal theory intervals for $\omega_{24}$ compared to their dismal failure for $\omega_{13}$ is what led me to the general principle.

Anyway, there are four tables with non-normal data where coverage of the three intervals for $\omega_{24}$ can be compared: Table 5.9 (exponential base distribution, $n = 200$), Table 5.10 (exponential, $n = 1,000$), Table 5.11 (exponential, $n = 50$), and Table 5.12 (scaled beta distribution, $n = 200$).

- For the Dip Down Path model (5.10), there is one variation (Table 5.17, exponential with $n = 200$), in which the two observable exogenous variables are independent, and we can assess coverage of the parameter $\phi_{12} = 0$.

- In the Standardized two-factor model (5.14), there is a variation (Table 5.26, exponential with $n = 200$) where the factors are independent, and coverage of the correlation between factors $\phi_{12} = 0$ can be compared.

With a Bonferroni correction for the $6 \times 3 = 18$ pairwise comparisons, the only significant differences were

- For the Dip Down Path model with $Cov(x_1, x_2) = \phi_{12} = 0$ because $x_1$ and $x_2$ were independent, the normal theory interval (coverage = 0.951) and sandwich interval (coverage = 0.938) both did significantly better than the bootstrap (coverage = 0.922).

- For the standardized Two-Factor model with $Corr(F_1, F_2) = \phi_{12} = 0$ because $F_1$ and $F_2$ were independent, the bootstrap coverage of 0.933 was significantly better than the sandwich's 0.917.

In short, the performance of the normal theory standard errors was quite good when the variables involved were independent, but not significantly better than the other methods.

**Yet another special case: Reliabilities** For the Extra Response Variable Regression model (5.9), the tables (5.2 through 5.7) include empirical coverage for three reliabilities, which are functions of the other parameters. The reliability of $w$ is $\frac{\phi}{\phi+\omega}$, the reliability of $y_1$ is $\frac{\beta_1^2\phi}{\beta_1^2\phi+\psi_1}$, and the reliability of $y_2$ is $\frac{\beta_2^2\phi}{\beta_2^2\phi+\psi_2}$. These quantities depend on straight-arrow parameters as well as variance parameters, so they are a sort of hybrid.

For the exponential base distribution (tables 5.3, 5.4 and 5.5), the normal theory standard errors are clearly not robust, and we only want to know whether the sandwich or bootstrap intervals have better coverage. That's nine tests, one for each reliability in each table. For the beta base distribution (tables 5.6 and 5.7), the normal theory standard errors enjoy some robustness, and all three pairwise comparisons are of interest. That's three pairwise comparisons for three reliabilities in two tables — a total of 18 more tests. In all, there are 27 tests, to which a Bonferroni correction will be applied. For a joint significance level of 0.05, the Bonferroni critical value of $z$ is 3.11 rather than 1.96.

With the Bonferroni correction, the significant comparisons were

- Exponential base distribution, $n = 200$: Reliability of $y_2$. Bootstrap better than Sandwich ($z = -4.143$)

- Exponential base distribution, $n = 50$: Reliability of $w$. Bootstrap better than Sandwich ($z = -7.619$)
- Exponential base distribution, $n = 50$: Reliability of $y_1$. Bootstrap better than Sandwich ($z = -7.839$)
- Exponential base distribution, $n = 50$: Reliability of $y_2$. Bootstrap better than Sandwich ($z = -8.183$)
- Beta base distribution, $n = 50$: Reliability of $w$. Bootstrap better than Sandwich ($z = -4.468$)
- Beta base distribution, $n = 50$: Reliability of $y_1$. Bootstrap better than Sandwich ($z = -4.359$)
- Beta base distribution, $n = 50$: Reliability of $y_2$. Bootstrap better than Sandwich ($z = -4.308$)

The conclusion is that for these reliabilities, problems with the normal theory standard errors are limited to heavy-tailed distribution, and that the bootstrap has a clear advantage over the sandwich for smaller samples, even when the non-normal data are not heavy-tailed.

## Variance and covariance parameters

We will start with the Big Data model (5.15). The error variance parameters $\omega_j$ are all equal to one another (the true value is one). We will adopt the same analysis strategy leading to Table 5.36, except that since the normal theory standard errors are clearly not robust for the heavy-tailed exponential base distribution (see Tables 5.31 through 5.33), they are excluded from Table 5.37. For a heavy-tailed distribution the contest is between the sandwich and the bootstrap.

Table 5.37: Testing differences in coverage for $\omega_j$ parameters in the "Big Data" model (5.15)

| | Base | | Estimated Coverage | | | Normal vs. | Normal vs. | Sandwich vs. |
|---|---|---|---|---|---|---|---|---|
| Table | Distribution | $n$ | Normal | Sandwich | Bootstrap | Sandwich | Bootstrap | Bootstrap |
| 5.31 | Exponential | 200 | 0.6774 | 0.8820 | 0.8852 | | | $z = -6.37^*$ |
| 5.32 | Exponential | 1500 | 0.6827 | 0.9369 | 0.9376 | | | $z = -1.69$ |
| 5.33 | Exponential | 2000 | 0.6798 | 0.9396 | 0.9395 | | | $z = 0.26$ |
| 5.34 | Beta | 200 | 0.9312 | 0.9314 | 0.9295 | $z = -0.28$ | $z = 2.57$ | $z = 4.21^*$ |
| 5.35 | Beta | 500 | 0.9382 | 0.9421 | 0.9413 | $z = -7.97^*$ | $z = -5.69^*$ | $z = 1.83$ |

$*$ Greater than Bonferroni critical value of 2.77, correcting for 9 tests.

For the exponential with $n = 200$ in Table 5.37, coverage is very poor for both the sandwich and the bootstrap, but it's a trifle (and significantly) better for the bootstrap.

With adequate sample size, there is no difference, though of course both the bootstrap and sandwich beat the normal theory method. With $n = 200$ and the beta distribution, the sandwich beats the bootstrap by a bit, though neither one is awful, and normal theory is okay for this base distribution. With a larger sample size, the performance of both the sandwich and bootstrap improves, while normal theory does not improve and is beaten by both the sandwich and bootstrap.

The conclusion is that in this setting, the sandwich and the bootstrap both show an advantage over normal theory even for a light-tailed distribution, and there are not any real grounds for choosing between the sandwich and the bootstrap.

**The rest of the variance and covariance parameters** The last job in this section is to consider variance and covariance parameters for the other models, excluding covariances that equal zero because the exogenous variables involved are independent. There are a *lot* of these parameters, because they appear in multiple tables. Here's the strategy. One model will be treated at a time, Bonferroni-correcting all the comparisons (from multiple tables) for a given model. It's clear from results already reported that normal theory standard errors for variance (and most covariance) parameters are not robust for the exponential base distribution, so in these cases we'll just compare the classical robust methods to the bootstrap. Because this is a study of robustness, no comparisons will be carried out for normal data.

*Extra Response Variable Regression model (5.9)* The variance parameters are $\phi$, $\omega$, $\psi_1$ and $\psi_2$. The source tables are Table 5.3 (exponential base distribution, $n = 200$), Table 5.4 (exponential base distribution, $n = 1,000$), Table 5.5 (exponential base distribution, $n = 50$), Table 5.6 (beta base distribution, $n = 200$) and Table 5.7 (beta base distribution, $n = 50$). Twelve tests (just the sandwich versus bootstrap) were carried out for the exponential data, and twenty-four pairwise comparisons for the data based on a scaled beta distribution.

Bonferroni-correcting for 36 tests, the only significant comparisons were for the beta distribution with $n = 50$, where coverage of the normal theory interval was better than the bootstrap for $\omega$ (z = 3.288), $\psi_1$ (z = 3.860) and $\psi_2$ (z = 4.034).

*Double Measurement Regression model (5.5)* The parameter $\omega_{24} = Cov(e_2, e_4)$ was covered earlier as a special case; its value was zero because $e_2$ and $e_4$ were independent. The parameters used in this analysis are the variances $\phi$, $\psi$, $\omega_{11}$, $\omega_{22}$, $\omega_{33}$, $\omega_{44}$ and the non-zero covariance $\omega_{13}$. The source tables are Table 5.9 (exponential base distribution, $n = 200$), Table 5.10 (exponential base distribution, $n = 1,000$), Table 5.11 (exponential base distribution, $n = 50$) and Table 5.12 (beta base distribution, $n = 200$). Twenty-one tests (sandwich versus bootstrap, seven parameters in three tables) were carried out for the exponential data, and twenty-one pairwise comparisons in the single table for the beta.

Bonferroni-correcting for 42 tests, only one null hypothesis was rejected. For the exponential base distribution with $n = 50$ coverage of $\psi$ was better for the sandwich than for the bootstrap ($z = 4.48$).

*Dip Down Path model (5.10)* The variance parameters are $\phi_{11}$, $\phi_{22}$, $\psi_1$, $\psi_2$ and $Var(e) = \omega$. The covariance parameter $\phi_{12} = Cov(x_1, x_2)$ is also included, except for the data of Table 5.17, where $x_1$ and $x_2$ are independent. The source tables are Table 5.16 (exponential base distribution, $n = 200$), Table 5.17 (exponential base distribution, $n = 200$ with $\phi_{12} = 0$), Table 5.19 (exponential base distribution, $n = 1,000$), Table 5.20 (beta base distribution, $n = 200$) and Table 5.21 (beta base distribution, $n = 500$).

The covariance parameter $\phi_{12} = Cov(x_1, x_2) = 0$ is excluded for the data of Table 5.17. Thus, when the base distribution is exponential there are six parameters for two tables and five parameters for one table. Comparing just the sandwich and bootstrap coverage for these parameters gives 12+5=17 tests. For each of the two tables with a beta base distribution, there are three pairwise comparisons of methods for six parameters. That's another 36 tests.

With a Bonferroni correction for the 53 tests, three comparisons were significant. For the exponential base distribution with $n = 200$ coverage of $\omega$ was better for the sandwich than for the bootstrap ($z = 3.52$). For the exponential base distribution with $n = 200$ with $phi_{12} = 0$, the bootstrap did better than the sandwich for $\psi_1$ ($z = -4.80$), and worse for $\omega$ ($z = 7.597$).

*Standardized Two-factor model (5.14)* The variance parameters are $\omega_1$ through $\omega_6$, and there is also a covariance parameter $\phi_{12} = Cov(F_1, F_2)$, which is included except when the factors are independent. The source tables are Table 5.24 (exponential base distribution, $n = 200$), Table 5.25 (exponential base distribution, $n = 1,000$), Table 5.26 (exponential base distribution, $n = 200$ with $\phi_{12} = 0$), Table 5.27 (beta base distribution, $n = 200$) and Table 5.28 (beta base distribution, $n = 500$).

The covariance parameter $\phi_{12} = 0 = Cov(F_1, F_2)$ is excluded for the data of Table 5.26. This means that when the base distribution is exponential and the normal theory standard error is definitely not robust (except for $\phi_{12}$) when the factors are independent), there are seven parameters for two tables and six parameters for one table, for a total of 20 comparisons. For the beta base distribution, there are two tables with seven parameters, and three pairwise comparisons for each parameter. That's an additional 42 comparisons.

With a Bonferroni correction for the 62 tests, there were only two significant comparisons. With the exponential base distribution and $n = 200$ (Table!5.24), coverage of $\phi_{12}$ was better for the bootstrap than for the sandwich. In the other exponential table with $n = 200$, this time with independent factors (Table 5.26), the sandwich did better than the bootstrap for $\omega_5$.

*Big Data model (5.15)* Just for $\phi = Var(F)$, there are three comparisons of the sandwich to the bootstrap in Tables 5.31, 5.32 and 5.33, and six pairwise comparisons in Tables 5.34 and 5.35. With a Bonferroni correction for the 9 tests, there were no significant differences.

## 5.7 Tests of Fit

**Normal theory likelihood ratio test** The covariance matrix of the observable variables as a function of the model parameters is written $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\theta})$. The default test of fit for a structural equation model is the likelihood ratio test statistic comparing the likelihood at $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})$ to the likelihood at the unrestricted estimate $\widehat{\boldsymbol{\Sigma}}$. Repeating Expression 1.18 from page 171, the test statistic is

$$\begin{aligned} G^2 &= -2\log\frac{L\left(\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\right)}{L(\widehat{\boldsymbol{\Sigma}})} \\ &= n\left(tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}| - k\right). \end{aligned}$$

With $r$ parameters and $k$ observed variables, there are $m = k(k+1)/2$ unique variances and covariances. If the model is correct, the parameters are identifiable, $m > r$ and the data are multivariate normal, $G^2$ has a limiting chi-squared distribution with $m - r$ degrees of freedom.

**Satorra and Bentler's mean-corrected statistic** Suppose that the data are not multivariate normal, but the model is correct and the parmeters identifiable with $m > r$. In this case, asymptotic distribution of the $G^2$ statistic is not necessarily chi-squared. Instead, it's a weighted sum of independent chi-squares:

$$G^2 \approx \sum_{j=1}^{m-r} a_j t_j,$$

where the weights $a_1, \ldots, a_{m-r}$ are constants, and $t_1, \ldots, t_{m-r}$ are independent chi-squared random variables with one degree of freedom. Naturally, if all the weights equal one or converge to one, then asymptotically, $G^2 \sim \chi^2(m-r)$ as in the normal case. Otherwise, the large-sample distribution of $G^2$ is something nameless that is not chi-squared.

Recall that the expected value of a chi-squared random variable equals its degrees of freedom, so that if the data are normal, $E(G^2) \approx m - r$. If the data are not normal,

$$E(G^2) = \sum_{j=1}^{m-r} a_j E(t_j) = \sum_{j=1}^{m-r} a_j.$$

Especially if the $a_j$ constants trend to be bigger than one, the expected value of $G^2$ would be too large, and one would expect the null hypothesis of model correctness to be rejected too often. In any case, the chi-squared approximation to $G^2$ should be better if at least it had the right expected value.

In a different paper from the one on robust standard errors [54], Satorra and Bentler [55] observe that the constants $a_1, \ldots, a_{m-r}$ are the non-zero eigenvalues of the matrix $\mathbf{U}_0\mathbf{L}$, where

- The matrix $\mathbf{L}$ (Satorra and Bentler call it $\Gamma$) is the asymptotic covariance matrix of $\sqrt{n}(\widehat{\boldsymbol{\sigma}}_n - \boldsymbol{\sigma})$ — see Theorem A.1. Recall that $\boldsymbol{\sigma} = vech(\boldsymbol{\Sigma})$ and $\widehat{\boldsymbol{\sigma}}_n = vech(\widehat{\boldsymbol{\Sigma}}_n)$.
- $\mathbf{U}_0 = \mathbf{W} - \boldsymbol{\Delta}\left(\boldsymbol{\Delta}^\top\mathbf{W}\boldsymbol{\Delta}\right)^{-1}\boldsymbol{\Delta}^\top\mathbf{W}$.
- $\boldsymbol{\Delta} = \left[\frac{\partial\sigma_i(\boldsymbol{\theta})}{\partial\theta_j}\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}$, where $\boldsymbol{\theta}_0$ is the vector of true parameter values.

The matrix $\mathbf{W}$ is a very special *weight matrix*, as in the weighted least squares estimator (5.1). There is a complicated formula for $\mathbf{W}$ that will not be given here; see [55] and other sources including a 1974 paper by Browne [12]. This particular weight matrix yields a least-squares estimate that is asymptotically equivalent to the MLE. It has the remarkable property that as $n \to \infty$, the probability that the associated least-squares estimator is equal to the MLE goes to one.

Let $c = \frac{tr(\mathbf{U}_0\mathbf{L})}{m-r}$. Because the trace is the sum of eigenvalues, $E(G^2/c) = m - r$, the correct expected value if $G^2$ has a chi-squared distribution. Because $\mathbf{U}_0$ and $\mathbf{L}$ are functions of the unknown parameter vector $\boldsymbol{\theta}_0$, Satorra and Bentler propose using $\widehat{c} = \frac{tr\left(\widehat{\mathbf{U}}_0\widehat{\mathbf{L}}\right)}{m-r}$, where the estimation is mostly accomplished by evaluating $\mathbf{U}_0$ and $\mathbf{L}$ (which are functions of the true parameter $\boldsymbol{\theta}_0$) at the MLE, $\widehat{\boldsymbol{\theta}}_n$. The crucial matrix $\mathbf{L}$ also contains some third and fourth-order moments, which can be consistently estimated by the method of moments. The result is the *mean-corrected* fit statistic

$$G^2_m = \frac{G^2}{\widehat{c}}. \tag{5.16}$$

In lavaan, this is produced by `test = "satorra.bentler"`

**The mean and variance-corrected fit statistic** The variance of a chi-squared random variable equals twice the degrees of freedom, and it is possible to correct $G^2 \approx \sum_{j=1}^{m-r} a_j t_j$ so that the expected value goes to $m - r$ and at the same time, the variance goes to $2(m - r)$. Satorra and Bentler [55] took a stab at it and came up with a statistic having fractional degrees of freedom (essentially a Satterthwaite correction). Rosseel [56] (the creator of lavaan) endorses a later, more refined version in a 2010 paper by Asparouhov and Muthén [1]. They propose a *mean and variance-corrected* statistic

$$G^2_{mv} = \frac{1}{\widehat{a}}G^2 + \widehat{b}, \tag{5.17}$$

where $a = \sqrt{tr\left((\mathbf{U}_0\mathbf{L})^2\right)/(m-r)}$ and $b = (m-r)\left(1 - \frac{c}{a}\right)$. The $G^2_{mv}$ statistic has an expected value of approximately $m - r$, and a variance of approximately $2(m - r)$. In lavaan, $G^2_{mv}$ is produced by `test = "scaled.shifted"`.

**Bootstrap** uh

---

## 5.8 Tests of general hypotheses

# Chapter 6

# Parameter Identifiability: The Algebra of the Knowable

When

, and is called an *over-identifying restriction*. Such constraints arise whenever there are more identifying equations than unknowns. Even non-identified models may imply constraints — testable constraints — on the covariance matrix. This is an interesting side-issue we shall not pursue here. At any rate, an identifiable parameter in a model with more moment structure equations than unknowns is called *over-identifiable* or *over-identified*.

When there is more than one way to do it as in this example, the parameter is called *over-identifiable*. If there were the same number of equations and unknown parameters (with a unique solution), the parameter vector would be called *just identifiable*. When the parameter vector is just identifiable, the model is called *saturated*.

# Chapter 7

# Assessing Model Correctness

The LR test for goodness of fit is valid when the parameters are identifiable; here's why. If there are more covariance structure equations than model parameters, the model imposes a set of non-linear equality constraints on the elements of Sigma, functions of Sigma that are zero if the model is true. The LR test for goodness of fit is testing the null hypothesis that these constraints hold. But Wilks' (1938?) proof of the large-sample distribution of the test statistic applies to linear null hypotheses. Is there a problem? No. Since we have identifiability, the model parameters together with the functions form a 1-1 re-parameterization of the elements of Sigma. By the invariance principle, the test of the non-linear null hypothesis in the original moment space is the same as the test of a linear null hypothesis in the in the new space.

# Chapter 8

# Gröbner Basis

Most proofs of identifiability for structural equation models consist of solving systems of covariance structure equations, or somehow showing that a unique solution exists. These equations take the form of polynomials set to zero, or at worst ratios of polynomials set to zero. So, it makes sense to take a look at the mathematical state of the art for solving systems of polynomial equations. That state of the art is represented by the theory and technology of Gröbner Basis.

I am fairly sure that Min Lim [43] was the first to apply Gröbner basis technology to covariance structure equations, in her 2010 Ph.D. thesis at the University of Toronto. Others have followed and gotten credit for it. Her work remains unpublished.

The story begins with what I still call Christine's Theorem[1]. I had the privilege to be Min's thesis supervisor, and after a series of conversations, I did the traditional thing, and said "Here, go prove this." Here's what I asked her to show.

**Theorem 8.1 (Christine's Theorem)** *A system of covariance structure equations has either finitely many real solutions, or uncountably many.*

That is, if there are infinitely many solutions, they are uncountable. A countable infinity is ruled out. To me, this seems "obvious," though I don't know how to prove it[2]. Christine never proved it either, but she went away and came back with Gröbner Basis.

[1] At the time, Min called herself Christine. As a brilliant undergraduate, she was Min Lim. As a graduate student, she was Christine. When she finished her thesis, she became Min again.

[2] The intuition is this. Geometrically, the polynomials involved are curvy surfaces in $\mathbb{R}^{t+1}$, where $t$ is the number of parameters. Their intersection is another curvy surface. The set of parameter values for which all the polynomials equal zero is the intersection of the curvy intersection surface with the plane $\theta_1, \ldots, \theta_t, 0$. To have a countable infinity of solutions, the intersection surface would need to be tangent to that plane at countably many points. But these are polynomials, and they only have finitely many bends. It's impossible: QED. Well, I'm sort of convinced, but it's not a proof. Maybe this is well known in some circles, or just a homework problem. Unfortunately, I don't move in those circles.

# Chapter 9

# Categorical data

# Bibliography


[1] Asparouhov, T., and Muthén, B. (2010). Simple second order chi-square correction https://www.statmodel.com/download/WLSMV_new_chi21.pdf

[2] Bastlevsky, A. (1994) *Statistical factor analysis and related methods*. New York: Wiley.

[3] Amemiya, Y. and Anderson, T. W. (1990) Asymptotic chi-square tests for a large class of factor analysis models. *The annals of statistics*, 18, 1453-1463.

[4] Anderson, T. W. and Amemiya, Y. (1988) The asymptotic normal distribution of estimators in factor analysis under general conditions. *The annals of statistics*, 16, 759-771.

Bernaards, C. A. and Jennrich, R. I. (2005) Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis. *Educational and Psychological Measurement*, 65(5), 676-696.

[5] Bekker, P. A., Merckens A. and Wansbeek T. J. (1994) *Identification, Equivalent models and computer algebra*. California: Academic Press, Inc.

[6] Bentler, P. M. and Dijkstra, T. (1985). Efficient estimation via linearization in structural models. In P. R. Kirshnaiah (Ed.), *Multivariate Analysis VI* pp. 9-42). Amsterdam: North-Holland.

[7] Bernaards, C. A. and Jennrich, R. I. (2005) Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis. *Educational and Psychological Measurement*, 65(5), 676-696.

[8] Billingsley, P. (1979). *Probability and measure*. New York: Wiley.

[9] Blalock, H. M. Jr. (1961). *Causal inferences in non-experimental research*. Chapel Hill, NC: University of North Carolina Press.

[10] Bollen, K. A. (1989). *Structural equations with latent variables*. New York: Wiley.

[11] Box, G. E. P. and Draper, N. R. (1987). *Empirical Model-Building and Response Surfaces*. New York: Wiley.

[12] Browne, M. W. (1974). Generalized least squares estimators in the analysis of covariance structures. *South African.Statistica1 Journal* **8**, 1-24.

[13] Browne, M. W. (1984). Asymptotic Distribution-Free Methods in the Analysis of Covariance Structures. *British Journal of Mathematical and Statistical Psychology*, 37, 62-83.

[14] Brunner, J. and Austin, P. C. (2009). Inflation of Type I error rate in multiple regression when independent variables are measured with error. *Canadian Journal of Statistics*, 37, 33-46.

[15] Cattell, R. B. (1966). The scree test for number of factors. *Multivariate behavioural research*, 1, 245-276.

[16] Cattell, R. B., Eber, H. W. and Tatsuoka, M. M. (1970). Handbook for the Sixteen Personality Factor Questionnaire (16PF). New York: Plenum.

[17] Clark, P. J., Vandenberg, S. G., and Proctor, C. H. (1961). On the relationship of scores on certain psychological tests with a number of anthropometric characters and birth order in twins. *Human Biology*, 33, 163-180.

[18] Cox, D. R. (1961) Tests of separate families of hypotheses, in: *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, pp. 105-123. Berkeley, CA: University of California Press.

[19] Crowne, D. P. and Marlowe, D. (1964). *The approval motive: Studies in evaluative dependence*. New York: Wiley.

[20] Davison, A. C. (2008) *Statistical models*. New York: Cambridge University Press.

[21] Duncan, O. D. (1975) *Introduction to structural equation models*. New York: Academic Press.

[22] Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7, 1-26.

[23] Efron, B. and Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York: Chapman and Hall.

[24] Eysenck, H. J. (1947). *Dimensions of Personality*. London: Methuen.

[25] Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., and Strahan, E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. *Psychological Methods*, 4, 272-299.

[26] Finney, S. J. and DiStefano, C. (2006). Non-Normal and Categorical Data in Structural Equation Modeling. In G. R. Hancock and R. O. Mueller (eds.), *Structural Equation Modeling: A Second Course*, pp. 269-314. Greenwich, Connecticut: Information Age Publising.

[27] Fisher, F. M. (1966) *The identification problem in econometrics.* New York: McGraw-Hill.

[28] Harman, H. H. (1976). *Modern factor analysis* (3d ed., revised) Chicago: University of Chicago Press.

[29] Heywood, H. B. (1931) On finite sequences of real numbers. *Proceedings of the Royal Society of London*, 134, 486-501.

[30] Hochberg, Y., and Tamhane, A. C. (1987). *Multiple comparison procedures.* New York: Wiley.

[31] Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*. 30, 179185.

[32] Huber, P. J. (1967) The Behavior of Maximum Likelihood Estimates Under Nonstandard Conditions. in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pp. 221-233. Berkeley, CA: University of California Press.

[33] Jennrich, R. I. and Sampson, P. F. (1966) Rotation for simple loadings. *Psychometrika*, 31, 313-323.

[34] Johnson, R. A. and Wichern, D. W. (2019) *Applied multivariate statistical analysis, 6th ed.* Upper Saddle River, New Jersey: Prentice-Hall.

[35] Jöreskog, K.G. (1967). Some contributions to maximum likelihood factor analysis, *Psychometrika*, 32(4), 443-482.

[36] Jöreskog, K.G. (1969). A general approach to confirmatory maximum likelihood factor analysis, *Psychometrika*, 34(2), 183-202.

[37] Jöreskog, K.G. (1978). Structural Analysis of Covariance and Correlation Matrices, *Psychometrika*, 43(4), 443-477.

[38] Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23, 187-200.

[39] Kaiser, H. F. (1960). The application of e1ectronic computers to factor analysis. *Educational and psychological measurement*, 20, 141-151.

[40] Kullback, S. (1959) *Information Theory and Statistics*. New York: Wiley.

[41] Lawley, D. N. (1940). The estimation of factor loadings by the method of maximum likelihood. *Proceedings of the Royal Society of Edinburgh*, 60, 64-82.

[42] Lawley, D. N. and Maxwell, A. E. (1971). *Factor analysis as a statistical method*. London: Butterworths.

[43] Lim, M. (2010). *Gröbner basis and structural equation modeling*. Ph.D. thesis, Department of Statistics, University of Toronto.

[44] Lord, F.M., and Novick, M.R. (1968). *Statistical theories of mental test scores, with contributions by Alan Birnbaum*. Reading, MA: Addison-Wesley.

[45] Merkle, E. c., You, D. and Preacher, K. J. (2015) Testing non-nested structural equation models. arXiv:1402.6720v3 [stat.AP] 12 May 2015

[46] Plato (370 B.C.?) *The Republic*. Free translation by B. Jowett available through Project Gutenberg at http://www.gutenberg.org/etext/150.

[47] R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org.

[48] Rosseel, Y. (2012) lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48(2), 1-36. URL http://www.jstatsoft.org/v48/i02/.

[49] Orne, M. T. (1962). On the social psychology of the psychological experiment: With particular reference to demand characteristics and their implications. *American Psychologist*, 17, 776783.

[50] Osgood, C.E., Suci, G., and Tannenbaum, P. (1957). *The measurement of meaning*. Urbana, IL: University of Illinois Press.

[51] Revelle W. (2021) *psych: Procedures for Psychological, Psychometric, and Personality Research.* Northwestern University, Evanston, Illinois. R package version 2.1.3, `https://CRAN.R-project.org/package=psych`.

[52] Rosenthal, R. and Rosnow, R. (2009) *Artifacts in behavioural research: Robert Rosenthal and Ralph L. Rosnow's Classic Books* (originally published 1969, 1966, 1974). New York: Oxford.

[53] SageMath, the Sage Mathematics Software System (Version 8.0.0), The Sage Developers, 2011, https://www.sagemath.org.

[54] Satorra, A. and Bentler, P. M. (1990) Model conditions for asymptotic robustness in the analysis of linear relations. *Computational Statistics and Data Analysis*, 10, 235-249

[55] Satorra, A. and Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In Alexander von Eye and Clifford Clogg (Eds.), Latent variables analysis: Applications for developmantal research (pp. 399419). Thousand Oaks, California: Sage.

[56] Savalei, V. and Rosseel, Y. (2022) Computational options for standard errors and test statistics with incomplete normal and nonnormal data in SEM. *Structural Equation Modeling*, In press.

[57] Seber, G. A. F. (2008) *A matrix handbook for statisticians*. Hoboken, New Jersey: Wiley.

[58] Shapiro, A. (1985) Asymptotic theory of overparameterized structural models. *Journal of the American Statistical Association* 81, 142-149.

[59] Simonsohn, U., Nelson, L. D. and Simmons, J. P. (2014a). *P*-curve: A key to the file drawer. *Journal of experimental psychology: General*, 143, 534-547.

[60] Spearman, C. (1904) General intelligence, objectively determined and measured. *Americal Journal of Psychology* 15, 201-292.

[61] Steiner, M. and Grieder, S. (2020) EFAtools: An R package with fast and flexible implementations of exploratory factor analysis tools. *Journal of Open Source Software*, 5(53), 2521.

[62] Stock, J. H. and Trebbi, F. (2003) Who Invented Instrumental Variable Regression? *Journal of Economic Perspectives* 17, 177-194.

[63] Stuart, A. and Ord, K. J. (1991) *Kendall's advanced theory of statistics, Volume 2 (5th Edition)*. New York: Oxford University Press.

[64] Thurstone, L. L. (1947) *Multiplefactor analysis*. Chicago: University of Chicago Press.

[65] Vuong, Q. H. (1989) Likelihood ratio tests for model selection and non-nested hypotheses. Econometrica, 57, 307-333.

[66] Wald, A. (1943) Tests of Statistical Hypotheses Concerning Several Parameters When the Number of Observations is Large. *Transactions of the American Mathematical Society*, 54, 426-482.

[67] Wald, A. (1949) Note on the consistency of the maximum likelihood estimate. *Annals of mathematical statistics*, 20, 595-601.

[68] White, H. (1982a) Maximum likelihood estimation of misspecified models. *Econometrica*, 50, l-26.

[69] White, H. (1982) Regularity Conditions for Cox's Test of Non-Nested Hypotheses. *Journal of Econometrics*, 19, 301-318.

[70] Wilks, S. S. (1938) The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. *Annals of Mathematical Statistics*, 9, 60-62.

[71] Wright, Philip G. (1928) *The Tariff on Animal and Vegetable Oils*. New York: Macmillan.

[72] Wright, S. (1921) Correlation and causation. *Journal of agricultural research*, 20, 557-584.

[73] Wright, S. (1934) The method of path coefficients. *Annals of mathematical statistics*, 5, 161-215.

[74] Wikipedia contributors. (2020, April 15). Natural experiment. In Wikipedia, The Free Encyclopedia. Retrieved 19:07, June 29, 2020, from https://en.wikipedia.org/w/index.php?title=Natural_experiment&oldid=951075932

# Appendix A

# Review and Background Material

## A.1 Expected Value, Variance and Covariance (Review)

**Expected Value** Let $X$ be a random variable. If $X$ is continuous, the expected value is defined as

$$E(X) = \int_{-\infty}^{\infty} x\, f_{_X}(x)\, dx.$$

If $X$ is discrete, the formula is

$$E(X) = \sum_{x} x\, p_{_X}(x).$$

Conditional expectation uses these same formulas, only with conditional densities or probability mass functions.

Let $Y = g(X)$. The change of variables formula (a very big Theorem[1]) tells us

$$E(Y) = \int_{-\infty}^{\infty} y\, f_{_Y}(y)\, dy = \int_{-\infty}^{\infty} g(x)\, f_{_X}(x)\, dx \tag{A.1}$$

or, for discrete random variables

$$E(Y) = \sum_{y} y\, p_{_Y}(y) = \sum_{x} g(x)\, p_{_X}(x).$$

One useful function $g(x)$ is the *indicator function* for a set $A$. $I_A(x) = 1$ if $x \in A$, and $I_A(x) = 0$ if $x \notin A$. The expected value of an indicator function is just a probability

[1]The change of variables formula holds under very general circumstances; see for example Theorem 16.12 in Billingsley's *Probability and measure* [8]. It is extremely convenient and easy to apply, because there is no need to derive the probability distribution of $Y$. So for example the sets of values where $f_X(x) \neq 0$ and $f_Y(y) \neq 0$ (and therefore the regions over which you are integrating in expression (A.1)) may be different and you don't have to think about it. Furthermore, the function $g(x)$ is almost arbitrary. In particular, it need not be differentiable, a condition you would need if you tried to prove anything for the continuous case with ordinary calculus.

because, for discrete random variables,

$$E(I_A(X)) = \sum_x I_A(x)\, p_X(x) = \sum_{x \in A} p_X(x) = P(X \in A).$$

For continuous random variables, something similar happens; multiplication by $I_A(x)$ erases the density for $x \notin A$, and integration of the product from zero to infinity is just integration over the set $A$, yielding $P(X \in A)$.

Another useful function is a conditional expectation. If we write the conditional density

$$f_{Y|X}(y|X) = \frac{f_{X,Y}(X, y)}{f_X(X)}$$

with the capital letter $X$, we really mean it. $X$ is a random variable, not a constant, and for any fixed $y$, the conditional density is a random variable. The conditional expected value is another random variable $g(x)$:

$$E(Y|X) = \int_{-\infty}^{\infty} y\, f_{Y|X}(y|X)\, dy.$$

This may be a strange-looking function, but still it is a function, and one can take its expected value using the change of variables formula A.1.

$$E(E(Y|X)) = \int_{-\infty}^{\infty} g(x)\, f_X(x)\, dx = \int_{-\infty}^{\infty} E(Y|x)\, f_X(x)\, dx.$$

Provided $|E(Y)| < \infty$, order of integration or summation may be exchanged[2], and we have the double expectation formula:

$$E(Y) = E(E(Y|X)).$$

You will prove a slightly more general and useful version as an exercise.

The change of variables formula (A.1) still holds if $\mathbf{x}$ is a vector, or even if both $\mathbf{x}$ and $\mathbf{y}$ are vectors, and integration or summation is replaced by multiple integration or summation. So, for example if $\mathbf{x} = (X_1, X_2)^\top$ has joint density $f_{\mathbf{x}}(\mathbf{x}) = f_{X_1,X_2}(x_1, x_2)$ and $g(x_1, x_2) = a_1 x_1 + a_2 x_2$,

$$\begin{aligned} E(a_1 X_1 + a_2 X_2) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (a_1 x_1 + a_2 x_2) f_{X_1,X_2}(x_1, x_2)\, dx_1 dx_2 \\ &= a_1 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1 f_{X_1,X_2}(x_1, x_2)\, dx_1 dx_2 + a_2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_2 f_{X_1,X_2}(x_1, x_2)\, dx_1 dx_2 \\ &= a_1\, E(X_1) + a_2\, E(X_2). \end{aligned}$$

Using this approach, it is easy to establish the linearity of expected value

$$E\left(\sum_{j=1}^{m} a_j X_j\right) = \sum_{j=1}^{m} a_j E(X_j) \tag{A.2}$$

[2]By Fubini's Theorem. Again, Billingsley's *Probability and measure* [8] is a good source.

and other familiar properties.

The change of variables formula holds if the function of the random vector is just one of the variables. So, for example, since $g(x_1, x_2, \ldots x_p) = x_3$ is one possible function of $x_1, x_2, \ldots x_p$,

$$\begin{aligned} \int \cdots \int x_3 \, f(\mathbf{x}) \, d\mathbf{x} &= \int \cdots \int x_3 \, f(x_1, \ldots x_p) \, dx_1 \cdots dx_p \\ &= E(X_3). \end{aligned}$$

**Variance and Covariance** Denote $E(X)$ by $\mu_{_X}$. The variance of $X$ is defined as

$$Var(X) = E[(X - \mu_{_X})^2],$$

and the covariance of $X$ and $Y$ is defined as

$$Cov(X, Y) = E[(X - \mu_{_X})(Y - \mu_{_Y})].$$

It is sometimes useful to say that $Var(X) = Cov(X, X)$.

The *correlation* between $X$ and $Y$ is

$$Corr(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}. \tag{A.3}$$

**Linear combinations** Let $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$ be random variables, and define the linear combinations $L_1$ and $L_2$ by

$$\begin{aligned} L_1 &= a_1 X_1 + \cdots + a_{n_1} X_{n_1} = \sum_{i=1}^{n_1} a_i X_i, \text{ and} \\ L_2 &= b_1 Y_1 + \cdots + b_{n_2} Y_{n_2} = \sum_{i=1}^{n_2} b_i Y_i, \end{aligned}$$

where the $a_j$ and $b_j$ are constants. Then

$$cov(L_1, L_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} a_i b_j Cov(X_i, Y_j). \tag{A.4}$$

The proof of this useful result is left as an exercise. It says, for example, that

$$\begin{aligned} Cov(X_1, \beta_1 X_1 + \beta_2 X_2 + \epsilon) &= \beta_1 Cov(X_1, X_1) + \beta_2 Cov(X_1, X_2) + Cov(X_1, \epsilon) \\ &= \beta_1 Var(X_1) + \beta_2 Cov(X_1, X_2) + 0, \end{aligned}$$

assuming explanatory variables to be uncorrelated with error terms.

As the example suggests, usually the linear combinations are regression equations or regression-like equations. In words, (A.4) says that you just calculate the covariance of each term in $L_1$ with each term in $L_2$, and add. If the random variables are multiplied by coefficients, multiply each covariance by a product of coefficients.

**Exercises A.1**

A.1.1) Let $P\{X = x\} = \frac{x}{10}$ for $x = 1, 2, 3, 4$.

(a) Find $E(X)$. Show your work. My answer is 3.

(b) Find $E(X^2)$. Show your work. My answer is 10.

(c) Find $Var(X)$. Show your work. My answer is 1.

2. The random variable $x$ is uniformly distributed on the integers $\{-3, -2, -1, 0, 1, 2, 3\}$, meaning $P(x = -1) = P(x = -2) = \cdots = P(x = 3) = \frac{1}{7}$. Let $y = x^2$.

(a) What is $E(x)$? The answer is a number. Show your work.

(b) Calculate the variance of $x$. The answer is a number. Show your work.

(c) What is $P(y = -1)$?

(d) What is $P(y = 9)$?

(e) What is the probability distribution of $y$? Give the $y$ values with their probabilities.

(f) What is $E(y)$? The answer is a number. Did you already do this question?

3. The discrete random variables $x$ and $y$ have joint distribution

| | $x = 1$ | $x = 2$ | $x = 3$ |
|---|---|---|---|
| $y = 1$ | 2/12 | 3/12 | 1/12 |
| $y = 2$ | 2/12 | 1/12 | 3/12 |

(a) What is the marginal distribution of $x$? List the values with their probabilities.

(b) What is the marginal distribution of $y$? List the values with their probabilities.

(c) Are $x$ and $y$ independent? Answer Yes or No and show some work.

(d) Calculate $E(x)$. Show your work.

(e) Denote a "centered" version of $x$ by $x_c = x - E(x) = x - \mu_x$.

i. What is the probability distribution of $x_c$? Give the values with their probabilities.

ii. What is $E(x_c)$? Show your work.

iii. What is the probability distribution of $x_c^2$? Give the values with their probabilities.

iv. What is $E(x_c^2)$? Show your work.

(f) What is $Var(x)$? If you have been paying attention, you don't have to show any work.

(g) Calculate $E(y)$. Show your work.

(h) Calculate $Var(y)$. Show your work. You may use Question **??** if you wish.

(i) Calculate $Cov(x, y)$. Show your work. You may use Question **??** if you wish.

(j) Let $Z_1 = g_1(x, y) = x + y$. What is the probability distribution of $Z_1$? Show some work.

(k) Calculate $E(Z_1)$. Show your work.

(l) Do we have $E(x+y) = E(x)+E(y)$? Answer yes or No. Note that the answer *does not require independence.*

(m) Let $Z_2 = g_2(x, y) = xy$. What is the probability distribution of $Z_2$? List the values with their probabilities. Show some work.

(n) Calculate $E(Z_2)$. Show your work.

(o) Do we have $E(xy) = E(x)E(y)$? Answer yes or No. The connection to independence is established in Question **??**.

4. Here is another joint distribution. The point of this question is that you can have zero covariance without independence.

| | $x = 1$ | $x = 2$ | $x = 3$ |
|---|---|---|---|
| $y = 1$ | 3/12 | 1/12 | 3/12 |
| $y = 2$ | 1/12 | 3/12 | 1/12 |

(a) Calculate $Cov(x, y)$. Show your work. You may use Question **??** if you wish.

(b) Are $x$ and $y$ independent? Answer Yes or No and show some work.

A.1.5) Let $X \sim U(0, \theta)$, meaning for $f(x) = \frac{1}{\theta}$ for $0 < x < \theta$, and zero otherwise.

(a) Find $E(X)$. Show your work. My answer is $\frac{\theta}{2}$.

(b) Find $E(X^2)$. Show your work. My answer is $\frac{\theta^2}{3}$.

(c) Find $Var(X)$. Show your work. My answer is $\frac{\theta^2}{12}$.

A.1.6) Let $a$ be a constant and let $X$ be a random variable, either continuous or discrete (you choose). Use the change of variables formula (A.1) to show that $E(a) = a$.

A.1.7) Use the change of variables formula to prove the linear property given in expression (A.2). If you assume independence, you get a zero.

A.1.8) Let $X$ and $Y$ be discrete random variables, with $E(|h(X)|) < \infty$. Use the change of variables formula to prove $E(h(X)) = E[E(h(X)|Y)]$. Because $E(|h(X)|) < \infty$, Fubini's Theorem says that you are free to exchange order of summation. Is the result of this problem also true for continuous random variables? Why or why not?

A.1.9) Let $X$ and $Y$ be continuous random variables. Prove

$$P(X \in A) = \int_{-\infty}^{\infty} P(X \in A|Y = y)\, f_Y(y)\, dy.$$

This is sometimes called the *Law of Total Probability.* Is it also true for discrete random variables? Why or why not? Hint: use indicator functions.

A.1.10) Let $X$ and $Y$ be continuous random variables. Prove that if $X$ and $Y$ are independent, $E(XY) = E(X)E(Y)$. Draw an arrow to the place in your answer where you use independence, and write "This is where I use independence."

A.1.11) Let $X$ and $Y$ be *discrete* random variables. Prove that if $X$ and $Y$ are independent, $E(XY) = E(X)E(Y)$. Draw an arrow to the place in your answer where you use independence, and write "This is where I use independence."

A.1.12) Let $P(X = 0) = \frac{1}{2}$ and $P(X = -1) = P(X = 1) = \frac{1}{2}$, and let $Y = X^2$.

(a) Find $Cov(X, Y)$.

(b) Are $X$ and $Y$ independent? Answer Yes or No and prove your answer.

(c) Does zero covariance necessarily imply independence? Answer Yes or No.

Below the line, please use only expected value signs, not integrals or summation.

---

A.1.13) Show that $Cov[X, Y] = E[XY] - \mu_X \mu_Y$.

A.1.14) Show that if the random variables $X$ and $Y$ are independent, $Cov(X, Y) = 0$.

A.1.15) Show that $Var(X) = E[X^2] - \mu_X^2$.

A.1.16) In the following, $X$ and $Y$ are random variables, while $a$ and $b$ are fixed constants. For each pair of statements below, one is true and one is false (that is, not true in general). State which one is true, and prove it. Zero marks if you prove both statements are true, even if one of the proofs is correct.

(a) $Var(aX) = aVar(X)$, or $Var(aX) = a^2Var(X)$.

(b) $Var(aX + b) = a^2Var(X) + b^2$, or $Var(aX + b) = a^2Var(X)$.

(c) $Var(a) = 0$, or $Var(a) = a^2$.

(d) $Cov(aX, bY) = abCov(X, Y)$, or $Cov(aX, bY) = a^2Var(X) + b^2Var(Y) + 2abCov(X, Y)$.

(e) $Cov(X + a, Y + b) = Cov(X, Y) + ab$, or $Cov(X + a, Y + b) = Cov(X, Y)$.

(f) $Var(aX + bY) = a^2Var(X) + b^2Var(Y)$, or $Var(aX + bY) = a^2Var(X) + b^2Var(Y) + 2abCov(X, Y)$.

A.1.17) Let $X$ and $Y$ be random variables with $Cov(X, Y) = \sigma_{xy}$, while $a$ and $b$ are fixed constants.

(a) Find $Cov(X + a, Y + b)$

(b) Find $Cov(aX, bY)$.

A.1.18) Let $Y_1, \ldots, Y_n$ be numbers, and $\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$. Show

(a) $\sum_{i=1}^{n}(Y_i - \overline{Y}) = 0$

(b) $\sum_{i=1}^{n}(Y_i - \overline{Y})^2 = \sum_{i=1}^{n} Y_i^2 - n\overline{Y}^2$

(c) The sum of squares $Q_m = \sum_{i=1}^{n}(Y_i - m)^2$ is minimized when $m = \overline{Y}$.

A.1.19) Let $X_1, \ldots, X_n$ be random variables, let $a_1, \ldots, a_n$ be constants, and let $Y = \sum_{i=1}^{n} a_i X_i$. Derive a general formula for $Var(Y)$. Show your work. Now give the useful special case that applies when $X_1, \ldots, X_n$ are independent.

A.1.20) Let $X_1, \ldots, X_n$ be independent and identically distributed random variables (the standard model of a random sample with replacement). Denoting $E(X_i)$ by $\mu$ and $Var(X_i)$ by $\sigma^2$,

(a) Show $E(\overline{X}) = \mu$; that is, the sample mean is unbiased for $\mu$.

(b) Find $Var(\overline{X})$.

(c) Letting $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2 = \sigma^2$, show that $E(S^2) = \sigma^2$. That is, the sample variance is an unbiased estimator of the population variance. Consider adding and subtracting $\mu$.

A.1.21) Let $Y_1, \ldots, Y_n$ be independent random variables with $E(Y_i) = \mu$ and $Var(Y_i) = \sigma^2$ for $i = 1, \ldots, n$. For this question, please use definitions and familiar properties of expected value, not integrals.

(a) Find $E(\sum_{i=1}^{n} Y_i)$.

(b) Find $Var\left(\sum_{i=1}^{n} Y_i\right)$. Show your work. Draw an arrow to the place in your answer where you use independence, and write "This is where I use independence."

(c) Using your answer to the last question, find $Var(\overline{Y})$.

(d) A statistic $T$ is an *unbiased estimator* of a parameter $\theta$ if $E(T) = \theta$. Show that $\overline{Y}$ is an unbiased estimator of $\mu$. This is very quick.

(e) Let $a_1, \ldots, a_n$ be constants and define the linear combination $L$ by $L = \sum_{i=1}^{n} a_i Y_i$. What condition on the $a_i$ values makes $L$ an unbiased estimator of $\mu$?

(f) Is $\overline{Y}$ a special case of $L$? If so, what are the $a_i$ values?

(g) What is $Var(L)$?

22. In this regression model, the explanatory variables are random. Independently for $i = 1, \ldots, n$, let $Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i$, where $E(X_{i,1}) = \mu_1$, $E(X_{i,2}) = \mu_2$, $E(\epsilon_i) = 0$, $Var(\epsilon_i) = \sigma^2$, $\epsilon_i$ is independent of both $X_{i,1}$ and $X_{i,2}$, and

$$cov\begin{pmatrix} X_{i,1} \\ X_{i,2} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}$$

(a) What is $Var(Y_i)$? You may be able to just write down the answer.

(b) What is $Cov(X_{i,1}, Y_i)$? Show your work.

(c) What is $Cov(X_{i,2}, Y_i)$?

23. Prove Equation A.4.

## A.2 Matrix Calculations

### Basic definitions

A matrix is a rectangular array of numbers. They are usually denoted by boldface letters like $\mathbf{A}$, while scalars (1×1 matrices) are lower case in italics, like $a, b, c$. Matrices are also written by giving their $(i, j)$ element in brackets, like $\mathbf{A} = [a_{i,j}]$.

Let $\mathbf{A} = [a_{i,j}]$ and $\mathbf{B} = [b_{i,j}]$ be $n \times p$ matrices of constants, $\mathbf{C} = [c_{i,j}]$ be $p \times q$, and let $u$ and $v$ be scalars (1×1 matrices). Define

*Matrix addition*: $\mathbf{A} + \mathbf{B} = [a_{i,j} + b_{i,j}]$. The matrices must have the same number of rows and the same number of columns for addition (or subtraction) to be defined.

*Matrix multiplication*: $\mathbf{AC} = [\sum_{k=1}^{p} a_{i,k} c_{k,j}]$. Each element of $\mathbf{AC}$ is the inner product of a row of $\mathbf{A}$ and a column of $\mathbf{C}$. Thus, the number of columns in $\mathbf{A}$ must equal the number of rows in $\mathbf{C}$. Even if $q = n$ so that multiplication in both orders is well defined, in general $\mathbf{AC} \neq \mathbf{CA}$.

*Scalar multiplication*: $u\,\mathbf{A} = [u \cdot a_{i,j}]$

*Transposition*: $\mathbf{A}^\top = [a_{j,i}]$

*Symmetric matrix*: A square matrix $\mathbf{D}$ is said to be *symmetric* if $\mathbf{D} = \mathbf{D}^\top$.

*Identity matrix*: $\mathbf{I}$ is a square matrix with ones on the main diagonal and zeros elsewhere. $\mathbf{IC} = \mathbf{C}$ and $\mathbf{AI} = \mathbf{A}$.

*Diagonal matrix*: A square matrix $\mathbf{D} = [d_{i,j}]$ is said to be *diagonal* if $d_{i,j} = 0$ for $i \neq j$.

*Triangular matrix*: A square matrix $\mathbf{D} = [d_{i,j}]$ is said to be *triangular* if $d_{i,j} = 0$ for $i < j$ or $i > j$ (or both, in which case it is also diagonal).

Distributive laws for matrix and scalar multiplication are easy to establish and are left as exercises.

### Transpose of a product

The transpose of a product is the product of transposes, in the reverse order: $(\mathbf{AC})^\top = \mathbf{C}^\top \mathbf{A}^\top$.

## Linear independence

The idea behind linear independence of a collection of vectors (say, the columns of a matrix) is that none of them can be written as a linear combination of the others. Formally, let $\mathbf{X}$ be an $n \times p$ matrix of constants. The columns of $\mathbf{X}$ are said to be *linearly dependent* if there exists a $p \times 1$ matrix $\mathbf{v} \neq \mathbf{0}$ with $\mathbf{Xv} = \mathbf{0}$. We will say that the columns of $\mathbf{X}$ are linearly *independent* if $\mathbf{Xv} = \mathbf{0}$ implies $\mathbf{v} = \mathbf{0}$.

## Row and column rank

The *row rank* of a matrix is the number of linearly independent rows. The *column rank* is the number of linearly independent columns. The rank of a matrix is the minimum of the row rank and the column rank. Thus, the rank of a matrix cannot exceed the minimum of the number of rows and the number of columns.

## Matrix Inverse

Let $\mathbf{A}$ and $\mathbf{B}$ be square matrices of the same size. $\mathbf{B}$ is said to be the *inverse* of $\mathbf{A}$ and may be written $\mathbf{B} = \mathbf{A}^{-1}$. The definition is $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$. Thus, there are always two equalities to establish when you are showing that one matrix is the inverse of another. Matrix inverses have the following properties, which may be proved as exercises.

- If a matrix inverse exists, it is unique.
- $\mathbf{A}^{-1\top} = \mathbf{A}^{\top-1}$
- If the scalar $u \neq 0$, $(u\mathbf{A})^{-1} = \frac{1}{u}\mathbf{A}^{-1}$.
- Suppose that the square matrices $\mathbf{A}$ and $\mathbf{B}$ both have inverses. Then $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.
- If $\mathbf{A}$ is a $p \times p$ matrix, $\mathbf{A}^{-1}$ exists if and only if the rank of $\mathbf{A}$ equals $p$.

Sometimes the following formula for the inverse of a $2 \times 2$ matrix is useful:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{A.5}$$

In some cases the inverse of the matrix is its transpose. When $\mathbf{A}^{\top} = \mathbf{A}^{-1}$, the matrix $\mathbf{A}$ is said to be *orthogonal*, because the column (row) vectors are all at right angles (zero inner product). In addition, they all have length one, because the inner product of each column (row) with itself equals one.

## Positive definite matrices

The $n \times n$ matrix $\mathbf{A}$ is said to be *positive definite* if

$$\mathbf{v}^\top \mathbf{A}\mathbf{v} > 0 \tag{A.6}$$

for *all* $n \times 1$ vectors $\mathbf{v} \neq \mathbf{0}$. It is called *non-negative definite* (or sometimes positive semi-definite) if $\mathbf{v}^\top \mathbf{A}\mathbf{v} \geq 0$. Positive definiteness is a critical property of variance-covariance matrices, because it says that the variance of any linear combination is greater than zero. See (A.15) on page 549.

## Determinants

Let $\mathbf{A} = [a_{i,j}]$ be an $n \times n$ matrix, so that the following applies to square matrices. The *determinant* of $\mathbf{A}$, denoted $|\mathbf{A}|$, is defined as a *sum of signed elementary products*. An elementary product is a product of elements of $\mathbf{A}$ such that there is exactly one element from every row and every column. The "signed" part is determined as follows.

Let $S_n$ denote the set of all permutations of the set $\{1, \ldots, n\}$, and denote such a permutation by $\sigma = (\sigma_1, \ldots, \sigma_n)$. Each permutation may be obtained from $(1, \ldots, n)$ by a finite number of switches of numbers. If the number of switches required is even (this includes zero), let $\text{sgn}(\sigma) = +1$; if it is odd, let $\text{sgn}(\sigma) = -1$. Then,

$$|\mathbf{A}| = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma_i}. \tag{A.7}$$

Some properties of determinants are:

- $|\mathbf{AB}| = |\mathbf{A}|\,|\mathbf{B}|$
- $|\mathbf{A}^\top| = |\mathbf{A}|$
- $|\mathbf{A}^{-1}| = 1/|\mathbf{A}|$, and if $|\mathbf{A}| = 0$, $\mathbf{A}^{-1}$ does not exist.
- If $\mathbf{A} = [a_{i,j}]$ is triangular, $|\mathbf{A}| = \prod_{i=1}^{n} a_{i,i}$. That is, for triangular (including diagonal) matrices, the determinant is the product of the elements on the main diagonal.
- Adding a multiple of one row to another row of a matrix, or adding a multiple of a column to another column leaves the determinant unchanged.
- Exchanging any two rows or any two columns of a matrix multiplies the determinant by $-1$.
- Multiplying a single row or column by a constant multiplies the determinant by that constant, so that $|v\mathbf{A}| = v^n|\mathbf{A}|$

## Eigenvalues and eigenvectors

Let $\mathbf{A} = [a_{i,j}]$ be an $n \times n$ matrix, so that the following applies to square matrices. $\mathbf{A}$ is said to have an *eigenvalue* $\lambda$ and (non-zero) *eigenvector* $\mathbf{x}$ corresponding to $\lambda$ if

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}. \tag{A.8}$$

Note that $\lambda$ is a scalar and $\mathbf{x} \neq \mathbf{0}$ is an $n \times 1$ matrix, typically chosen so that it has length one. It is also possible and desirable to choose the eigenvectors so they are mutually perpendicular (the inner product of any two equals zero).

To solve the eigenvalue equation, write

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \Rightarrow \mathbf{A}\mathbf{x} - \lambda\mathbf{x} = \mathbf{A}\mathbf{x} - \lambda\mathbf{I}\mathbf{x} = (\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}.$$

If $(\mathbf{A} - \lambda\mathbf{I})^{-1}$ existed, it would be possible to solve for $\mathbf{x}$ by multiplying both sides on the left by $(\mathbf{A} - \lambda\mathbf{I})^{-1}$, yielding $\mathbf{x} = \mathbf{0}$. But the definition specifies $\mathbf{x} \neq \mathbf{0}$, so the inverse cannot exist for the definition of an eigenvalue to be satisfied. Since $(\mathbf{A} - \lambda\mathbf{I})^{-1}$ fails to exist precisely when the determinant $|\mathbf{A} - \lambda\mathbf{I}| = 0$, the eigenvalues are the $\lambda$ values that solve the determinantal equation

$$|\mathbf{A} - \lambda\mathbf{I}| = 0.$$

The left-hand side is a polynomial in $\lambda$, called the *characteristic polynomial*. If the matrix $\mathbf{A}$ is real-valued and also symmetric, then all its eigenvalues are guaranteed to be real-valued — a handy characteristic not generally true of solutions to polynomial equations. The eigenvectors can also be chosen to be real, and for our purposes they always will be.

One of the many useful properties of eigenvalues is that **the determinant is the product of the eigenvalues**:

$$|\mathbf{A}| = \prod_{i=1}^{n} \lambda_i$$

## Spectral decomposition of symmetric matrices

The *Spectral decomposition theorem*[3] says that every square and symmetric matrix $\mathbf{A} = [a_{i,j}]$ may be written

$$\mathbf{A} = \mathbf{C}\mathbf{D}\mathbf{C}^\top, \tag{A.9}$$

where the columns of $\mathbf{C}$ (which may also be denoted $\mathbf{x}_1, \ldots, \mathbf{x}_1$) are the eigenvectors of $\mathbf{A}$, and the diagonal matrix $\mathbf{D}$ contains the corresponding eigenvalues, which are guaranteed to be real numbers, sorted from largest to smallest.

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

[3]The version we will use is the original, due to the Baron Augustin-Louis Cauchy (1789-1857). This is the guy after whom the Cauchy distribution is named. He is also responsible for the rigorous use of epsilons and deltas in calculus, and for lots of other good things.

Because the eigenvectors are orthonormal, $\mathbf{C}$ is an orthogonal matrix; that is, $\mathbf{C}\mathbf{C}^\top = \mathbf{C}^\top\mathbf{C} = \mathbf{I}$.

The following shows how to get a spectral decomposition from R.

```
> help(eigen)
> A = rbind(c(-10,2),
+           c(2,5)) # Symmetric
> eigenA = eigen(A); eigenA
$values
[1]   5.262087 -10.262087

$vectors
          [,1]       [,2]
[1,] 0.1299328  0.9915228
[2,] 0.9915228 -0.1299328

> det(A)
[1] -54
> prod(eigenA$values)
[1] -54

> Lambda = diag(eigenA$values); Lambda
         [,1]      [,2]
[1,] 5.262087   0.00000
[2,] 0.000000 -10.26209

> P = eigenA$vectors; P
          [,1]       [,2]
[1,] 0.1299328  0.9915228
[2,] 0.9915228 -0.1299328

> P %*% Lambda %*% t(P) # Matrix multiplication
     [,1] [,2]
[1,]  -10    2
[2,]    2    5
```

Another way to express the spectral decomposition is

$$\mathbf{A} = \sum_{i=1}^{n} \lambda_i \mathbf{x}_i \mathbf{x}_i^\top, \tag{A.10}$$

where again, $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are the eigenvectors of $\mathbf{A}$, and $\lambda_1, \ldots, \lambda_n$ are the corresponding eigenvalues. It's a weighted sum of outer (not inner) products of the eigenvectors; the weights are the eigenvalues.

Continuing the R example, here is $\mathbf{x}_1\mathbf{x}_1^\top$. Notice how the diagonal elements add to one, as they must.

```
> eigenA$vectors[,1] %*% t(eigenA$vectors[,1])
           [,1]      [,2]
[1,] 0.01688253 0.1288313
[2,] 0.12883133 0.9831175
```

Reproducing (A.10) for completeness,

```
> prod1 = eigenA$vectors[,1] %*% t(eigenA$vectors[,1])
> prod2 = eigenA$vectors[,2] %*% t(eigenA$vectors[,2])
> eigenA$values[1]
[1] 5.262087
> eigenA$values[1]*prod1 + eigenA$values[2]*prod2
     [,1] [,2]
[1,]  -10    2
[2,]    2    5
> A
     [,1] [,2]
[1,]  -10    2
[2,]    2    5
```

## Real symmetric matrices

For a symmetric $n \times n$ matrix $\mathbf{A}$, the eigenvalues are all real numbers, and the eigenvectors can be chosen to be real, perpendicular (inner product zero), and of length one. If a real symmetric matrix is also non-negative definite, as a variance-covariance matrix must be, the following conditions are equivalent:

- Rows linearly independent
- Columns linearly independent
- Rank $= n$
- Positive definite
- Non-singular ($\mathbf{A}^{-1}$ exists)
- Determinant is non-zero
- All eigenvalues are strictly positive

Most of the equivalence is shown using the spectral decomposition theorem.

## Trace of a square matrix

The *trace* of a square matrix $\mathbf{A} = [a_{i,j}]$ is the sum of its diagonal elements. Write

$$tr(\mathbf{A}) = \sum_{i=1}^{n} a_{i,i}.$$

Properties like $tr(\mathbf{A}+\mathbf{B}) = tr(\mathbf{A})+tr(\mathbf{B})$ follow immediately from the definition. Perhaps less obvious is the following. Let $\mathbf{A}$ be an $r \times p$ matrix and $\mathbf{B}$ be a $p \times r$ matrix, so that the product matrices $\mathbf{AB}$ and $\mathbf{BA}$ are both defined. These two matrices are not necessarily equal; in fact, they need not even be the same size. But still,

$$tr(\mathbf{AB}) = tr(\mathbf{BA}). \tag{A.11}$$

To see this, write

$$\begin{aligned}
tr(\mathbf{AB}) &= tr\left(\left(\sum_{k=1}^{p} a_{i,k}b_{k,j}\right)\right) \\
&= \sum_{i=1}^{r}\sum_{k=1}^{p} a_{i,k}b_{k,i} \\
&= \sum_{k=1}^{p}\sum_{i=1}^{r} b_{k,i}a_{i,k} \\
&= \sum_{i=1}^{p}\sum_{k=1}^{r} b_{i,k}a_{k,i} \quad \text{(Switching } i \text{ and } k\text{)} \\
&= tr\left(\left(\sum_{k=1}^{r} b_{i,k}a_{k,j}\right)\right) \\
&= tr(\mathbf{BA})
\end{aligned}$$

Notice how the indices of summation $i$ and $k$ have been changed. This is legitimate, because for example $\sum_{i=1}^{r} c_i$ and $\sum_{k=1}^{r} c_k$ both mean $c_1 + \cdots + c_r$.

Also, from the spectral decomposition (A.10), the trace is the sum of the eigenvalues:

$$tr(\mathbf{A}) = \sum_{i=1}^{n} \lambda_i.$$

This follows easily using (A.11), but actually it applies to *any* square matrix; the matrix need not be symmetric.

## Similar matrices

The square matrix $\mathbf{B}$ is said to be *similar* to $\mathbf{A}$ if there is an invertible matrix $\mathbf{P}$ with $\mathbf{B} = \mathbf{P}^{-1}\mathbf{AP}$. If $\mathbf{B}$ is similar to $\mathbf{A}$, then of course $\mathbf{A}$ is similar to $\mathbf{B}$. By the spectral

decomposition theorem, any square symmetric matrix is similar to a diagonal matrix. In other words, it is "diagonalizable."

Similar matrices share important characteristics. If two matrices are similar,

- They have the same eigenvalues.
- Their eigenvectors are in general *not* the same.
- They have the same determinant.
- One matrix has an inverse if and only if the other one does.
- They have the same rank.
- They have the same trace.
- They have the same number of linearly independent eigenvectors associated with each distinct eigenvalue.
- They have the same characteristic polynomial.

## The *vech* notation

Sometimes, it is helpful to represent the non-redundant elements of a symmetric matrix in the form of a column vector. Let $\mathbf{A} = [a_{i,j}]$ be an $n \times n$ symmetric matrix. $\mathbf{A}$ has $\frac{n(n+1)}{2}$ non-redundant elements: say the main diagonal plus the upper triangular half. Then

$$vech(\mathbf{A}) = \begin{pmatrix} a_{1,1} \\ \vdots \\ a_{1,n} \\ a_{2,2} \\ \vdots \\ a_{2,n} \\ \vdots \\ a_{n,n} \end{pmatrix}.$$

The *vech* operation is distributive: $vech(A + B) = vech(A) + vech(B)$.

### Exercises A.2

A.2.1) Which statement is true?

(a) $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$

(b) $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{BA} + \mathbf{CA}$

(c) Both a and b

(d) Neither a nor b

A.2.2) Which statement is true?

(a) $a(\mathbf{B}+\mathbf{C}) = a\mathbf{B} + a\mathbf{C}$

(b) $a(\mathbf{B}+\mathbf{C}) = \mathbf{B}a + \mathbf{C}a$

(c) Both a and b

(d) Neither a nor b

A.2.3) Which statement is true?

(a) $(\mathbf{B}+\mathbf{C})\mathbf{A} = \mathbf{A}\mathbf{B} + \mathbf{A}\mathbf{C}$

(b) $(\mathbf{B}+\mathbf{C})\mathbf{A} = \mathbf{B}\mathbf{A} + \mathbf{C}\mathbf{A}$

(c) Both a and b

(d) Neither a nor b

A.2.4) Which statement is true?

(a) $(\mathbf{A}\mathbf{B})^\top = \mathbf{A}^\top\mathbf{B}^\top$

(b) $(\mathbf{A}\mathbf{B})^\top = \mathbf{B}^\top\mathbf{A}^\top$

(c) Both a and b

(d) Neither a nor b

A.2.5) Which statement is true?

(a) $\mathbf{A}^{\top\top} = \mathbf{A}$

(b) $\mathbf{A}^{\top\top\top} = \mathbf{A}^\top$

(c) Both a and b

(d) Neither a nor b

A.2.6) Suppose that the square matrices $\mathbf{A}$ and $\mathbf{B}$ both have inverses. Which statement is true?

(a) $(\mathbf{A}\mathbf{B})^{-1} = \mathbf{A}^{-1}\mathbf{B}^{-1}$

(b) $(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$

(c) Both a and b

(d) Neither a nor b

A.2.7) Which statement is true?

(a) $(\mathbf{A}+\mathbf{B})^\top = \mathbf{A}^\top + \mathbf{B}^\top$

(b) $(\mathbf{A}+\mathbf{B})^\top = \mathbf{B}^\top + \mathbf{A}^\top$

(c) $(\mathbf{A}+\mathbf{B})^\top = (\mathbf{B}+\mathbf{A})^\top$

(d) All of the above

(e) None of the above

A.2.8) Which statement is true?

(a) $tr(\mathbf{A}+\mathbf{B}) = tr(\mathbf{A}) + tr(\mathbf{B})$

(b) $tr(\mathbf{A}+\mathbf{B}) = tr(\mathbf{B}) + tr(\mathbf{A})$

(c) Both a and b

(d) Neither a nor b

A.2.9) Which statement is true?

(a) $a\,tr(\mathbf{B}) = tr(a\mathbf{B})$.

(b) $tr(\mathbf{B})a = tr(a\mathbf{B})$

(c) Both a and b

(d) Neither a nor b

A.2.10) Which statement is true?

(a) $(a+b)\mathbf{C} = a\mathbf{C} + b\mathbf{C}$

(b) $(a+b)\mathbf{C} = \mathbf{C}a + \mathbf{C}b$

(c) $(a+b)\mathbf{C} = \mathbf{C}(a+b)$

(d) All of the above

(e) None of the above

A.2.11) Let $\mathbf{A}$ and $\mathbf{B}$ be $2 \times 2$ matrices. Either

- Prove $\mathbf{AB} = \mathbf{BA}$, or
- Give a numerical example in which $\mathbf{AB} \neq \mathbf{BA}$

A.2.12) In the following, $\mathbf{A}$ and $\mathbf{B}$ are $n \times p$ matrices of constants, $\mathbf{C}$ is $p \times q$, $\mathbf{D}$ is $p \times n$ and $a, b, c$ are scalars. For each statement below, either prove it is true, or prove that it is not true in general by giving a counter-example. Small numerical counter-examples are best. To give an idea of the kind of proof required for most of these, denote element $(i, j)$ of matrix $\mathbf{A}$ by $[a_{i,j}]$.

(a) $\mathbf{A}+\mathbf{B} = \mathbf{B}+\mathbf{A}$

(b) $a(\mathbf{B}+\mathbf{C}) = a\mathbf{B} + a\mathbf{C}$

(c) $\mathbf{AC} = \mathbf{CA}$

(d) $(\mathbf{A}+\mathbf{B})^\top = \mathbf{A}^\top + \mathbf{B}^\top$

(e) $(\mathbf{AC})^\top = \mathbf{C}^\top\mathbf{A}^\top$

(f) $(\mathbf{A}+\mathbf{B})\mathbf{C} = \mathbf{AC}+\mathbf{BC}$

(g) $(\mathbf{AD})^{-1} = \mathbf{A}^{-1}\mathbf{D}^{-1}$

A.2.13) Recall that $\mathbf{A}$ symmetric means $\mathbf{A} = \mathbf{A}^\top$. Let $\mathbf{X}$ be an $n$ by $p$ matrix. Prove that $\mathbf{X}^\top\mathbf{X}$ is symmetric.

A.2.14) The formal definition of a matrix inverse is that an inverse of the matrix $\mathbf{A}$ (denoted $\mathbf{A}^{-1}$) is defined by two properties: $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ and $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$. If you want to prove that one matrix is the inverse of another using the definition, you'd have two things to show. This homework problem establishes that you only need to do it in one direction.

Let $\mathbf{A}$ and $\mathbf{B}$ be square matrices with $\mathbf{AB} = \mathbf{I}$. Show that $\mathbf{A} = \mathbf{B}^{-1}$ and $\mathbf{A} = \mathbf{B}^{-1}$. To make it easy, use well-known properties of determinants.

A.2.15) Prove that inverses are unique, as follows. Let $\mathbf{B}$ and $\mathbf{C}$ both be inverses of $\mathbf{A}$. Show that $\mathbf{B} = \mathbf{C}$.

A.2.16) Let $\mathbf{X}$ be an $n$ by $p$ matrix with $n \neq p$. Why is it incorrect to say that $(\mathbf{X}^\top\mathbf{X})^{-1} = \mathbf{X}^{-1}\mathbf{X}^{\top-1}$?

A.2.17) Suppose that the matrices $\mathbf{A}$ and $\mathbf{B}$ both have inverses. Prove that $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

A.2.18) Let $\mathbf{A}$ be a non-singular matrix. Prove $(\mathbf{A}^{-1})^\top = (\mathbf{A}^\top)^{-1}$.

A.2.19) Using $(\mathbf{A}^{-1})^\top = (\mathbf{A}^\top)^{-1}$, prove that the inverse of a symmetric matrix is also symmetric.

A.2.20) Let $\mathbf{A}$ be a square matrix with the determinant of $\mathbf{A}$ (denoted $|\mathbf{A}|$) equal to zero. What does this tell you about $\mathbf{A}^{-1}$? No proof is necessary here.

A.2.21) Let $\mathbf{a}$ be an $n \times 1$ matrix of real constants. How do you know $\mathbf{a}^\top\mathbf{a} \geq 0$?

A.2.22) Let $\mathbf{A}$ be an $n \times p$ matrix of real constants. Is it true that $\mathbf{A}^\top\mathbf{A} \geq 0$? Briefly explain.

A.2.23) Let $\mathbf{X}$ be an $n \times p$ matrix of constants. Recall the definition of linear independence. The columns of $\mathbf{X}$ are said to be *linearly dependent* if there exists $\mathbf{v} \neq \mathbf{0}$ with $\mathbf{Xv} = \mathbf{0}$. We will say that the columns of $\mathbf{X}$ are linearly *independent* if $\mathbf{Xv} = \mathbf{0}$ implies $\mathbf{v} = \mathbf{0}$.

(a) Show that if the columns of $\mathbf{X}$ are linearly dependent, then the columns of $\mathbf{X}^\top\mathbf{X}$ are also linearly dependent.

(b) Show that if the columns of $\mathbf{X}$ are linearly dependent, then the *rows* of $\mathbf{X}^\top\mathbf{X}$ are linearly dependent.

(c) Show that if the columns of $\mathbf{X}$ are linearly independent, then the columns of $\mathbf{X}^\top\mathbf{X}$ are also linearly independent. Use $\mathbf{a}^\top\mathbf{a} \geq 0$ and the definition of linear independence.

(d) Show that if $(\mathbf{X}^\top\mathbf{X})^{-1}$ exists, then the columns of $\mathbf{X}$ are linearly independent.

(e) Show that if the columns of $\mathbf{X}$ are linearly independent, then $\mathbf{X}^\top\mathbf{X}$ is positive definite. Does this imply the existence of $(\mathbf{X}^\top\mathbf{X})^{-1}$? Locate the rule in the text, and answer Yes or No.

A.2.24) Let $\mathbf{A}$ be a square matrix. Show that

(a) If $\mathbf{A}^{-1}$ exists, the columns of $\mathbf{A}$ are linearly independent.

(b) If the columns of $\mathbf{A}$ are linearly dependent, $\mathbf{A}^{-1}$ cannot exist.

A.2.25) Let $\mathbf{A}$ be a symmetric matrix, and $\mathbf{A}^{-1}$ exists. Show that $\mathbf{A}^{-1}$ is also symmetric.

A.2.26) The *trace* of a square matrix is the sum of its diagonal elements; we write $tr(\mathbf{A})$. Let $\mathbf{A}$ be $r \times c$ and $\mathbf{B}$ be $c \times r$. Show $tr(\mathbf{AB}) = tr(\mathbf{BA})$.

A.2.27) Recall that the square matrix $\mathbf{A}$ is said to have an eigenvalue $\lambda$ and corresponding eigenvector $\mathbf{x} \neq \mathbf{0}$ if $\mathbf{Ax} = \lambda\mathbf{x}$.

(a) Suppose that an eigenvalue of $\mathbf{A}$ equals zero. Show that the columns of $\mathbf{A}$ are linearly dependent.

(b) Suppose that the columns of $\mathbf{A}$ are linearly dependent. Show that $\mathbf{A}^{-1}$ does not exist.

(c) Suppose that the columns of $\mathbf{A}$ are linearly independent. Show that the eigenvalues of $\mathbf{A}$ are all non-zero.

(d) Suppose $\mathbf{A}^{-1}$ exists. Show that the eigenvalues of $\mathbf{A}^{-1}$ are the reciprocals of the eigenvalues of $\mathbf{A}$. What about the eigenvectors?

A.2.28) The (square) matrix $\mathbf{\Sigma}$ is said to be *positive definite* if $\mathbf{a}^\top\mathbf{\Sigma a} > 0$ for all vectors $\mathbf{a} \neq \mathbf{0}$. Show that the diagonal elements of a positive definite matrix are positive numbers. Hint: Choose the right vector $\mathbf{a}$.

A.2.29) Show that the eigenvalues of a positive definite matrix are strictly positive.

A.2.30) Recall the *spectral decomposition* of a real symmetric matrix (For example, a variance-covariance matrix). Any such matrix $\mathbf{\Sigma}$ can be written as $\mathbf{\Sigma} = \mathbf{CDC}^\top$, where $\mathbf{C}$ is a matrix whose columns are the (orthonormal) eigenvectors of $\mathbf{\Sigma}$, $\mathbf{D}$ is a diagonal matrix of the corresponding (non-negative) eigenvalues, and $\mathbf{C}^\top\mathbf{C} = \mathbf{CC}^\top = \mathbf{I}$.

(a) Let $\mathbf{\Sigma}$ be a real symmetric matrix with eigenvalues that are all strictly positive.

i. What is $\mathbf{D}^{-1}$?

ii. Show $\mathbf{\Sigma}^{-1} = \mathbf{CD}^{-1}\mathbf{C}^\top$. So, the inverse exists.

(b) Let the eigenvalues of $\mathbf{\Sigma}$ be non-negative.

i. What do you think $\mathbf{D}^{1/2}$ might be?

ii. Define $\mathbf{\Sigma}^{1/2}$ as $\mathbf{C}\mathbf{D}^{1/2}\mathbf{C}^\top$. Show $\mathbf{\Sigma}^{1/2}$ is symmetric.

iii. Show $\mathbf{\Sigma}^{1/2}\mathbf{\Sigma}^{1/2} = \mathbf{\Sigma}$.

iv. Show that if the columns of $\mathbf{\Sigma}$ are linearly independent, then the columns of $\mathbf{\Sigma}^{1/2}$ are also linearly independent.

(c) Now return to the situation where the eigenvalues of the square symmetric matrix $\mathbf{\Sigma}$ are all strictly positive. Define $\mathbf{\Sigma}^{-1/2}$ as $\mathbf{C}\mathbf{D}^{-1/2}\mathbf{C}^\top$, where the elements of the diagonal matrix $\mathbf{D}^{-1/2}$ are the reciprocals of the corresponding elements of $\mathbf{D}^{1/2}$.

i. Show that the inverse of $\mathbf{\Sigma}^{1/2}$ is $\mathbf{\Sigma}^{-1/2}$, justifying the notation.

ii. Show $\mathbf{\Sigma}^{-1/2}\mathbf{\Sigma}^{-1/2} = \mathbf{\Sigma}^{-1}$.

A.2.31) In the following, let $\mathbf{\Sigma}$ be a real symmetric matrix, so that its eigenvalues are all real.

(a) Suppose that$\mathbf{\Sigma}$ has an inverse. Using the definition of linear independence, show that the columns of $\mathbf{\Sigma}$ are linearly independent.

(b) Let the columns of $\mathbf{\Sigma}$ be linearly independent, and also let $\mathbf{\Sigma}$ be at least non-negative definite (as, for example, a variance-covariance matrix must be). Show that $\mathbf{\Sigma}$ is strictly positive definite.

A.2.32) Show that if the real symmetric matrix $\mathbf{\Sigma}$ is positive definite, then $\mathbf{\Sigma}^{-1}$ is also positive definite.

A.2.33) Using the spectral decomposition (A.10) and $tr(\mathbf{AB}) = tr(\mathbf{BA})$, show that the trace of a square symmetric matrix is the sum of its eigenvalues.

A.2.34) Recall that the square matrix $\mathbf{B}$ is said to be *similar* to $\mathbf{A}$ if there is an invertible matrix $\mathbf{P}$ with $\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$. Useing this definition, prove the following.

(a) Any square symmetric matrix is similar to a diagonal matrix.

(b) Similar matrices have the same eigenvalues, but their eigenvectors are not the same in greneral.

(c) Similar matrices have the same determinant.

(d) If two matrices are similar, one has an inverse if and only if the other one does.

(e) Similar matrices have the same rank.

(f) Similar matrices have the same trace.

# A.3 Random Vectors and Matrices

A *random matrix* is just a matrix of random variables. Their joint probability distribution is the distribution of the random matrix. Random matrices with just one column (say, $p$) may be called *random vectors*.

## Expected Value and Variance-Covariance

### Expected Value

The expected value of a matrix is defined as the matrix of expected values. Denoting the $p \times c$ random matrix $\mathbf{X}$ by $[X_{i,j}]$,

$$E(\mathbf{X}) = [E(X_{i,j})].$$

Immediately we have natural properties like

$$\begin{aligned} E(\mathbf{X}+\mathbf{Y}) &= E([X_{i,j}+Y_{i,j}]) \\ &= [E(X_{i,j}+Y_{i,j})] \\ &= [E(X_{i,j})+E(Y_{i,j})] \\ &= [E(X_{i,j})]+[E(Y_{i,j})] \\ &= E(\mathbf{X})+E(\mathbf{Y}). \end{aligned}$$

Let $\mathbf{A} = [a_{i,j}]$ be an $r \times p$ matrix of constants, while $\mathbf{X}$ is still a $p \times c$ random matrix. Then

$$\begin{aligned} E(\mathbf{AX}) &= E\left(\left(\sum_{k=1}^{p} a_{i,k}X_{k,j}\right)\right) \\ &= \left(E\left(\sum_{k=1}^{p} a_{i,k}X_{k,j}\right)\right) \\ &= \left(\sum_{k=1}^{p} a_{i,k}E(X_{k,j})\right) \\ &= \mathbf{A}E(\mathbf{X}). \end{aligned}$$

Similar calculations yield $E(\mathbf{XB}) = E(\mathbf{X})\mathbf{B}$, where $\mathbf{B}$ is a matrix of constants. This yields the useful formula

$$E(\mathbf{AXB}) = \mathbf{A}E(\mathbf{X})\mathbf{B}. \tag{A.12}$$

### Variance-Covariance Matrices

Let $\mathbf{X}$ be a $p \times 1$ random vector with $E(\mathbf{X}) = \boldsymbol{\mu}$. The *variance-covariance matrix* of $\mathbf{X}$ (sometimes just called the *covariance matrix*), denoted by $cov(\mathbf{X})$, is defined as

$$cov(\mathbf{X}) = E\left\{(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})^\top\right\}. \tag{A.13}$$

The covariance matrix $cov(\mathbf{X})$ is a $p \times p$ matrix of constants. To see exactly what it is, suppose $p = 3$. Then

$$\begin{aligned}
cov(\mathbf{X}) &= E\left\{ \begin{pmatrix} X_1-\mu_1 \\ X_2-\mu_2 \\ X_3-\mu_3 \end{pmatrix} \begin{pmatrix} X_1-\mu_1 & X_2-\mu_2 & X_3-\mu_3 \end{pmatrix} \right\} \\
&= E\left\{ \begin{pmatrix} (X_1-\mu_1)^2 & (X_1-\mu_1)(X_2-\mu_2) & (X_1-\mu_1)(X_3-\mu_3) \\ (X_2-\mu_2)(X_1-\mu_1) & (X_2-\mu_2)^2 & (X_2-\mu_2)(X_3-\mu_3) \\ (X_3-\mu_3)(X_1-\mu_1) & (X_3-\mu_3)(X_2-\mu_2) & (X_3-\mu_3)^2 \end{pmatrix} \right\} \\
&= \begin{pmatrix} E\{(X_1-\mu_1)^2\} & E\{(X_1-\mu_1)(X_2-\mu_2)\} & E\{(X_1-\mu_1)(X_3-\mu_3)\} \\ E\{(X_2-\mu_2)(X_1-\mu_1)\} & E\{(X_2-\mu_2)^2\} & E\{(X_2-\mu_2)(X_3-\mu_3)\} \\ E\{(X_3-\mu_3)(X_1-\mu_1)\} & E\{(X_3-\mu_3)(X_2-\mu_2)\} & E\{(X_3-\mu_3)^2\} \end{pmatrix} \\
&= \begin{pmatrix} cov(X_1) & Cov(X_1,X_2) & Cov(X_1,X_3) \\ Cov(X_1,X_2) & cov(X_2) & Cov(X_2,X_3) \\ Cov(X_1,X_3) & Cov(X_2,X_3) & cov(X_3) \end{pmatrix}.
\end{aligned}$$

So, the covariance matrix $cov(\mathbf{X})$ is a $p \times p$ symmetric matrix with variances on the main diagonal and covariances on the off-diagonals.

The matrix of covariances between two random vectors may also be written in a convenient way. Let $\mathbf{X}$ be a $p \times 1$ random vector with $E(\mathbf{X}) = \boldsymbol{\mu}_x$ and let $\mathbf{Y}$ be a $q \times 1$ random vector with $E(\mathbf{Y}) = \boldsymbol{\mu}_y$. The $p \times q$ matrix of covariances between the elements of $\mathbf{X}$ and the elements of $\mathbf{Y}$ is

$$cov(\mathbf{X}, \mathbf{Y}) = E\left\{(\mathbf{X} - \boldsymbol{\mu}_x)(\mathbf{Y} - \boldsymbol{\mu}_y)^\top\right\}. \tag{A.14}$$

The following rule is analogous to $Var(a\,X) = a^2\,Var(X)$ for scalars. Let $\mathbf{X}$ be a $p \times 1$ random vector with $E(\mathbf{X}) = \boldsymbol{\mu}$ and $cov(\mathbf{X}) = \boldsymbol{\Sigma}$, while $\mathbf{A} = [a_{i,j}]$ is an $r \times p$ matrix of constants. Then

$$\begin{aligned}
cov(\mathbf{AX}) &= E\left\{(\mathbf{AX} - \mathbf{A}\boldsymbol{\mu})(\mathbf{AX} - \mathbf{A}\boldsymbol{\mu})^\top\right\} \\
&= E\left\{\mathbf{A}(\mathbf{X} - \boldsymbol{\mu})\left(\mathbf{A}(\mathbf{X} - \boldsymbol{\mu})\right)^\top\right\} \\
&= E\left\{\mathbf{A}(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^\top\mathbf{A}^\top\right\} \\
&= \mathbf{A}E\{(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^\top\}\mathbf{A}^\top \\
&= \mathbf{A}cov(\mathbf{X})\mathbf{A}^\top \\
&= \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top
\end{aligned} \tag{A.15}$$

Similarly,

$$\begin{aligned} cov(\mathbf{AX}, \mathbf{BY}) &= E\left\{(\mathbf{AX} - \mathbf{A}\boldsymbol{\mu}_x)(\mathbf{BY} - \mathbf{B}\boldsymbol{\mu}_y)^\top\right\} \\ &= E\left\{\mathbf{A}(\mathbf{X} - \boldsymbol{\mu})\left(\mathbf{B}(\mathbf{Y} - \boldsymbol{\mu}_y)\right)^\top\right\} \\ &= E\left\{\mathbf{A}(\mathbf{X} - \boldsymbol{\mu})(\mathbf{Y} - \boldsymbol{\mu}_y)^\top\mathbf{B}^\top\right\} \\ &= \mathbf{A}E\{(\mathbf{X} - \boldsymbol{\mu})(\mathbf{Y} - \boldsymbol{\mu}_y)^\top\}\mathbf{B}^\top \\ &= \mathbf{A}cov(\mathbf{X}, \mathbf{Y})\mathbf{B}^\top \\ &= \mathbf{A}\boldsymbol{\Sigma}_{xy}\mathbf{B}^\top \end{aligned} \tag{A.16}$$

For scalars, $Var(X + b) = Var(X)$, and the same applies to vectors. Covariances are also unaffected by adding a constant; this amounts to shifting the whole joint distribution by a fixed amount, which has no effect on relationships among variables. So, the following rule is "obvious." Let $\mathbf{X}$ be a $p \times 1$ random vector with $E(\mathbf{X}) = \boldsymbol{\mu}$ and let $\mathbf{b}$ be a $p \times 1$ vector of constants. Then $cov(\mathbf{X} + \mathbf{b}) = cov(\mathbf{X})$. To see this, note $E(\mathbf{X} + \mathbf{b}) = \boldsymbol{\mu} + \mathbf{b}$ and write

$$\begin{aligned} cov(\mathbf{X} + \mathbf{b}) &= E\left\{(\mathbf{X} + \mathbf{b} - (\boldsymbol{\mu} + \mathbf{b}))(\mathbf{X} + \mathbf{b} - (\boldsymbol{\mu} + \mathbf{b}))^\top\right\} \\ &= E\left\{(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^\top\right\} \\ &= cov(\mathbf{X}) \end{aligned} \tag{A.17}$$

A similar rule apples to $cov(\mathbf{X} + \mathbf{b}, \mathbf{Y} + \mathbf{c})$. A direct calculation is not even necessary, though it is a valuable exercise. Think of stacking $\mathbf{X}$ and $\mathbf{Y}$ one on top of another, to form a bigger random vector. Then,

$$cov\begin{pmatrix}\mathbf{X}\\ \hline \mathbf{Y}\end{pmatrix} = \left(\begin{array}{c|c} cov(\mathbf{X}) & cov(\mathbf{X}, \mathbf{Y}) \\ \hline cov(\mathbf{X}, \mathbf{Y})^\top & cov(\mathbf{Y}) \end{array}\right).$$

This is an example of a *partitioned matrix* – a matrix of matrices. At any rate, it is clear from (A.17) that adding a stack of constant vectors to the stack of random vectors has no effect upon the (partitioned) covariance matrix, and in particular no effect upon $cov(\mathbf{X}, \mathbf{Y})$.

**Linear combinations** In a direct analogy to (A.4) on page 530, let $\mathbf{X}_1, \ldots, \mathbf{X}_{n_1}$ and $\mathbf{Y}_1, \ldots, \mathbf{Y}_{n_2}$ be random vectors, and define the linear combinations $\mathbf{L}_1$ and $\mathbf{L}_2$ by

$$\begin{aligned} \mathbf{L}_1 &= \mathbf{A}_1\mathbf{X}_1 + \cdots + \mathbf{A}_{n_1}\mathbf{X}_{n_1} = \sum_{i=1}^{n_1}\mathbf{A}_i\mathbf{X}_i, \text{ and} \\ \mathbf{L}_2 &= \mathbf{B}_1\mathbf{Y}_1 + \cdots + \mathbf{B}_{n_2}\mathbf{Y}_{n_2} = \sum_{i=1}^{n_2}\mathbf{B}_i\mathbf{Y}_i, \end{aligned}$$

where the $\mathbf{A}_j$ and $\mathbf{B}_j$ are matrices of constants. It is assumed that the dimensions of the matrices allow the operations to be carried out. For example, the $\mathbf{A}_j$ all must have

the same number of rows, and the $\mathbf{B}_j$ must have the same number of rows. The result analogous to (A.4) is

$$cov(\mathbf{L}_1, \mathbf{L}_2) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \mathbf{A}_i cov(\mathbf{X}_i, \mathbf{Y}_j)\mathbf{B}_j^\top. \tag{A.18}$$

In words, (A.18) says that you just calculate the covariance matrix of each term in $\mathbf{L}_1$ with each term in $\mathbf{L}_2$ and add, treating the constant matrices as in (A.16).

To prove (A.18),

$$\begin{aligned}
cov(\mathbf{L}_1, \mathbf{L}_2) &= E\{(\mathbf{L}_1 - E(\mathbf{L}_1))\,(\mathbf{L}_2 - E(\mathbf{L}_2))^\top\} \\
&= E\left\{\left(\sum_{i=1}^{n_1}\mathbf{A}_i\mathbf{X}_i - \sum_{i=1}^{n_1}\mathbf{A}_i E(\mathbf{X}_i)\right)\left(\sum_{i=1}^{n_2}\mathbf{B}_i\mathbf{Y}_i - \sum_{i=1}^{n_2}\mathbf{B}_i E(\mathbf{Y}_i)\right)^\top\right\} \\
&= E\left\{\left(\sum_{i=1}^{n_1}\mathbf{A}_i\,(\mathbf{X}_i - E(\mathbf{X}_i))\right)\left(\sum_{i=1}^{n_2}\mathbf{B}_i\,(\mathbf{Y}_i - E(\mathbf{Y}_i))\right)^\top\right\} \\
&= E\left\{\left(\sum_{i=1}^{n_1}\mathbf{A}_i\,(\mathbf{X}_i - E(\mathbf{X}_i))\right)\left(\sum_{i=1}^{n_2}(\mathbf{Y}_i - E(\mathbf{Y}_i))^\top\,\mathbf{B}_i^\top\right)\right\} \\
&= E\left\{\sum_{i=1}^{n_1}\sum_{i=1}^{n_2}\mathbf{A}_i\,(\mathbf{X}_i - E(\mathbf{X}_i))\,(\mathbf{Y}_i - E(\mathbf{Y}_i))^\top\,\mathbf{B}_i^\top\right\} \\
&= \sum_{i=1}^{n_1}\sum_{i=1}^{n_2}\mathbf{A}_i E\left\{(\mathbf{X}_i - E(\mathbf{X}_i))\,(\mathbf{Y}_i - E(\mathbf{Y}_i))^\top\right\}\mathbf{B}_i^\top \\
&= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\mathbf{A}_i cov(\mathbf{X}_i, \mathbf{Y}_j)\mathbf{B}_j^\top \qquad \blacksquare
\end{aligned}$$

**Exercises A.3** This exercise set has an unusual feature. *Some of the questions ask you to prove things that are false.* That is, they are not true in general. In such cases, just write "The statement is false," and give a brief explanation to make it clear that you are not just guessing. The explanation is essential for full marks. A small counter-example is always good enough.

A.3.1) Let $\mathbf{X} = [X_j]$ be a random matrix. Show $E(\mathbf{X}^\top) = E(\mathbf{X})^\top$.

A.3.2) Let $\mathbf{X}$ and $\mathbf{Y}$ be random matrices of the same dimensions. Show $E(\mathbf{X} + \mathbf{Y}) = E(\mathbf{X}) + E(\mathbf{Y})$. Recall the definition $E(\mathbf{Z}) = [E(Z_{i,j})]$.

A.3.3) Let $\mathbf{X}$ be a random matrix, and $\mathbf{B}$ be a matrix of constants. Show $E(\mathbf{XB}) = E(\mathbf{X})\mathbf{B}$. Recall the definition $\mathbf{AB} = [\sum_k a_{i,k}b_{k,j}]$.

A.3.4) Let $\mathbf{X}$ be a $p \times 1$ random vector. Starting with Definition (A.13) on page 548, prove $cov(\mathbf{X}) = \mathbf{0}$..

A.3.5) Let the $p \times 1$ random vector $\mathbf{X}$ have expected value $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, and let $\mathbf{A}$ be an $m \times p$ matrix of constants. Prove that the variance-covariance matrix of $\mathbf{AX}$ is either

- $\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top$, or
- $\mathbf{A}^2\boldsymbol{\Sigma}$.

Pick one and prove it. Start with the definition of a variance-covariance matrix (A.13) on page 548.

A.3.6) If the $p \times 1$ random vector $\mathbf{X}$ has mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, show $\boldsymbol{\Sigma} = E(\mathbf{X}\mathbf{X}^\top) - \boldsymbol{\mu}\boldsymbol{\mu}^\top$.

A.3.7) Starting with Definition (A.14) on page 549, show $cov(\mathbf{X}, \mathbf{Y}) = cov(\mathbf{Y}, \mathbf{X})$..

A.3.8) Starting with Definition (A.14) on page 549, show $cov(\mathbf{X}, \mathbf{Y}) = E(\mathbf{X}\mathbf{Y}^\top) - \boldsymbol{\mu}_x\boldsymbol{\mu}_y^\top$.

A.3.9) Starting with Definition (A.14) on page 549, show $cov(\mathbf{X}, \mathbf{Y}) = \mathbf{0}$..

A.3.10) Let $\mathbf{X}$ be a $p \times 1$ random vector with expected value $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, and let $\mathbf{v}$ be a $p \times 1$ vector of constants.

(a) Let the scalar random variable $Y = \mathbf{v}^\top\mathbf{X}$. What is $Var(Y)$? Use this to prove tell you that *any* variance-covariance matrix must be positive semi-definite. (See the definition on Page 537.)

(b) Using the definition of an eigenvalue (A.8) on Page 538, show that eigenvalues of a variance-covariance matrix cannot be negative[4].

(c) How do you know that the determinant of a variance-covariance matrix must be greater than or equal to zero? The answer is one short sentence.

(d) Let $X$ and $Y$ be scalar random variables. Using what you have shown about the determinant, show $-1 \leq Corr(X, Y) \leq 1$. See the definition of a correlation on Page 530 if necessary. You have just proved the Cauchy-Schwarz inequality using probability tools.

A.3.11) Let the $p \times 1$ random vector $\mathbf{X}$ have mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$, and let $\mathbf{c}$ be a $p \times 1$ vector of constants. Find $cov(\mathbf{X} + \mathbf{c})$. Show your work, starting with the definition (A.13). Don't use the centering rule yet.

A.3.12) Let $\mathbf{X}$ be a $p \times 1$ random vector with mean $\boldsymbol{\mu}_x$ and variance-covariance matrix $\boldsymbol{\Sigma}_x$, and let $\mathbf{Y}$ be a $q \times 1$ random vector with mean $\boldsymbol{\mu}_y$ and variance-covariance matrix $\boldsymbol{\Sigma}_y$. Recall that $cov(\mathbf{X}, \mathbf{Y})$ is the $p \times q$ matrix $cov(\mathbf{X}, \mathbf{Y}) = E\left((\mathbf{X} - \boldsymbol{\mu}_x)(\mathbf{Y} - \boldsymbol{\mu}_y)^\top\right)$. Don't use the centering rule yet.

(a) What is the $(i, j)$ element of $cov(\mathbf{X}, \mathbf{Y})$?

[4]This property of covariance matrices can sometimes be used to detect problems with the numerical estimation of structural equation models.

(b) Find an expression for $cov(\mathbf{X}+\mathbf{Y})$ in terms of $\mathbf{\Sigma}_x$, $\mathbf{\Sigma}_y$ and $cov(\mathbf{X},\mathbf{Y})$. Show your work.

(c) Simplify further for the special case where $Cov(X_i, Y_j) = 0$ for all $i$ and $j$.

(d) Let $\mathbf{c}$ be a $p \times 1$ vector of constants and $\mathbf{d}$ be a $q \times 1$ vector of constants. Find $cov(\mathbf{X}+\mathbf{c}, \mathbf{Y}+\mathbf{d})$. Show your work.

A.3.13) Prove (**??**). This is the *basis* of the centering rule, so you are not allowed to use the centering rule.

A.3.14) Use the centering rule to show $cov(\mathbf{AX}+\mathbf{BY}) = \mathbf{A}cov(\mathbf{X})\mathbf{A}^\top + \mathbf{B}cov(\mathbf{Y})\mathbf{B}^\top$..

A.3.15) Use the centering rule to find $cov(\mathbf{AX}+\mathbf{BY}+\mathbf{c})$. What do you need to specify about the dimensions of the matrices for this to be true?

A.3.16) Write down $cov(\mathbf{AX}+\mathbf{BY})$ for the case where $\mathbf{X}$ and $\mathbf{Y}$ are independent. There is no need to show any work.

A.3.17) Use the centering rule to find $cov(\mathbf{AX}+\mathbf{c}, \mathbf{BX}+\mathbf{d})$. Must $\mathbf{A}$ and $\mathbf{B}$ have the same number of rows?

A.3.18) Let $X_1, \ldots, X_n$ be scalar random variables. Use the centering rule to show

$$Var\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} Var(X_i) + \sum_{i \neq j} Cov(X_i, X_j).$$

## A.4 The Multivariate Normal Distribution

The $p \times 1$ random vector $\mathbf{X}$ is said to have a *multivariate normal distribution*, and we write $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \mathbf{\Sigma})$, if $\mathbf{X}$ has (joint) density

$$f(\mathbf{x}) = \frac{1}{|\mathbf{\Sigma}|^{\frac{1}{2}}(2\pi)^{\frac{p}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{A.19}$$

where $\boldsymbol{\mu}$ is $p \times 1$ and $\mathbf{\Sigma}$ is $p \times p$ symmetric and positive definite. Positive definite means that for any non-zero $p \times 1$ vector $\mathbf{a}$, we have $\mathbf{a}^\top \mathbf{\Sigma} \mathbf{a} > 0$.

- Since the one-dimensional random variable $Y = \sum_{i=1}^{p} a_i X_i$ may be written as $Y = \mathbf{a}^\top \mathbf{X}$ and $Var(Y) = cov(\mathbf{a}^\top \mathbf{X}) = \mathbf{a}^\top \mathbf{\Sigma} \mathbf{a}$, it is natural to require that $\mathbf{\Sigma}$ be positive definite. All it means is that every non-zero linear combination of $\mathbf{X}$ values has a positive variance.

- $\mathbf{\Sigma}$ positive definite is equivalent to $\mathbf{\Sigma}^{-1}$ positive definite.

The multivariate normal reduces to the univariate normal when $p = 1$. Other properties of the multivariate normal include the following.

1. $E(\mathbf{X}) = \boldsymbol{\mu}$

2. $cov(\mathbf{X}) = \boldsymbol{\Sigma}$

3. If $\mathbf{c}$ is a vector of constants, $\mathbf{X} + \mathbf{c} \sim N_p(\mathbf{c} + \boldsymbol{\mu}, \boldsymbol{\Sigma})$

4. If $\mathbf{A}$ is a $q \times p$ matrix of constants, $\mathbf{AX} \sim N_q(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top)$.

5. Linear combinations of multivariate normals are multivariate normal.

6. All the marginals (dimension less than $p$) of $\mathbf{X}$ are (multivariate) normal, but it is possible in theory to have a collection of univariate normals whose joint distribution is not multivariate normal.

7. For the multivariate normal, zero covariance implies independence. The multivariate normal is the only continuous distribution with this property.

8. The random variable $(\mathbf{X} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{X} - \boldsymbol{\mu})$ has a chi-squared distribution with $p$ degrees of freedom.

9. After a bit of work, the multivariate normal likelihood may be written as

$$L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-np/2} \exp -\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) + (\overline{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu})\right\}, \quad \text{(A.20)}$$

where $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^\top$ is the sample variance-covariance matrix (it would be unbiased if divided by $n - 1$).

Here's how Expression (A.20) above for $L(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is obtained.

$$\begin{aligned} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \prod_{i=1}^n \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}(2\pi)^{\frac{p}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right\} \\ &= |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-np/2} \exp\left\{-\frac{1}{2}\sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right\} \end{aligned}$$

Adding and subtracting $\overline{\mathbf{x}}$ in $\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})$, we get

$$\begin{aligned}
\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}) &= \sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}}+\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\overline{\mathbf{x}}+\overline{\mathbf{x}}-\boldsymbol{\mu}) \\
&= \sum_{i=1}^{n}(\mathbf{a}_i+\mathbf{b})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{a}_i+\mathbf{b}) \\
&= \sum_{i=1}^{n}\left(\mathbf{a}_i^\top\boldsymbol{\Sigma}^{-1}\mathbf{a}_i+\mathbf{a}_i^\top\boldsymbol{\Sigma}^{-1}\mathbf{b}+\mathbf{b}^\top\boldsymbol{\Sigma}^{-1}\mathbf{a}_i+\mathbf{b}^\top\boldsymbol{\Sigma}^{-1}\mathbf{b}\right) \\
&= \left(\sum_{i=1}^{n}\mathbf{a}_i^\top\boldsymbol{\Sigma}^{-1}\mathbf{a}_i\right)+\mathbf{0}+\mathbf{0}+n\,\mathbf{b}^\top\boldsymbol{\Sigma}^{-1}\mathbf{b} \\
&= \sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\overline{\mathbf{x}}) \;+\; n\,(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu})
\end{aligned}$$

Now, because $\sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\overline{\mathbf{x}})$ is a $1\times 1$ matrix, it equals its own trace and we can use $tr(\mathbf{AB})=tr(\mathbf{BA})$.

$$\begin{aligned}
\sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\overline{\mathbf{x}}) &= tr\left\{\sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\overline{\mathbf{x}})\right\} \\
&= \sum_{i=1}^{n}tr\left\{(\mathbf{x}_i-\overline{\mathbf{x}})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\overline{\mathbf{x}})\right\} \\
&= \sum_{i=1}^{n}tr\left\{\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\overline{\mathbf{x}})(\mathbf{x}_i-\overline{\mathbf{x}})^\top\right\} \\
&= tr\left\{\sum_{i=1}^{n}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\overline{\mathbf{x}})(\mathbf{x}_i-\overline{\mathbf{x}})^\top\right\} \\
&= tr\left\{\boldsymbol{\Sigma}^{-1}\sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})(\mathbf{x}_i-\overline{\mathbf{x}})^\top\right\} \\
&= n\,tr\left\{\boldsymbol{\Sigma}^{-1}\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})(\mathbf{x}_i-\overline{\mathbf{x}})^\top\right\} \\
&= n\,tr\left(\boldsymbol{\Sigma}^{-1}\widehat{\boldsymbol{\Sigma}}\right),
\end{aligned}$$

where $\widehat{\boldsymbol{\Sigma}}=\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i-\overline{\mathbf{x}})(\mathbf{x}_i-\overline{\mathbf{x}})^\top$ is the sample variance-covariance matrix. Substituting for $\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})$,

$$\begin{aligned}
L(\boldsymbol{\mu},\boldsymbol{\Sigma}) &= |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-np/2}\exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})\right\} \\
&= |\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-np/2}\exp-\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})+(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu})\right\}.
\end{aligned}$$

Notice how the multivariate normal likelihood depends on the sample data only through the sufficient statistic $(\overline{\mathbf{X}}, \widehat{\mathbf{\Sigma}})$.

**Exercises A.4**

A.4.1) Let $X_1$ be Normal$(\mu_1, \sigma_1^2)$, and $X_2$ be Normal$(\mu_2, \sigma_2^2)$, independent of $X_1$. What is the joint distribution of $Y_1 = X_1 + X_2$ and $Y_2 = X_1 - X_2$? What is required for $Y_1$ and $Y_2$ to be independent?

A.4.2) Let $\mathbf{X} = (X_1, X_2, X_3)^\top$ be multivariate normal with

$$\boldsymbol{\mu} = \begin{pmatrix} 1 \\ 0 \\ 6 \end{pmatrix} \text{ and } \mathbf{\Sigma} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Let $Y_1 = X_1 + X_2$ and $Y_2 = X_2 + X_3$. Find the joint distribution of $Y_1$ and $Y_2$.

A.4.3) Let $X_1$ be Normal$(\mu_1, \sigma_1^2)$, and $X_2$ be Normal$(\mu_2, \sigma_2^2)$, independent of $X_1$. What is the joint distribution of $Y_1 = X_1 + X_2$ and $Y_2 = X_1 - X_2$? What is required for $Y_1$ and $Y_2$ to be independent? Hint: Use matrices.

A.4.4) Let $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{X}$ is an $n \times p$ matrix of known constants, $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown constants, and $\boldsymbol{\epsilon}$ is multivariate normal with mean zero and covariance matrix $\sigma^2 \mathbf{I}_n$, where $\sigma^2 > 0$ is a constant. In the following, it may be helpful to recall that $(\mathbf{A}^{-1})^\top = (\mathbf{A}^\top)^{-1}$.

(a) What is the distribution of $\mathbf{Y}$?

(b) The maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$. What is the distribution of $\hat{\boldsymbol{\beta}}$? Show the calculations.

(c) Let $\widehat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$. What is the distribution of $\widehat{\mathbf{Y}}$? Show the calculations.

(d) Let the vector of residuals $\mathbf{e} = (\mathbf{Y} - \widehat{\mathbf{Y}})$. What is the distribution of $\mathbf{e}$? Show the calculations. Simplify both the expected value (which is zero) and the covariance matrix.

A.4.5) Show that if $\mathbf{X} \sim N(\boldsymbol{\mu}, \mathbf{\Sigma})$, $Y = (\mathbf{X} - \boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu})$ has a chi-square distribution with $p$ degrees of freedom.

A.4.6) Write down a scalar version of formula (A.20) for the multivariate normal likelihood, showing that you understand the notation. Then derive your formula from the univariate normal likelihood.

A.4.7) Prove the formula (A.20) for the multivariate normal likelihood. Show all the calculations.

A.4.8) Prove that for *any* positive definite $\mathbf{\Sigma}$, the likelihood (A.20) is maximized when $\overline{\mathbf{x}} = \boldsymbol{\mu}$. How do you know this maximum must be unique? Cite the necessary matrix facts from Section A.2 of this Appendix.

## A.5 A Bit of Large Sample Theory

For this part, it helps to start by going down to the basement and taking a look at the foundations of the building. There is an underlying sample space $\Omega$, consisting of sample points $\omega \in \Omega$[5]. The specific nature of a point $\omega$ in applications depends on what is being observed. For example, if we were observing whether a single individual is male or female, $\Omega$ might be $\{F, M\}$. If we selected a pair of individuals and observed their genders in order, $\Omega$ might be $\{(F,F),(F,M),(M,F),(M,M)\}$. If we selected $n$ individuals and just *counted* the number of females, $\Omega$ might be $\{0,\ldots,n\}$. For limits problems, the points in $\Omega$ are infinite sequences.

Let $\mathcal{A}$ be a class of subsets of $\Omega$ (that is, a set of *events*), and let $\mathcal{P}$ be a probability function that assigns numbers between zero and one inclusive to the elements of $\mathcal{A}$. A *random variable* $X = X(\omega)$ is a function that maps $\Omega$ into some other space, typically $\mathbb{R}$ or $\mathbb{R}^k$. Think of taking a measurement: if $\Omega$ is a set of students, $X(\omega)$ might be the cumulative grade point average of student $\omega$.

Suppose the random variable $X$ maps $\Omega$ into the set of real numbers $\mathbb{R}$. Then $X$ induces a probability measure on a class[6] $\mathcal{B}$ of subsets of $\mathbb{R}$, by means of

$$Pr\{X \in B\} = \mathcal{P}(\{\omega \in \Omega : X(\omega) \in B\})$$

for $B \in \mathcal{B}$.

Suppose we have a sample of data $X_1(\omega),\ldots,X_n(\omega)$, and we calculate a function of the sample data $T = T(X_1,\ldots,X_n)$. For example $T$ could be a *statistic* like the sample mean $\overline{X}$. It is helpful to write $T = T_n(\omega)$, to indicate that $T$ is a random variable (a function from $\Omega$ into $\mathbb{R}$) that depend upon the sample size $n$.

Frequently it is useful to let $n \to \infty$, because when the sequence $T_1, T_2, \ldots$ converges, it is an indication of what happens when the sample is large enough. But this is not just a sequence of numbers; it is a sequence of functions. Several different types of convergence are meaningful.

### A.5.1 Modes of Convergence

Throughout, let $T_1, T_2, \ldots$ be a sequence of random variables, and let $T$ be another random variable. It is quite possible and often useful for $T = T(\omega)$ to be a constant — that is, a constant function of $\omega$. In that case $T$ is a "degenerate" random variable, with $P\{T = c\} = 1$ for some constant $c$.

#### Almost Sure Convergence

We say that $T_n$ converges *almost surely* to $T$, and write $T_n \overset{a.s.}{\to}$ if

$$\mathcal{P}\{\omega : \lim_{n\to\infty} T_n(\omega) = T(\omega)\} = 1.$$

[5]Throughout most of this book, $\Omega$ is a covariance matrix. The symbol will briefly have its usual meaning here, just for the discussion of almost sure convergence

[6]I'm thinking of the Borel $\sigma$-algebra, but there is no need to go that far.

That is, except possibly for $\omega \in A$ with $\mathcal{P}(A) = 0$, $T_n(\omega)$ converges to the random variable $T(\omega)$ like an ordinary limit, and all the usual rules apply — for example, the limit of a continuous function is the continuous function of the limit, L'Hôpital's rule and so on. Almost sure convergence is also called *convergence with probability one*, or sometimes *strong convergence*.

Almost sure convergence may be the most technically "advanced" mode of convergence, but it is also perhaps the easiest to work with, because you treat the sequence $T_1, T_2, \ldots$ like numbers, find the limit, and then mention that the result applies "except possibly on a set of probability zero."

The main entry point to establishing almost sure convergence is the *Strong Law of Large Numbers,* which involves almost sure convergence to a constant. Let $X_1, \ldots X_n$ be independent and identically distributed random variables with expected value $\mu$. Denote the sample mean as usual by $\overline{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$. The Strong Law of Large Numbers (SLLN) says

$$\overline{X}_n \stackrel{a.s.}{\to} \mu. \tag{A.21}$$

The only condition required for this to hold is the existence of the expected value.

Let $X_1, \ldots X_n$ be independent and identically distributed random variables; let $X$ be a general random variable from this same distribution, and $Y = g(X)$. The change of variables formula (A.1) can be combined with the Strong Law of Large Numbers to write

$$\frac{1}{n}\sum_{i=1}^n g(X_i) = \frac{1}{n}\sum_{i=1}^n Y_i \stackrel{a.s.}{\to} E(Y) = E(g(X)). \tag{A.22}$$

This means that sample moments converge almost surely to population moments:

$$\frac{1}{n}\sum_{i=1}^n X_i^k \stackrel{a.s.}{\to} E(X^k)$$

It even yields rules like

$$\frac{1}{n}\sum_{i=1}^n U_i^2 V_i W_i^3 \stackrel{a.s.}{\to} E(U^2 V W^3).$$

### Convergence in Probability

We say that $T_n$ converges *in probability* to $T$, and write $T_n \stackrel{P}{\to} T$ if for all $\epsilon > 0$,

$$\lim_{n\to\infty} P\{|T_n - T| < \epsilon\} = 1.$$

Convergence in probability is implied by almost sure convergence, so corresponding to the Strong Law of Large Numbers is the Weak Law of Large Numbers (WLLN). Let $X_1, \ldots X_n$ be independent and identically distributed random variables with expected value $\mu$. Then the sample mean converges in probability to $\mu$:

$$\overline{X}_n \stackrel{P}{\to} \mu. \tag{A.23}$$

A change of variables rule like expression (A.22) holds, and sample moments converge in probability to population moments. These rules follow from the corresponding facts about almost sure convergence.

Another way of establishing convergence in probability to a constant without using the definition is the *Variance Rule*. Let $\theta$ be a constant. Then if $\lim_{n\to\infty} E(T_n) = \theta$ and $\lim_{n\to\infty} Var(T_n) = 0$, it follows that $T_n \xrightarrow{P} \theta$. But convergence in probability does not imply the conditions of the Variance Rule.

### Convergence in Distribution

Denote the cumulative distribution functions of $T_1, T_2, \ldots$ by $F_1(t), F_2(t), \ldots$ respectively, and denote the cumulative distribution function of $T$ by $F(t)$. We say that $T_n$ converges *in distribution* to $T$, and write $T_n \xrightarrow{d} T$ if for every point $t$ at which $F$ is continuous,

$$\lim_{n\to\infty} F_n(t) = F(t).$$

The main entry point to convergence in distribution is the *Central Limit Theorem*. Let $X_1, \ldots X_n$ be independent and identically distributed random variables with mean $\mu$ and variance $\sigma^2$. Then

$$Z_n = \frac{\sqrt{n}(\overline{X}_n - \mu)}{\sigma} \xrightarrow{d} Z \sim N(0,1).$$

In applications, the sample standard deviation may be substituted for $\sigma$, and the result still holds.

A useful tool is provided by the univariate *delta method*[7]. Let $\sqrt{n}(X_n - \theta) \xrightarrow{d} X$, and let $g(x)$ be a function with $g'(\theta) \neq 0$ and $g''(x)$ continuous at $x = \theta$. Then

$$\sqrt{n}(g(X_n) - g(\theta)) \xrightarrow{d} g'(\theta)X.$$

In particular, $\sqrt{n}(g(\overline{X}_n) - g(\mu)) \xrightarrow{d} Y \sim N(0, g'(\mu)^2\sigma^2)$.

### Connections among the Modes of Convergence

- $T_n \xrightarrow{a.s.} T \Rightarrow T_n \xrightarrow{P} T \Rightarrow T_n \xrightarrow{d} T$.
- If $a$ is a constant, $T_n \xrightarrow{d} a \Rightarrow T_n \xrightarrow{P} a$.

Sometimes we say the distribution of the sample mean is approximately normal, or asymptotically normal. This is justified by the Central Limit Theorem, but it does *not* mean that $\overline{X}_n$ converges in distribution to a normal random variable. The Law of Large Numbers says that $\overline{X}_n$ converges almost surely (and in probability) to a constant, $\mu$. This

[7]The delta method is named after the way it is proved; it uses Taylor's theorem, and the "delta" part is connected to the definition of a derivative. We will just use it.

means $\overline{X}_n$ converges to $\mu$ in distribution as well. So why would we say that for large $n$, the sample mean is approximately $N(\mu, \frac{\sigma^2}{n})$?

What we have is $Z_n = \frac{\sqrt{n}(\overline{X}_n - \mu)}{\sigma} \xrightarrow{d} Z \sim N(0,1)$. So,

$$\begin{aligned} Pr\{\overline{X}_n \leq x\} &= Pr\left\{\frac{\sqrt{n}(\overline{X}_n - \mu)}{\sigma} \leq \frac{\sqrt{n}(x-\mu)}{\sigma}\right\} \\ &= Pr\left\{Z_n \leq \frac{\sqrt{n}(x-\mu)}{\sigma}\right\} \approx \Phi\left(\frac{\sqrt{n}(x-\mu)}{\sigma}\right), \end{aligned}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal.

Now suppose that $Y$ is *exactly* $N(\mu, \frac{\sigma^2}{n})$. Then,

$$\begin{aligned} Pr\{Y \leq x\} &= Pr\left\{\frac{\sqrt{n}(Y - \mu)}{\sigma} \leq \frac{\sqrt{n}(x-\mu)}{\sigma}\right\} \\ &= Pr\left\{Z \leq \frac{\sqrt{n}(x-\mu)}{\sigma}\right\} = \Phi\left(\frac{\sqrt{n}(x-\mu)}{\sigma}\right). \end{aligned}$$

So we see that the Central Limit Theorem tells us to calculate probabilities for $\overline{X}_n$ just as we would if $\overline{X}_n$ had a distribution that was exactly normal with expected value $\mu$ and variance $\frac{\sigma^2}{n}$. This the justification for saying that the sample mean is "asymptotically normal," and writing $\overline{X}_n \dot{\sim} N(\mu, \frac{\sigma^2}{n})$. Here are three additional remarks.

- Quantities like $\frac{1}{n}\sum_{i=1}^n X_i^2$ and $\frac{1}{n}\sum_{i=1}^n X_i Y_i$ and so on are asymptotically normal too, because they are just sample means.
- The delta method says that smooth functions of the sample mean are asymptotically normal.
- All this generalizes nicely to the multivariate case.

### A.5.2 Consistency

For this application, $T_1, T_2, \ldots$ are not just random variables: They are *statistics*[8] that estimate some parameter $\theta$. The statistic $T_n$ is said to be *consistent* for $\theta$ if $T_n \xrightarrow{P} \theta$ for all $\theta \in \Theta$.

Let us take a closer look at this important concept. Using the definition of convergence in probability, saying that $T_n$ is consistent for $\theta$ means that for any tiny positive constant $\epsilon$, no matter *how* tiny,

$$\lim_{n\to\infty} P\{|T_n - \theta| < \epsilon\} = 1.$$

So, take an arbitrarily small interval around the true parameter value. For any given sample size $n$, a certain amount of the probability distribution of $T_n$ falls between $\theta - \epsilon$

[8] A statistic is a function of the sample data that does not depend functionally upon any unknown parameter. That is, symbol for the parameter does not appear in the formula for the statistic.

and $\theta+\epsilon$. Consistency means that in the limit, *all* the probability falls in this interval, no matter how small the interval is. Basically, consistency is saying that for a large enough sample size, the statistic (estimator) will probably be close to parameter it is estimating — regardless of how strict your definitions of "probably" and "close" might be.

Even better than ordinary consistency is *strong consistency*, which means $T_n \stackrel{a.s.}{\to} \theta$. Instead of saying $T_n$ will probably be close to $\theta$, strong consistency says that for a large enough sample size, the probability that it *will* be close equals one. Because almost sure convergence implies convergence in probability, strong consistency implies ordinary consistency.

One last remark is that while consistency is an important property in an estimator, in a way it is the least we should expect. Consistency means that with an infinite amount of data, we would know the truth. If this is *not* the case, something is seriously wrong[9].

**Exercises A.5.2**

A.5.1) Let $X_1, \ldots, X_n$ be a random sample from a continuous distribution with density

$$f(x;\theta) = \frac{1}{\theta^{1/2}\sqrt{2\pi}}\, e^{-\frac{x^2}{2\theta}},$$

where the parameter $\theta > 0$. Propose a reasonable estimator for the parameter $\theta$, and use the Law of Large Numbers to show that your estimator is consistent.

A.5.2) Let $X_1, \ldots, X_n$ be a random sample from a Binomial distribution with parameters 3 and $\theta$. That is,

$$P(X_i = x_i) = \binom{3}{x_i} \theta^{x_i}(1-\theta)^{3-x_i},$$

for $x_i = 0, 1, 2, 3$. Find a reasonable estimator of $\theta$, and prove that it is strongly consistent. Where you get your estimator does not really matter, but please state how you thought of it.

A.5.3) Let $X_1, \ldots, X_n$ be a random sample from a continuous distribution with density

$$f(x;\tau) = \frac{\tau^{1/2}}{\sqrt{2\pi}}\, e^{-\frac{\tau x^2}{2}},$$

where the parameter $\tau > 0$. Let

$$\widehat{\tau} = \frac{n}{\sum_{i=1}^{n} X_i^2}.$$

Is $\widehat{\tau}$ consistent for $\tau$? Answer Yes or No and prove your answer. Hint: You can just write down $E(X^2)$ by inspection. This is a very familiar distribution; have confidence!

[9] In structural equation models, a parameter that is not identifiable cannot be estimated consistently. This is why model identification is such an important topic.

A.5.4) Independently for $i = 1, \ldots, n$, let

$$Y_i = \beta X_i + \epsilon_i,$$

where $E(X_i) = E(\epsilon_i) = 0$, $Var(X_i) = \sigma_x^2$, $Var(\epsilon_i) = \sigma_\epsilon^2$, and $\epsilon_i$ is independent of $X_i$. Let

$$\widehat{\beta} = \frac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n X_i^2}.$$

Is $\widehat{\beta}$ consistent for $\beta$? Answer Yes or No and prove your answer.

A.5.5) Another Method of Moments estimator for Problem A.5.2 is $\widehat{\beta}_2 = \frac{\overline{Y}_n}{\overline{X}_n}$.

(a) Show that $\widehat{\beta}_2 \xrightarrow{p} \beta$ in most of the parameter space.

(b) However, consistency means that the estimator converges to the parameter in probability *everywhere* in the parameter space. Where does $\widehat{\beta}_2$ fail, and why?

A.5.6) Let $X_1, \ldots, X_n$ be a random sample from a Gamma distribution with $\alpha = \beta = \theta > 0$. That is, the density is

$$f(x; \theta) = \frac{1}{\theta^\theta \Gamma(\theta)} e^{-x/\theta} x^{\theta - 1},$$

for $x > 0$. Let $\widehat{\theta} = \overline{X}_n$. Is $\widehat{\theta}$ consistent for $\theta$? Answer Yes or No and prove your answer.

A.5.7) Let $X_1, \ldots, X_n$ be a random sample from a distribution with expected value $\mu$ and variance $\sigma_x^2$. Independently of $X_1, \ldots, X_n$, let $Y_1, \ldots, Y_n$ be a random sample from a distribution with the same expected value $\mu$ and variance $\sigma_y^2$. Let Let $T_n = \alpha \overline{X}_n + (1 - \alpha)\overline{Y}_n$, where $0 \leq \alpha \leq 1$. Is $T_n$ always a consistent estimator of $\mu$? Answer Yes or No and show your work.

A.5.8) Let $X_1, \ldots, X_n$ be a random sample from a distribution with mean $\mu$. Show that $T_n = \frac{1}{n+400} \sum_{i=1}^n X_i$ is consistent for $\mu$.

A.5.9) Let $X_1, \ldots, X_n$ be a random sample from a distribution with mean $\mu$ and variance $\sigma^2$. Prove that the sample variance $S^2 = \frac{\sum_{i=1}^n (X_i - \overline{X})^2}{n-1}$ is consistent for $\sigma^2$.

A.5.10) Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be a random sample from a bivariate distribution with $E(X_i) = \mu_x$, $E(Y_i) = \mu_y$, $Var(X_i) = \sigma_x^2$, $Var(Y_i) = \sigma_y^2$, and $Cov(X_i, Y_i) = \sigma_{xy}$. Show that the sample covariance $S_{xy} = \frac{\sum_{i=1}^n (X_i - \overline{X})(Y_i - \overline{Y})}{n-1}$ is a consistent estimator of $\sigma_{xy}$.

A.5.11) Let $X_1, \ldots, X_n$ be a random sample from a Poisson distribution with parameter $\lambda$. You know that $E(X_i) = Var(X_i) = \lambda$; there is no need to prove it.

From the Strong Law of Large Numbers, it follows immediately that $\overline{X}_n$ is strongly consistent for $\lambda$. Let

$$\widehat{\lambda} = \frac{\sum_{i=1}^n (X_i - \overline{X}_n)^2}{n - 4}.$$

Is $\widehat{\lambda}$ also consistent for $\lambda$? Answer Yes or No and prove your answer.

### A.5.3 Convergence of random vectors

Almost all applied problems are multi-parameter, and that certainly applies to the ones in this book. Parameter estimates are usually random vectors. It is very convenient that in terms of convergence, the multivariate case is very similar to the univariate case just discussed. The is based on material in Thomas Ferguson's beautiful little book *A course in large sample theory*, which is highly recommended. All quantities in boldface are vectors in $\mathbb{R}^m$ unless otherwise indicated.

1. Definitions
   - $\star$ $\mathbf{T}_n \xrightarrow{a.s.} \mathbf{T}$ means $P\{\omega : \lim_{n\to\infty} \mathbf{T}_n(\omega) = \mathbf{T}(\omega)\} = 1$.
   - $\star$ $\mathbf{T}_n \xrightarrow{P} \mathbf{T}$ means $\forall \epsilon > 0$, $\lim_{n\to\infty} P\{||\mathbf{T}_n - \mathbf{T}|| < \epsilon\} = 1$.
   - $\star$ $\mathbf{T}_n \xrightarrow{d} \mathbf{T}$ means for every continuity point $\mathbf{t}$ of $F_{\mathbf{T}}$, $\lim_{n\to\infty} F_{\mathbf{T}_n}(\mathbf{t}) = F_{\mathbf{T}}(\mathbf{t})$.
2. $\mathbf{T}_n \xrightarrow{a.s.} \mathbf{T} \Rightarrow \mathbf{T}_n \xrightarrow{P} \mathbf{T} \Rightarrow \mathbf{T}_n \xrightarrow{d} \mathbf{T}$.
3. If $\mathbf{a}$ is a vector of constants, $\mathbf{T}_n \xrightarrow{d} \mathbf{a} \Rightarrow \mathbf{T}_n \xrightarrow{P} \mathbf{a}$.
4. Strong Law of Large Numbers: Let $\mathbf{X}_1, \ldots \mathbf{X}_n$ be independent and identically distributed random vectors with finite first moment, and let $\mathbf{X}$ be a general random vector from the same distribution. Then $\overline{\mathbf{X}}_n \xrightarrow{a.s.} E(\mathbf{X})$.
5. Central Limit Theorem: Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be i.i.d. random vectors with expected value vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Then $\sqrt{n}(\overline{\mathbf{X}}_n - \boldsymbol{\mu})$ converges in distribution to a multivariate normal with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$.
6. Slutsky Theorems for Convergence in Distribution:
   (a) If $\mathbf{T}_n \in \mathbb{R}^m$, $\mathbf{T}_n \xrightarrow{d} \mathbf{T}$ and if $f : \mathbb{R}^m \to \mathbb{R}^q$ (where $q \leq m$) is continuous except possibly on a set $C$ with $P(\mathbf{T} \in C) = 0$, then $f(\mathbf{T}_n) \xrightarrow{d} f(\mathbf{T})$.
   (b) If $\mathbf{T}_n \xrightarrow{d} \mathbf{T}$ and $(\mathbf{T}_n - \mathbf{Y}_n) \xrightarrow{P} 0$, then $\mathbf{Y}_n \xrightarrow{d} \mathbf{T}$.
   (c) If $\mathbf{T}_n \in \mathbb{R}^d$, $\mathbf{Y}_n \in \mathbb{R}^k$, $\mathbf{T}_n \xrightarrow{d} \mathbf{T}$ and $\mathbf{Y}_n \xrightarrow{P} \mathbf{c}$, then
   $$\begin{pmatrix} \mathbf{T}_n \\ \mathbf{Y}_n \end{pmatrix} \xrightarrow{d} \begin{pmatrix} \mathbf{T} \\ \mathbf{c} \end{pmatrix}$$
7. Slutsky Theorems for Convergence in Probability:
   (a) If $\mathbf{T}_n \in \mathbb{R}^m$, $\mathbf{T}_n \xrightarrow{P} \mathbf{T}$ and if $f : \mathbb{R}^m \to \mathbb{R}^q$ (where $q \leq m$) is continuous except possibly on a set $C$ with $P(\mathbf{T} \in C) = 0$, then $f(\mathbf{T}_n) \xrightarrow{P} f(\mathbf{T})$.

(b) If $\mathbf{T}_n \xrightarrow{P} \mathbf{T}$ and $(\mathbf{T}_n - \mathbf{Y}_n) \xrightarrow{P} 0$, then $\mathbf{Y}_n \xrightarrow{P} \mathbf{T}$.

(c) If $\mathbf{T}_n \in \mathbb{R}^d$, $\mathbf{Y}_n \in \mathbb{R}^k$, $\mathbf{T}_n \xrightarrow{P} \mathbf{T}$ and $\mathbf{Y}_n \xrightarrow{P} \mathbf{Y}$, then

$$\begin{pmatrix} \mathbf{T}_n \\ \mathbf{Y}_n \end{pmatrix} \xrightarrow{P} \begin{pmatrix} \mathbf{T} \\ \mathbf{Y} \end{pmatrix}$$

8. Delta Method (Theorem of Cramér, Ferguson p. 45): Let $g : \mathbb{R}^d \to \mathbb{R}^k$ be such that the elements of $\dot{\mathrm{g}}(\mathbf{x}) = \left[\frac{\partial g_i}{\partial x_j}\right]_{k \times d}$ are continuous in a neighborhood of $\boldsymbol{\theta} \in \mathbb{R}^d$. If $\mathbf{T}_n$ is a sequence of $d$-dimensional random vectors such that $\sqrt{n}(\mathbf{T}_n - \boldsymbol{\theta}) \xrightarrow{d} \mathbf{T}$, then $\sqrt{n}(g(\mathbf{T}_n) - g(\boldsymbol{\theta})) \xrightarrow{d} \dot{\mathrm{g}}(\boldsymbol{\theta})\mathbf{T}$. In particular, if $\sqrt{n}(\mathbf{T}_n - \boldsymbol{\theta}) \xrightarrow{d} \mathbf{T} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, then $\sqrt{n}(g(\mathbf{T}_n) - g(\boldsymbol{\theta})) \xrightarrow{d} \mathbf{Y} \sim N(\mathbf{0}, \dot{\mathrm{g}}(\boldsymbol{\theta})\boldsymbol{\Sigma}\dot{\mathrm{g}}(\boldsymbol{\theta})^\top)$.

In the multivariate delta method, the matrix $\dot{\mathrm{g}}(\boldsymbol{\theta})$ is the Jacobian of the transformation $g$. The idea is that smooth functions of asymptotically normal random variables are also asymptotically normal.

**Asymptotic normality of variances and covariances**

The following theorem says that even for non-normal data, the unique elements of the sample variance-covariance matrix have a joint distribution that is approximately multivariate normal for large samples. The means are the corresponding elements of the true variance-covariance matrix, and the asymptotic variance-covariance matrix (of the variances and covariances) is $\mathbf{L}/n$, where $\mathbf{L}$ is given below. The proof is a good workout in the Slutsky lemmas.

**Theorem A.1** *Let $\mathbf{d}_1, \ldots, \mathbf{d}_n$ be a random sample from a $k$-dimensional distribution with expected value $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$, and finite fourth moments. Define $\mathbf{w} = vech\{(\mathbf{d}_1 - \boldsymbol{\mu})(\mathbf{d}_1 - \boldsymbol{\mu})^\top\}$ and let $\mathbf{L} = cov(\mathbf{w})$. Then*

$$\sqrt{n}\left(vech(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma})\right) \xrightarrow{d} \mathbf{t} \sim N(\mathbf{0}, \mathbf{L}).$$

**Proof**

$$\begin{aligned}
\widehat{\boldsymbol{\Sigma}} &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i-\overline{\mathbf{d}}_n)(\mathbf{d}_i-\overline{\mathbf{d}}_n)^\top \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i-\boldsymbol{\mu}+\boldsymbol{\mu}-\overline{\mathbf{d}}_n)(\mathbf{d}_i-\boldsymbol{\mu}+\boldsymbol{\mu}-\overline{\mathbf{d}}_n)^\top \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i-\boldsymbol{\mu})(\mathbf{d}_i-\boldsymbol{\mu})^\top \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i-\boldsymbol{\mu})(\boldsymbol{\mu}-\overline{\mathbf{d}}_n)^\top + \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\mu}-\overline{\mathbf{d}}_n)(\mathbf{d}_i-\boldsymbol{\mu})^\top \\
&\quad + (\boldsymbol{\mu}-\overline{\mathbf{d}}_n)(\boldsymbol{\mu}-\overline{\mathbf{d}}_n)^\top \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i-\boldsymbol{\mu})(\mathbf{d}_i-\boldsymbol{\mu})^\top \\
&\quad + (\overline{\mathbf{d}}_n-\boldsymbol{\mu})(\boldsymbol{\mu}-\overline{\mathbf{d}}_n)^\top + (\boldsymbol{\mu}-\overline{\mathbf{d}}_n)(\overline{\mathbf{d}}_n-\boldsymbol{\mu})^\top + (\overline{\mathbf{d}}_n-\boldsymbol{\mu})(\overline{\mathbf{d}}_n-\boldsymbol{\mu})^\top \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i-\boldsymbol{\mu})(\mathbf{d}_i-\boldsymbol{\mu})^\top - (\overline{\mathbf{d}}_n-\boldsymbol{\mu})(\overline{\mathbf{d}}_n-\boldsymbol{\mu})^\top.
\end{aligned}$$

So,

$$\sqrt{n}(\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}) = \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i-\boldsymbol{\mu})(\mathbf{d}_i-\boldsymbol{\mu})^\top\right) - \sqrt{n}(\overline{\mathbf{d}}_n-\boldsymbol{\mu})(\overline{\mathbf{d}}_n-\boldsymbol{\mu})^\top.$$

The second term goes to zero in probability, because the Central Limit Theorem (item 5 in the list of large-sample results) says that $\sqrt{n}(\boldsymbol{\mu}-\overline{\mathbf{d}}_n) \xrightarrow{d} \mathbf{Y} \sim N(\mathbf{0},\boldsymbol{\Sigma})$, while the Law of Large Numbers (item 4) tells us $\overline{\mathbf{d}}_n-\boldsymbol{\mu} \xrightarrow{P} \mathbf{0}$. Then Slutsky Lemma 6c implies

$$\begin{pmatrix} \sqrt{n}(\boldsymbol{\mu}-\overline{\mathbf{d}}_n) \\ \\ \overline{\mathbf{d}}_n-\boldsymbol{\mu} \end{pmatrix} \xrightarrow{d} \begin{pmatrix} \mathbf{Y} \\ \\ \mathbf{0} \end{pmatrix},$$

and Slutsky Lemma 6a (continuous mapping) establishes $\sqrt{n}(\overline{\mathbf{d}}_n-\boldsymbol{\mu})(\overline{\mathbf{d}}_n-\boldsymbol{\mu})^\top \xrightarrow{d} \mathbf{Y}\mathbf{0}^\top = \mathbf{0} \Rightarrow \sqrt{n}(\overline{\mathbf{d}}_n-\boldsymbol{\mu})(\overline{\mathbf{d}}_n-\boldsymbol{\mu})^\top \xrightarrow{P} \mathbf{0}$.

Therefore by Slutsky Lemma 6b, $\sqrt{n}(\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma})$ and $\sqrt{n}(\widetilde{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma})$ converge in distribution to the same random matrix, where $\widetilde{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i-\boldsymbol{\mu})(\mathbf{d}_i-\boldsymbol{\mu})^\top$. Now $vech\left(\widetilde{\boldsymbol{\Sigma}}\right)$ is just the mean of $n$ independent and identically distributed random vectors, each with mean $vech\,(\boldsymbol{\Sigma})$ and covariance matrix $\mathbf{L}$ as given by the theorem. The Central Limit Theorem then implies $\sqrt{n}\left(vech(\widetilde{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma})\right) \xrightarrow{d} \mathbf{T} \sim N(\mathbf{0},\mathbf{L})$, and the conclusion follows. ■

**Using the delta method instead** The multivariate delta method (item 8 in the list of large-sample results) can also be used to establish Theorem A.1. The details are a useful

illustration of how to apply the delta method. The calculations will be carried out for a $2 \times 2$ covariance matrix, and the extension to larger problems will be clear.

Independently for $i = 1, \ldots, n$, let

$$\mathbf{d}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \text{ with } E(\mathbf{d}_i) = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix} \text{ and } cov(\mathbf{d}_i) = \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}.$$

The sample variance of $x$ (with $n$ in the denominator, which is more convenient for asymptotics) is

$$\widehat{\sigma}_x^2 = \tfrac{1}{n}\textstyle\sum_{i=1}^n (x_i - \bar{x}_n)^2 = \tfrac{1}{n}\sum_{i=1}^n x_i^2 \; - \; \bar{x}_n^2,$$

and the sample covariance of $x$ and $y$ is

$$\widehat{\sigma}_{xy} = \tfrac{1}{n}\textstyle\sum_{i=1}^n (x_i - \bar{x}_n)(y_i - \bar{y}_n) = \tfrac{1}{n}\sum_{i=1}^n x_i y_i \; - \; \bar{x}_n \bar{y}_n.$$

It's clear that the sample variances and covariances are functions of a collection of sample means. The sample means can be assembled into a vector

$$\overline{\mathbf{T}}_n = \begin{pmatrix} \bar{x}_n \\ \frac{1}{n}\sum_{i=1}^n x_i^2 \\ \bar{y}_n \\ \frac{1}{n}\sum_{i=1}^n y_i^2 \\ \frac{1}{n}\sum_{i=1}^n x_i y_i \end{pmatrix}.$$

To apply the multivariate central limit theorem we need the vectors that are being averaged in order to get $\overline{\mathbf{T}}_n$. That's easy:

$$\mathbf{T}_i = \begin{pmatrix} x_i \\ x_i^2 \\ y_i \\ y_i^2 \\ x_i y_i \end{pmatrix}, \quad \text{with} \quad E(\mathbf{T}_i) = \boldsymbol{\mu} = \begin{pmatrix} E(x) \\ E(x^2) \\ E(y) \\ E(y^2) \\ E(xy) \end{pmatrix} = \begin{pmatrix} \mu_x \\ \sigma_x^2 + \mu_x^2 \\ \mu_y \\ \sigma_y^2 + \mu_y^2 \\ \sigma_{xy} + \mu_x \mu_y \end{pmatrix}.$$

Denoting $cov(\mathbf{T}_i)$ by $\mathbf{W}$, the central limit theorem (item 5 in the list of large-sample results) yields $\sqrt{n}(\overline{\mathbf{X}}_n - \boldsymbol{\mu}) \xrightarrow{d} \mathbf{T} \sim N(\mathbf{0}, \mathbf{W})$.

Using the notation

$$\mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{pmatrix}, \quad \text{let} \quad g(\mathbf{t}) = \begin{pmatrix} g_1(\mathbf{t}) \\ g_2(\mathbf{t}) \\ g_3(\mathbf{t}) \end{pmatrix} = \begin{pmatrix} t_2 - t_1^2 \\ t_5 - t_1 t_3 \\ t_4 - t_3^2 \end{pmatrix}.$$

This yields

$$g(\overline{\mathbf{T}}_n) = \begin{pmatrix} \widehat{\sigma}_x^2 \\ \widehat{\sigma}_{xy} \\ \widehat{\sigma}_y^2 \end{pmatrix} \quad \text{and} \quad g(\boldsymbol{\mu}) = \begin{pmatrix} \sigma_x^2 \\ \sigma_{xy} \\ \sigma_y^2 \end{pmatrix}.$$

In other words, $g(\overline{\mathbf{T}}_n) = vech(\widehat{\boldsymbol{\Sigma}}_n)$ and $g(\boldsymbol{\mu}) = vech(\boldsymbol{\Sigma})$. By the delta method,

$$\sqrt{n}\left(g(\overline{\mathbf{T}}_n) - g(\boldsymbol{\mu})\right) \xrightarrow{d} \mathbf{T} \sim N\left(\mathbf{0}, \dot{\mathrm{g}}(\boldsymbol{\mu})\mathbf{W}\dot{\mathrm{g}}(\boldsymbol{\mu})^\top\right)$$

That is, $vech(\widehat{\boldsymbol{\Sigma}}_n)$ is asymptotically multivariate normal, with asymptotic mean $vech(\boldsymbol{\Sigma})$, and asymptotic covariance matrix $\frac{1}{n}\dot{\mathrm{g}}(\boldsymbol{\mu})\mathbf{W}\dot{\mathrm{g}}(\boldsymbol{\mu})^\top$. It is worth the effort to calculate the asymptotic covariance matrix for this two-variable case.

Using elementary formulas for variance and covariance together with a slightly extended version of the change of variables formula (A.1), the matrix $\mathbf{W} = cov(\mathbf{T}_i)$ may be written (in upper triangular form and without parentheses on the expected values to fit more material on the page) as

$$\mathbf{W} = cov\begin{pmatrix} x_i \\ x_i^2 \\ y_i \\ y_i^2 \\ x_i y_i \end{pmatrix} = \begin{pmatrix} Ex^2 - (Ex)^2 & Ex^3 - ExEx^2 & Exy - ExEy & Exy^2 - ExEy^2 & Ex^2y - ExExy \\ & Ex^4 - (Ex^2)^2 & Ex^2y - Ex^2Ey & Ex^2y^2 - Ex^2Ey^2 & Ex^3y - Ex^2Exy \\ & & Ey^2 - (Ey)^2 & Ey^3 - EyEy^2 & Exy^2 - EyExy \\ & & & Ey^4 - (Ey^2)^2 & Exy^3 - Ey^2Exy \\ & & & & Ex^2y^2 - (Exy)^2 \end{pmatrix}$$

Recall that

$$g(\mathbf{t}) = \begin{pmatrix} g_1(\mathbf{t}) \\ g_2(\mathbf{t}) \\ g_3(\mathbf{t}) \end{pmatrix} = \begin{pmatrix} t_2 - t_1^2 \\ t_5 - t_1 t_3 \\ t_4 - t_3^2 \end{pmatrix}.$$

The Jacobian evaluated at a general point $\mathbf{t}$ is $[\frac{\partial g_i}{\partial t_j}]$. In this case,

$$\begin{aligned} \dot{\mathrm{g}}(\mathbf{t}) &= \begin{pmatrix} \frac{\partial g_1}{\partial t_1} & \frac{\partial g_1}{\partial t_2} & \frac{\partial g_1}{\partial t_3} & \frac{\partial g_1}{\partial t_4} & \frac{\partial g_1}{\partial t_5} \\ \frac{\partial g_2}{\partial t_1} & \frac{\partial g_2}{\partial t_2} & \frac{\partial g_2}{\partial t_3} & \frac{\partial g_2}{\partial t_4} & \frac{\partial g_2}{\partial t_5} \\ \frac{\partial g_3}{\partial t_1} & \frac{\partial g_3}{\partial t_2} & \frac{\partial g_3}{\partial t_3} & \frac{\partial g_3}{\partial t_4} & \frac{\partial g_3}{\partial t_5} \end{pmatrix} \\ &= \begin{pmatrix} -2t_1 & 1 & 0 & 0 & 0 \\ -t_3 & 0 & -t_1 & 0 & 1 \\ 0 & 0 & -2t_3 & 0 & 0 \end{pmatrix}. \end{aligned}$$

The asymptotic covariance matrix of $vech(\widehat{\boldsymbol{\Sigma}}_n)$ is $\frac{1}{n}\dot{\mathrm{g}}(\boldsymbol{\mu})\mathbf{W}\dot{\mathrm{g}}(\boldsymbol{\mu})^\top$. Carrying out the matrix multiplication and substituting[10],

[10] This is a substantial clerical task, with many opportunities for error. I used a combination of Sage (see Appendix B) and manual editing.

$$cov\begin{pmatrix}\widehat{\sigma}_x^2\\ \widehat{\sigma}_{xy}\\ \widehat{\sigma}_y^2\end{pmatrix} \doteq \frac{1}{n}\begin{pmatrix} 3\,\mu_x^4+6\,\mu_x^2\sigma_x^2-\sigma_x^4 & 3\,\mu_x^3\mu_y+3\,\mu_x\mu_y\sigma_x^2+3\,\mu_x^2\sigma_{xy} & 3\,\mu_x^2\mu_y^2+\mu_y^2\sigma_x^2+\mu_x^2\sigma_y^2-\sigma_x^2\sigma_y^2+ \\ -4\,E(x^3)\mu_x+E(x^4) & -\sigma_x^2\sigma_{xy}-3\,E(x^2y)\mu_x-E(x^3)\mu_y & 4\,\mu_x\mu_y\sigma_{xy}-2\,E(xy^2)\mu_x \\ & +E(x^3y) & -2\,E(x^2y)\mu_y+E(x^2y^2) \\ & & \\ & 3\,\mu_x^2\mu_y^2+\mu_y^2\sigma_x^2+\mu_x^2\sigma_y^2+ & 3\,\mu_x\mu_y^3+3\,\mu_x\mu_y\sigma_y^2+3\,\mu_y^2\sigma_{xy}- \\ & 4\,\mu_x\mu_y\sigma_{xy}-2\,E(xy^2)\mu_x- & \sigma_{xy}\sigma_y^2-E(y^3)\mu_x-3\,E(xy^2)\mu_y+ \\ & 2\,E(x^2y)\mu_y-\sigma_{xy}^2+E(x^2y^2) & E(xy^3) \\ & & \\ & & 3\,\mu_y^4+6\,\mu_y^2\sigma_y^2-\sigma_y^4-4\,E(y^3)\mu_y+ \\ & & E(y^4) \end{pmatrix}. \tag{A.24}$$

The extension to larger numbers of variables is clear, though the details are unavoidably messy. The advantage of the delta method over the proof of Theorem A.1 is that you can see where it's going in advance. As soon as the sample variance and covariance are written as a function of sample means, consistency is guaranteed by the law of large numbers and continuous mapping, and asymptotic normality is guaranteed by the delta method. This applies regardless of how many variables there are. The actual calculation of $\dot{g}(\boldsymbol{\mu})\mathbf{W}\dot{g}(\boldsymbol{\mu})^\top$ is necessary only if you need the formulas for another purpose.

## A.6 Estimation and inference

### A.6.1 Statistical Models

A *statistical model* is a set of assertions that partly specify the probability distribution of the observable data. The specification may be direct or indirect. As an example of direct specification, let $X_1, \ldots, X_n$ be a random sample from a normal distribution with expected value $\mu$ and variance $\sigma^2$. As an example of indirect specification, let $Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \epsilon_i$ for $i = 1, \ldots, n$, where

$\beta_0, \ldots, \beta_k$ are unknown constants. $x_{ij}$ are known constants.
$\epsilon_1, \ldots, \epsilon_n$ are independent $N(0, \sigma^2)$ random variables.
$\sigma^2$ is an unknown constant.

Statistical models leave something unknown. Otherwise, they are probability models. The unknown part of the model for the data is called the *parameter*. Usually, parameters are numbers or vectors of numbers – unknown constants. They are usually denoted by $\theta$ or $\boldsymbol{\theta}$ or other Greek letters.

The *parameter space* is the set of values that can be taken on by the parameter, and will be denoted by $\Theta$, with $\theta \in \Theta$. For the normal random sample example, the parameter space is $\Theta = \{(\mu, \sigma^2) : -\infty < \mu < \infty, \sigma^2 > 0\}$. For the regression example given above, $\Theta = \{(\beta_0, \ldots, \beta_k, \sigma^2) : -\infty < \beta_j < \infty, \sigma^2 > 0\}$.

Parameters need not be numbers. For example, let $X_1, \ldots, X_n$ be a random sample from a continuous distribution with unknown distribution function $F(x)$. The parameter is the unknown distribution function $F(x)$, and the parameter space is a space of

distribution functions. We may be interested only in a *function* of the parameter, like

$$\mu = \int_{-\infty}^{\infty} x f(x)\, dx$$

The rest of $F(x)$ is just a nuisance parameter.

We will use the following framework for parameter estimation and statistical inference. The data are $D_1, \ldots, D_n$ (the letter $D$ stands for data). The distribution of these independent and identically distributed random variables depends on the parameter $\theta$, which is an element of the parameter space $\Theta$. That is,

$$D_1, \ldots, D_n \overset{i.i.d.}{\sim} P_\theta,\ \theta \in \Theta.$$

Both the data values and the parameter may be vectors, even though they are not written in boldface.

To give one more example, the data vector could be $D = \mathbf{X}_1, \ldots \mathbf{X}_n$, a vector of independent multivariate normals of dimension $p$. The parameter space is $\{\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}) : \boldsymbol{\mu} \in \mathbb{R}^p$, and $\boldsymbol{\Sigma}$ is a $p \times p$ symmetric positive definite matrix. $P_\theta$ is the joint distribution function of $\mathbf{X}_1, \ldots \mathbf{X}_n$, with joint density

$$f(\mathbf{x}_1, \ldots \mathbf{x}_n) = \prod_{i=1}^{n} f(\mathbf{x}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where $f(\mathbf{x}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate normal density (A.19) on page 553.

For the model $D \sim P_\theta, \theta \in \Theta$, we don't know $\theta$. We never know $\theta$. All we can do is guess. We will estimate $\theta$ (or a function of $\theta$) based on the observable data. Let $T$ denote an *estimator* of $\theta$ (or a function of $\theta$): $T = T(D)$ For example, if $D = X_1, \ldots, X_n \overset{i.i.d}{\sim} N(\mu, \sigma^2)$, the usual estimator is $T = (\overline{X}, S^2)$. For an ordinary fixed-$x$ multiple regression model, $T = (\widehat{\boldsymbol{\beta}}, MSE)$. In these and in all other cases, $T$ is a *statistic*, a random variable or vector that can be computed from the data without knowing the values of any unknown parameters.

How do we get a recipe for $T$? Guess? It's good to be systematic. Lots of methods are available. We will consider two: Method of moments and Maximum Likelihood.

## A.6.2 Method of Moments Estimation

The following is based on a random sample like $(X_1, Y_1), \ldots, (X_n, Y_n)$. Moments are quantities like $E\{X_i\}$, $E\{X_i^2\}$, $E\{X_i Y_i\}$, $E\{W_i X_i^2 Y_i^3\}$, and so on. *Central* moments are moments of *centered* random variables, such as

$$E\{(X_i - \mu_x)^2\}$$

$$E\{(X_i - \mu_x)(Y_i - \mu_y)\}$$

$$E\{(X_i - \mu_x)^2 (Y_i - \mu_y)^3 (Z_i - \mu_z)^2\}$$

These are all *population* moments. Sample moments are analogous to population moments, and are natural estimators.

| Population moment | Sample moment |
|---|---|
| $E\{X_i\}$ | $\frac{1}{n}\sum_{i=1}^{n} X_i$ |
| $E\{X_i^2\}$ | $\frac{1}{n}\sum_{i=1}^{n} X_i^2$ |
| $E\{X_i Y_i\}$ | $\frac{1}{n}\sum_{i=1}^{n} X_i Y_i$ |
| $E\{(X_i-\mu_x)^2\}$ | $\frac{1}{n}\sum_{i=1}^{n}(X_i-\overline{X}_n)^2$ |
| $E\{(X_i-\mu_x)(Y_i-\mu_y)\}$ | $\frac{1}{n}\sum_{i=1}^{n}(X_i-\overline{X}_n)(Y_i-\overline{Y}_n)$ |
| $E\{(X_i-\mu_x)(Y_i-\mu_y)^2\}$ | $\frac{1}{n}\sum_{i=1}^{n}(X_i-\overline{X}_n)(Y_i-\overline{Y}_n)^2$ |

The method of moments is based on estimating population moments by the corresponding sample moments. For the model $D \sim P_\theta$ with $\theta \in \Theta$, the population moments are a function of $\theta$. The procedure is to first find $\theta$ as a function of the population moments, and then estimate $\theta$ with that function of the *sample* moments.

Let $m$ denote a vector of population moments, and let $\widehat{m}$ denote the corresponding vector of sample moments. First, find $m = g(\theta)$. Then solve for $\theta$, obtaining $\theta = g^{-1}(m)$. Let $\widehat{\theta} = g^{-1}(\widehat{m})$. It doesn't matter if you solve first or put hats on first[11].

For example, suppose $X_1, \ldots, X_n \overset{i.i.d}{\sim} U(0,\theta)$. That is, the data are a random sample from a uniform distribution on $(0,\theta)$, so that the model density is $f(x) = \frac{1}{\theta}$ for $0 < x < \theta$. First, find the moment (expected value).

$$\begin{aligned} E(X_i) &= \int_0^\theta x\frac{1}{\theta}\,dx \\ &= \frac{1}{\theta}\int_0^\theta x\,dx \\ &= \frac{1}{\theta}\,\frac{x^2}{2}\Big|_0^\theta = \frac{1}{2\theta}(\theta^2-0) \\ &= \frac{\theta}{2} \end{aligned}$$

So $m = \frac{\theta}{2} \Leftrightarrow \theta = 2m$, and $\widehat{\theta} = 2\overline{X}$.

**Sample problem** Let $X_1, \ldots, X_n$ be a random sample from a uniform distribution on $(0,\theta)$. Estimate $\theta$ by the Method of Moments for the following data. Your answer is a number. Show some work. Data: `4.09 0.13 0.84 3.83 2.13 4.67 4.61 0.40 4.19 0.71`.

[11] For most models the function $g$ is well behaved, with continuous mixed partial derivatives. In that case the multivariate delta method from the end of Section A.5 guarantees that $\widehat{\theta}$ is asymptotically multivariate normal even when the data are definitely not normal. This yields distribution-free tests and confidence intervals with surprisingly little effort.

**Answer** $\overline{X} = 2.56$ so $\widehat{\theta} = 2\overline{X} = 2 * 2.56 = 5.12$.

Method of moments estimators are not unique. What moments you use are up to you.

$$E(X_i^2) = \frac{1}{\theta}\int_0^\theta x^2\,dx = \frac{\theta^2}{3}$$

So set $m = \frac{\theta^2}{3} \Leftrightarrow \theta = \sqrt{3m}$, and

$$\widehat{\theta} = \sqrt{\frac{3}{n}\sum_{i=1}^{n} X_i^2},$$

which is not equal to $2\overline{X}$. Presumably estimates based on lower-order moments are better in some sense, but I don't know the details.

To compare the two estimates $\widehat{\theta}_1 = 2\overline{X}$ and $\widehat{\theta}_2 = \sqrt{\frac{3}{n}\sum_{i=1}^{n} X_i^2}$ for the numerical example,

```
x     4.09   0.13   0.84    3.83   2.13    4.67    4.61   0.40  4.19   0.71
x^2  16.7281 0.0169 0.7056 14.6689 4.5369 21.8089 21.2521 0.16 17.5561 0.5041
```

yielding $\widehat{\theta}_1 = 5.12$ and $\widehat{\theta}_2 = 5.42$.

**Method of Moments estimator for the normal** Let $X_1, \ldots, X_n \overset{i.i.d}{\sim} N(\mu, \sigma^2)$. From the moment-generating function or a textbook, $E(X_i) = \mu$ and $E(X_i^2) = \sigma^2 + \mu^2$. Solving for the parameters, $\mu = E(X_i)$ and $\sigma^2 = E(X_i^2) - (E(X_i))^2$. The Method of Moments estimators are $\widehat{\mu} = \overline{X}$ and $\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \overline{X}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$.

**A regression example** Independently for $i = 1, \ldots, n$, let $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$, where

- $E(X_i) = \mu_x$, $Var(X_i) = \sigma_x^2$
- $E(\epsilon_i) = 0$, $Var(\epsilon_i) = \sigma_\epsilon^2$
- $X_i$ and $\epsilon_i$ are independent.

The distributions of $X_i$ and $\epsilon_i$ are unknown, so they are part of the parameter. The parameter is $(\beta_0, \beta_1, F_\epsilon(\epsilon), F_x(x))$. As mentioned earlier, there is no conceptual problem with parameters that are functions (infinite-dimensional) instead of just real numbers or vectors.

We want to estimate $\beta_0$ and $\beta_1$, a two-dimensional *function* of the parameter. First, calculate some moments.

$$\begin{array}{ll} E(X_i) = \mu_x & Var(X_i) = \sigma_x^2 \\ E(Y_i) = \beta_0 + \beta_1\mu_x & Cov(X_i, Y_i) = \beta_1\sigma_x^2 \end{array}$$

Use the Centering Rule on Page **??** to get the last one:

$$\begin{aligned} Cov(X_i, Y_i) &= E(\overset{c}{X}_i \overset{c}{Y}_i) \\ &= E\{\overset{c}{X}_i \, (\beta_1 \, \overset{c}{X}_i + \epsilon_i)\} \\ &= E\{\beta_1 \, \overset{c}{X}_i^{\,2} + \overset{c}{X}_i \, \epsilon_i)\} \\ &= \beta_1 E\{\overset{c}{X}_i^{\,2}\} + E\{\overset{c}{X}_i\} E\{\epsilon_i\} \\ &= \beta_1 \sigma_x^2 \end{aligned}$$

Putting hats on first (optional), we solve $\overline{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 \overline{X}$ and $\widehat{\sigma}_{xy} = \widehat{\beta}_1 \widehat{\sigma}_x^2$ for $\widehat{\beta}_0$ and $\widehat{\beta}_1$, obtaining

$$\begin{aligned} \widehat{\beta}_1 &= \frac{\widehat{\sigma}_{xy}}{\widehat{\sigma}_x^2} = \frac{\sum_{i=1}^n (X_i - \overline{X}_n)(Y_i - \overline{Y}_n)}{\sum_{i=1}^n (X_i - \overline{X}_n)^2} \text{ and} \\ \widehat{\beta}_0 &= \overline{Y} - \widehat{\beta}_1 \overline{X} \end{aligned}$$

These happen to be the same as the least-squares estimates.

Since $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are nice differentiable functions of various quantities that are essentially sample means, the multivariate delta method from the end of Section A.5 implies that the asymptotic joint distribution of $\widehat{\beta}_0$ and $\widehat{\beta}_1$ is bivariate normal. This holds regardless of the distributions of $X_i$ and $\epsilon_i$, provided only that their moments exist, and opens the door to distribution-free tests and confidence intervals. The story for multiple regression is almost exactly the same. The only requirement is a sample large enough for the Central Limt Theorem to work.

### A.6.3 Maximum Likelihood Estimation

The idea behind maximum likelihood is to estimate the unknown parameter by the quantity that makes the probability of obtaining the observed data as large as possible. This probability is represented[12] by the likelihood function

$$L(\theta) = \prod_{i=1}^{n} f(d_i; \theta),$$

where $f(d_i; \theta)$ is the density or probability mass function evaluated at $d_i$.

Let $\widehat{\theta}$ denote the usual Maximum Likelihood Estimate (MLE). That is, it is the parameter value for which the likelihood function is greatest, over all $\theta \in \Theta$. Because the log is an increasing function, maximizing the likelihood is equivalent to maximizing the log likelihood, which will be denoted

$$\ell(\theta) = \ln L(\theta).$$

[12]If the data are discrete, the likelihood function is exactly the probability of observing the data that actually were observed. In the continuous case the likelihood function is approximately proportional to the probability of observing a data vector that falls into a small region surrounding the vector (point) that was observed.

In elementary situations where the support of the distribution does not depend on the parameter, you get the MLE by closing your eyes, differentiating the log likelihood, setting the derivative to zero, and solving for $\theta$. Then if you are being careful, you carry out the second derivative test; if $\ell''(\widehat{\theta}) < 0$, the log likelihood is concave down at your answer, and you have found the maximum. Here is an example, useful mostly to clarify ideas and serve as a contrast to more realistic cases.

**Example** Let $D_1, \ldots, D_n$ be a random sample (independent and identically distributed random variables) from a distribution with density $f(y) = \frac{\theta}{(d+1)^{\theta+1}}$ for $d > 0$, where the unknown parameter $\theta$ is strictly greater than zero. The log likelihood is

$$\begin{aligned}
\ell(\theta) &= \ln \prod_{i=1}^{n} \frac{\theta}{(d_i+1)^{\theta+1}} \\
&= \sum_{i=1}^{n} (\ln\theta - (\theta+1)\ln(d_i+1)) \\
&= n\ln\theta - (\theta+1)\sum_{i=1}^{n} \ln(d_i+1)
\end{aligned}$$

Differentiating with respect to $\theta$,

$$\begin{aligned}
\ell'(\theta) &= \frac{n}{\theta} - \sum_{i=1}^{n} \ln(d_i+1) \stackrel{\text{set}}{=} 0 \\
&\Rightarrow \quad \theta = \frac{n}{\sum_{i=1}^{n} \ln(d_i+1)}.
\end{aligned}$$

Carrying out the second derivative test,

$$\ell''(\theta) = -n\theta^{-2} = -\frac{n}{\theta^2} < 0,$$

so the log likelihood function is concave down and we have located a maximum. This justifies writing $\widehat{\theta} = n/\sum_{i=1}^{n} \ln(d_i+1)$. In R, if the data were in a numeric vector called `d`, the MLE would be `thetahat = 1/mean(log(d+1))`.

### Some Very Basic Math

If the calculations in that last example seemed obvious, you can skip this section.

I have noticed that a major obstacle for many students when doing maximum likelihood calculations is a set of basic mathematical operations they actually know. But the mechanics are rusty, or the notation used in Statistics is troublesome. So, with sincere apologies to those who don't need this, here are some basic rules.

- The distributive law: $a(b+c) = ab + ac$. You may see this in a form like

$$\theta \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} \theta x_i$$

- Power of a product is the product of powers: $(ab)^c = a^c\, b^c$. You may see this in a form like
$$\left(\prod_{i=1}^{n} x_i\right)^{\alpha} = \prod_{i=1}^{n} x_i^{\alpha}$$
- Multiplication is addition of exponents: $a^b a^c = a^{b+c}$. You may see this in a form like
$$\prod_{i=1}^{n} \theta e^{-\theta x_i} = \theta^n \exp(-\theta \sum_{i=1}^{n} x_i)$$
- Powering is multiplication of exponents: $(a^b)^c = a^{bc}$. You may see this in a form like
$$(e^{\mu t + \frac{1}{2}\sigma^2 t^2})^n = e^{n\mu t + \frac{1}{2} n\sigma^2 t^2}$$
- Log of a product is sum of logs: $\ln(ab) = \ln(a) + \ln(b)$. You may see this in a form like
$$\ln \prod_{i=1}^{n} x_i = \sum_{i=1}^{n} \ln x_i$$
- Log of a power is the exponent times the log: $\ln(a^b) = b \ln(a)$. You may see this in a form like
$$\ln(\theta^n) = n \ln \theta$$
- The log is the inverse of the exponential function: $\ln(e^a) = a$. You may see this in a form like
$$\ln\left(\theta^n \exp(-\theta \sum_{i=1}^{n} x_i)\right) = n \ln \theta - \theta \sum_{i=1}^{n} x_i$$

**Exercises A.6.3**

1. Choose the correct answer.
   (a) $\prod_{i=1}^{n} e^{x_i} =$
      i. $\exp(\prod_{i=1}^{n} x_i)$
      ii. $e^{nx_i}$
      iii. $\exp(\sum_{i=1}^{n} x_i)$
   (b) $\prod_{i=1}^{n} \lambda e^{-\lambda x_i} =$
      i. $\lambda e^{-\lambda^n x_i}$
      ii. $\lambda^n e^{-\lambda n x_i}$
      iii. $\lambda^n \exp(-\lambda \sum_{i=1}^{n} x_i)$
      iv. $\lambda^n \exp(-n\lambda \sum_{i=1}^{n} x_i)$
      v. $\lambda^n \exp(-\lambda^n \sum_{i=1}^{n} x_i)$

(c) $\prod_{i=1}^{n} a_i^b =$
   i. $na^b$
   ii. $a^{nb}$
   iii. $(\prod_{i=1}^{n} a_i)^b$

(d) $\prod_{i=1}^{n} a^{b_i} =$
   i. $na^{b_i}$
   ii. $a^{nb_i}$
   iii. $\sum_{i=1}^{n} a^{b_i}$
   iv. $a^{\prod_{i=1}^{n} b_i}$
   v. $a^{\sum_{i=1}^{n} b_i}$

(e) $\left(e^{\lambda(e^t-1)}\right)^n =$
   i. $ne^{\lambda(e^t-1)}$
   ii. $e^{n\lambda(e^t-1)}$
   iii. $e^{\lambda(e^{nt}-1)}$
   iv. $e^{n\lambda(e^t-n)}$

(f) $\left(\prod_{i=1}^{n} e^{-\lambda x_i}\right)^2 =$
   i. $e^{-2n\lambda x_i}$
   ii. $e^{-2\lambda \sum_{i=1}^{n} x_i}$
   iii. $2e^{-\lambda \sum_{i=1}^{n} x_i}$

2. True, or False?

(a) $\sum_{i=1}^{n} \frac{1}{x_i} = \frac{1}{\sum_{i=1}^{n} x_i}$

(b) $\prod_{i=1}^{n} \frac{1}{x_i} = \frac{1}{\prod_{i=1}^{n} x_i}$

(c) $\frac{a}{b+c} = \frac{a}{b} + \frac{a}{c}$

(d) $\ln(a+b) = \ln(a) + \ln(b)$

(e) $e^{a+b} = e^a + e^b$

(f) $e^{a+b} = e^a e^b$

(g) $e^{ab} = e^a e^b$

(h) $\prod_{i=1}^{n}(x_i + y_i) = \prod_{i=1}^{n} x_i + \prod_{i=1}^{n} y_i$

(i) $\ln(\prod_{i=1}^{n} a_i^b) = b \sum_{i=1}^{n} \ln(a_i)$

(j) $\sum_{i=1}^{n} \prod_{j=1}^{n} a_j = n \prod_{j=1}^{n} a_j$

(k) $\sum_{i=1}^{n} \prod_{j=1}^{n} a_i = \sum_{i=1}^{n} a_i^n$

(l) $\sum_{i=1}^{n} \prod_{j=1}^{n} a_{i,j} = \prod_{j=1}^{n} \sum_{i=1}^{n} a_{i,j}$

3. Simplify as much as possible.

(a) $\ln \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i}$

(b) $\ln \prod_{i=1}^{n} \binom{m}{x_i}\theta^{x}(1-\theta)^{m-x_i}$

(c) $\ln \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$

(d) $\ln \prod_{i=1}^{n} \theta(1-\theta)^{x_i-1}$

(e) $\ln \prod_{i=1}^{n} \frac{1}{\theta}e^{-x_i/\theta}$

(f) $\ln \prod_{i=1}^{n} \frac{1}{\beta^{\alpha}\Gamma(\alpha)}e^{-x_i/\beta}x_i^{\alpha-1}$

(g) $\ln \prod_{i=1}^{n} \frac{1}{2^{\nu/2}\Gamma(\nu/2)}e^{-x_i/2}x_i^{\nu/2-1}$

(h) $\ln \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$

(i) $\prod_{i=1}^{n} \frac{1}{\beta-\alpha}I(\alpha \leq x_i \leq \beta)$ (Express in terms of the minimum and maximum $y_1$ and $y_n$.)

### Maximum likelihood for the multivariate normal

Maximum likelihood estimation for the multivariate normal distribution plays an important role in this book. It's a case where closing your eyes and differentiating will get you nowhere. It's helpful to express the MLE as a theorem, making it easy to reference in the main body of the text.

**Theorem A.2** *Let* $\mathbf{x}_1, \ldots, \mathbf{x}_n$ *be a random sample from a* $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *distribution. The unique maximum likelihood estimate is* $\widehat{\boldsymbol{\mu}} = \overline{\mathbf{x}}$ *and* $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^\top$.

When I am producing proofs for a student audience, I frequently wonder whether I should provide a model of how to write a clean proof, or give a longer proof that is easier to follow. Perhaps because I'm naturally long-winded anyway, I often wind up giving more detail. Here, I will try doing it both ways. The brief one comes first. If you can fill in the gaps without too much effort, great. If necessary or if you wish, look at the second proof.

**Proof One** Rather than maximizing the likelihood, equivalently minimize

$$-\frac{2}{n}\log\frac{L(\boldsymbol{\mu},\boldsymbol{\Sigma})}{L(\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{\Sigma}})} = tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}| - p + (\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu}).$$

Because $\boldsymbol{\Sigma}$ is positive definite, the last term is nonnegative, and equal to zero if and only if $\boldsymbol{\mu} = \overline{\mathbf{x}}$. Setting $\boldsymbol{\mu} = \overline{\mathbf{x}}$, the task is now to minimize $tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}|$.

The matrix $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}$ is similar to $\boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}$, so the eigenvalues of $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}$ are real, and positive with probability one. Thus,

$$tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}| = \sum_{j=1}^{p}\lambda_j - \sum_{j=1}^{p}\log\lambda_j = \sum_{j=1}^{p}(\lambda_j - \log\lambda_j).$$

Each term in the sum is positive, and uniquely minimized when $\lambda_j = 1$. So to maximize the likelihood, all the eigenvalues of $\boldsymbol{\Sigma}$ must equal one. By the spectral decomposition (A.9), $\boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2} = \mathbf{C}\mathbf{D}\mathbf{C}^\top = \mathbf{C}\mathbf{I}_p\mathbf{C}^\top = \mathbf{I}_p$, so that $\boldsymbol{\Sigma} = \widehat{\boldsymbol{\Sigma}}$. ■

**Proof Two** Rather than maximizing the likelihood, equivalently, (1) Divide the likelihood by a well-chosen expression that is constant with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, (2) Take the natural log, (3) Multiply by $-\frac{2}{n}$, and (4) minimize the result. Using Property 9 of the multivariate normal on page 554,

$$\begin{aligned}
-\frac{2}{n}\log\frac{L(\boldsymbol{\mu},\boldsymbol{\Sigma})}{L(\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{\Sigma}})} &= -\frac{2}{n}\log\frac{|\boldsymbol{\Sigma}|^{-n/2}(2\pi)^{-np/2}\exp-\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})+(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu})\right\}}{|\widehat{\boldsymbol{\Sigma}}|^{-n/2}(2\pi)^{-np/2}\exp-\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Sigma}}^{-1})+(\overline{\mathbf{x}}-\overline{\mathbf{x}})^\top\widehat{\boldsymbol{\Sigma}}^{-1}(\overline{\mathbf{x}}-\overline{\mathbf{x}})\right\}} \\
&= -\frac{2}{n}\log\frac{|\boldsymbol{\Sigma}|^{-n/2}\exp-\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})+(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu})\right\}}{|\widehat{\boldsymbol{\Sigma}}|^{-n/2}\exp-\frac{n}{2}\left\{tr(\mathbf{I}_p)+0\right\}} \\
&= -\frac{2}{n}\log\left(\frac{|\boldsymbol{\Sigma}|\exp\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})+(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu})\right\}}{|\widehat{\boldsymbol{\Sigma}}|\exp\{p\}}\right)^{-\frac{n}{2}} \\
&= \log\left(\frac{|\boldsymbol{\Sigma}|\exp\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})+(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu})\right\}}{|\widehat{\boldsymbol{\Sigma}}|e^p}\right) \\
&= \log\frac{|\boldsymbol{\Sigma}|}{|\widehat{\boldsymbol{\Sigma}}|}+tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})+(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu})-p \\
&= tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})-\log\frac{|\widehat{\boldsymbol{\Sigma}}|}{|\boldsymbol{\Sigma}|}-p+(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu}) \\
&= tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})-\log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}|-p+(\overline{\mathbf{x}}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}}-\boldsymbol{\mu})
\end{aligned}$$

If $\boldsymbol{\Sigma}$ is positive definite, so is $\boldsymbol{\Sigma}^{-1}$. Therefore the last term is nonnegative, and equal to zero if and only if $\overline{\mathbf{x}}-\boldsymbol{\mu}=\mathbf{0} \iff \boldsymbol{\mu}=\overline{\mathbf{x}}$. That is, the function is minimized when $\boldsymbol{\mu}=\overline{\mathbf{x}}$, regardless of what the positive definite matrix $\boldsymbol{\Sigma}$ happens to be.

This establishes $\widehat{\boldsymbol{\mu}}=\overline{\mathbf{x}}$. Setting $\boldsymbol{\mu}=\overline{\mathbf{x}}$, the last term vanishes, and the task is now to minimize

$$tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})-\log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}| \tag{A.25}$$

over all symmetric and positive definite $\boldsymbol{\Sigma}$.

Recall that the square matrix $\mathbf{B}$ is said to be *similar* to $\mathbf{A}$ if there is an invertible matrix $\mathbf{P}$ with $\mathbf{B}=\mathbf{P}^{-1}\mathbf{A}\mathbf{P}$. Similar matrices share important characteristics; for example, they have the same eigenvalues, and the numbers of times each eigenvalue occurs (the multiplicities) are the same for the two matrices.

Choosing $\mathbf{P}=\boldsymbol{\Sigma}^{-1/2}$, write

$$\boldsymbol{\Sigma}^{-1/2}\left(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{\Sigma}^{1/2}=\boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}.$$

Thus $\boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}$ is similar to $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}$. The matrix $\widehat{\boldsymbol{\Sigma}}$ has an inverse with probability

one[13]. Therefore the symmetric matrix $\boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}$ has an inverse, is positive definite, and all its eigenvalues are strictly positive. This means the eigenvalues of $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}$ are positive too, and

$$\begin{aligned} tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}| &= \sum_{j=1}^{p}\lambda_j - \log\prod_{j=1}^{p}\lambda_j \\ &= \sum_{j=1}^{p}\lambda_j - \sum_{j=1}^{p}\log\lambda_j \\ &= \sum_{j=1}^{p}(\lambda_j - \log\lambda_j)\,. \end{aligned} \tag{A.26}$$

For $x > 0$, the function $y = x - \log x > 0$, and achieves a unique minimum when $x = 1$. Thus (A.26) can be minimized by choosing $\boldsymbol{\Sigma}$ so that the eigenvalues of $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}$ all equal one. Such a choice is possible, because $\boldsymbol{\Sigma} = \widehat{\boldsymbol{\Sigma}}$ yields $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1} = \mathbf{I}_p$. The conclusion is that $\widehat{\boldsymbol{\Sigma}}$ is a maximum likelihood estimator of $\boldsymbol{\Sigma}$. Now we will see it is the only one. Let $\boldsymbol{\Sigma}$ be another covariance matrix such that all the eigenvalues of $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}$ equal one.

The similarity of $\boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}$ to $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}$ means that the eigenvalues of $\boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}$ are also all equal to one. Thus by the spectral decomposition theorem (A.9),

$$\begin{aligned} \boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2} &= \mathbf{C}\mathbf{D}\mathbf{C}^\top \\ &= \mathbf{C}\mathbf{I}_p\mathbf{C}^\top = \mathbf{C}\mathbf{C}^\top = \mathbf{I}_p, \end{aligned}$$

because the eigenvectors in the columns of $\mathbf{C}$ are orthonormal. Then,

$$\begin{aligned} & \mathbf{I}_p = \boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2} \\ \iff \quad & \boldsymbol{\Sigma}^{1/2}\,\mathbf{I}_p\,\boldsymbol{\Sigma}^{1/2} = \boldsymbol{\Sigma}^{1/2}\left(\boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1/2}\right)\boldsymbol{\Sigma}^{1/2} \\ \iff \quad & \boldsymbol{\Sigma} = \widehat{\boldsymbol{\Sigma}}. \end{aligned}$$

This establishes that with probability one, the likelihood function has a unique maximum at $\boldsymbol{\mu} = \overline{\mathbf{x}}$ and and $\boldsymbol{\Sigma} = \widehat{\boldsymbol{\Sigma}}$. ■

### A.6.4 Numerical maximum likelihood

In this course, as in much of applied statistics, you will find that you can write the log likelihood and differentiate it easily enough, but when you set the derivatives to zero, you obtain a set of equations that are impossible to solve explicitly. This means that the problem needs to be solved numerically. That is, you use a computer to calculate the

[13]The multivariate normal distribution is continuous, and for that reason, so is the joint distribution of the unique variances and covariances in $\widehat{\boldsymbol{\Sigma}}$. The set of variances and covariances such that one of the columns is a linear combination of others is a set of volume zero in $\mathbb{R}^{p(p+1)/2}$, and hence has probability zero.

value of the log likelihood for a set of parameter values, and you search until you have found the biggest one.

But how do you search? It's easy in one or two dimensions, but structural equation models can easily involve dozens, scores or even hundreds of parameters. It's a bit like being dropped by helicopter onto a mountain range, and asked to find the highest peak blindfolded. All you can do is walk uphill. The gradient is the direction of steepest increase, so walk that way. How big a step should you take? That's a good question. When you come to a place where the surface is level, or approximately level, stop. How level is level enough? That's another good question. Once you find a "level" place, you can check to see if the surface is concave down there. If so, you're at a maximum. Is it the global maximum (the real MLE), or just a local maximum? It's usually impossible to tell for sure. You can get the helicopter to drop you in several different places fairly far apart, and if you always arrive at the same maximum you will feel more confident of your answer. But it could still be just a local maximum that is easy to reach. The main thing to observe is that where you start is *very* important. Another point is that for realistically big problems, you need high-grade, professionally written software.

The following example is one that you can do by hand, though maybe not with your eyes closed. But it will serve to illustrate the basic ideas of numerical maximum likelihood.

**Example A.6.1** *Normal with Mean Equal to Standard Deviation*

Let $D_1, \ldots, D_n$ be a random sample from a normal distribution with mean $\theta$ and variance $\theta^2$. A sample of size 50 yields:

```
  5.85 -15.02 -13.24  -1.63  -0.07  -2.40  -3.02  -3.19  -5.16   0.79  -1.03 -10.69
-12.96  -4.55   0.57  -7.94  -6.80   2.95  -9.01  -9.33 -11.93  -7.13  10.34  -1.01
 -4.18  -1.30  -7.56  -1.25  -4.64  -4.88  -4.06  -1.91  -1.81  -6.92 -13.27  -5.52
  4.40 -12.17  -4.55  -5.82  -0.81 -19.28  -4.97  -7.78  -5.07  -5.45  -4.27  -4.98
 -9.56  -9.33
```

Find the maximum likelihood estimate of $\theta$. You only need an approximate value; one decimal place of accuracy will do.

Again, this is a problem that can be solved explicitly by differentiation, and the reader is invited to give it a try before proceeding. Have the answer? Is it still the same day you started? Now for the numerical solution. First, write the log likelihood as

$$\begin{aligned} \ell(\theta) &= \ln \prod_{i=1}^{n} \frac{1}{|\theta|\sqrt{2\pi}} e^{-\frac{(d_i-\theta)^2}{2\theta^2}} \\ &= -n \ln |\theta| - \frac{n}{2} \ln(2\pi) - \frac{\sum_{i=1}^{n} d_i^2}{2\theta^2} + \frac{\sum_{i=1}^{n} d_i}{\theta} - \frac{n}{2}. \end{aligned}$$

We will do this in `R`. The data are in a file called `norm1.data`. Read it. Remember that $>$ is the `R` prompt.

```
> D <- scan("norm1.data")
Read 50 items
```

Now define a function to compute the log likelihood.

```
loglike1 <- function(theta) # Assume data are in a vector called D
    {
    sumdsq <- sum(D^2); sumd <- sum(D); n <- length(D)
    loglike1 <- -n * log(abs(theta)) - (n/2)*log(2*pi) - sumdsq/(2*theta^2) +
                sumd/theta - n/2
    loglike1 # Return value of function
    } # End definition of function loglike1
```

Just to show how the function works, compute it at a couple of values, say $\theta = 2$ and $\theta = -2$.

```
> loglike1(2)
[1] -574.2965
> loglike1(-2)
[1] -321.7465
```

Negative values of the parameter look more promising, but it is time to get systematic. The following is called a *grid search*. It is brutal, inefficient, and usually effective. It is too slow to be practical for large problems, but this is a one-dimensional parameter and we are only asked for one decimal place of accuracy. Where should we start? Since the parameter is the mean of the distribution, it should be safe to search within the range of the data. Start with widely spaced values and then refine the search. All we are doing is to calculate the log likelihood for a set of (equally spaced) parameter values and see where it is greatest. After all, that is the *idea* behind the MLE.

```
> min(D); max(D)
[1] -19.28
[1] 10.34
> Theta <- -20:10
> cbind(Theta,loglike1(Theta))
      Theta
 [1,]   -20  -211.5302
 [2,]   -19  -208.6709
 [3,]   -18  -205.6623
 [4,]   -17  -202.4911
 [5,]   -16  -199.1423
 [6,]   -15  -195.6002
 [7,]   -14  -191.8486
 [8,]   -13  -187.8720
 [9,]   -12  -183.6580
[10,]   -11  -179.2022
```

```
[11,]   -10  -174.5179
[12,]    -9  -169.6565
[13,]    -8  -164.7513
[14,]    -7  -160.1163
[15,]    -6  -156.4896
[16,]    -5  -155.6956
[17,]    -4  -162.7285
[18,]    -3  -193.8796
[19,]    -2  -321.7465
[20,]    -1 -1188.0659
[21,]     0        NaN
[22,]     1 -1693.1659
[23,]     2  -574.2965
[24,]     3  -362.2463
[25,]     4  -289.0035
[26,]     5  -256.7156
[27,]     6  -240.6729
[28,]     7  -232.2734
[29,]     8  -227.8888
[30,]     9  -225.7788
[31,]    10  -225.0279
```

First, we notice that at $\theta = 0$, the log likelihood is indeed Not a Number. For this problem, the parameter space is all the real numbers except zero – unless one wants to think of a normal random variable with zero variance as being degenerate at $\mu$; that is, $P(D = \mu) = 1$. (In his case, what would the data look like?)

But the log likelihood is greatest around $\theta = -5$. We are asked for one decimal place of accuracy, so,

```
> Theta <- seq(from=-5.5,to=-4.5,by=0.1)
> Loglike <- loglike1(Theta)
> cbind(Theta,Loglike)
      Theta   Loglike
 [1,]  -5.5 -155.5445
 [2,]  -5.4 -155.4692
 [3,]  -5.3 -155.4413
 [4,]  -5.2 -155.4660
 [5,]  -5.1 -155.5487
 [6,]  -5.0 -155.6956
 [7,]  -4.9 -155.9136
 [8,]  -4.8 -156.2106
 [9,]  -4.7 -156.5950
[10,]  -4.6 -157.0767
[11,]  -4.5 -157.6665
> thetahat <- Theta[Loglike==max(Loglike)]
```

```
>             # Theta such that Loglike is the maximum of Loglike
> thetahat
[1] -5.3
```

To one decimal place of accuracy, the maximum is at $\theta = -5.3$. It would be easy to refine the grid and get more accuracy, but that will do. This is the last time we will see our friend the grid search, but you may find the approach useful in homework.

Now let's do the search in a more sophisticated way, using R's `nlm` (non-linear minimization) function. [14] The `nlm` function has quite a few arguments; try `help(nlm)`. The ones you always need are the first two: the name of the function, and a starting value (or vector of starting values, for multiparameter problems).

Where should we start? Since the parameter equals the expected value of the distribution, how about the sample mean? It is often a good strategy to use Method of Moment estimators as starting values for numerical maximum likelihood. Method of Moments estimation is reviewed in Section **??**.

One characteristic that `nlm` shares with most optimization routines is that it likes to *minimize* rather than maximizing. So we will minimize the negative of the log likelihood function. For this, it is necessary to define a new function, `loglike2`.

```
> mean(D)
[1] -5.051
> loglike2 <- function(theta) { loglike2 <- -loglike1(theta); loglike2 }
> nlm(loglike2,mean(D))
$minimum
[1] 155.4413

$estimate
[1] -5.295305

$gradient
[1] -1.386921e-05

$code
[1] 1

$iterations
[1] 4
```

By default, `nlm` returns a list with four elements; `minimum` is the value of the function at the point where it reaches its minimum, `estimate` is the value at which the minimum

[14] The `nlm` function is good but generic. See Numerical Recipes for a really good discussion of routines for numerically minimizing a function. They also provide source code. The *Numerical Recipes* books have versions for the Pascal, Fortran and Basic languages as well as C. This is a case where a book definitely delivers more than the title promises. It may be a cookbook, but it is a very good cookbook written by expert chemists.

was located; that's the MLE. `Gradient` is the slope in the direction of greatest increase; it should be near zero. `Code` is a diagnosis of how well the optimization went; the value of 1 means everything seemed okay. See `help(nlm)` for more detail.

We could have gotten just the MLE with

```
> nlm(loglike2,mean(D))$estimate
[1] -5.295305
```

That's the answer, but the numerical approach misses some interesting features of the problem, which can be done with paper and pencil in this simple case. Differentiating the log likelihood separately for $\theta < 0$ and $\theta > 0$ to get rid of the absolute value sign, and then re-uniting the two cases since the answer is the same, we get

$$\ell'(\theta) = -\frac{n}{\theta} + \frac{\sum_{i=1}^n d_i^2}{\theta^3} - \frac{\sum_{i=1}^n d_i}{\theta^2}.$$

Setting $\ell'(\theta) = 0$ and re-arranging terms, we get

$$n\theta^2 + (\sum_{i=1}^n d_i)\theta - (\sum_{i=1}^n d_i^2) = 0.$$

Of course this expression is not valid at $\theta = 0$, because the function we are differentiating is not even defined there. The quadratic formula yields two solutions:

$$\frac{-\sum_{i=1}^n d_i \pm \sqrt{(\sum_{i=1}^n d_i)^2 + 4n\sum_{i=1}^n d_i^2}}{2n} = \frac{1}{2}\left(-\overline{d} \pm \sqrt{\overline{d}^2 + 4\frac{\sum_{i=1}^n d_i^2}{n}}\right), \tag{A.27}$$

where $\overline{d}$ is the sample mean.

Let's calculate these for the given data.

```
> meand <- mean(D) ; meandsq <- sum(D^2)/length(D)
> (-meand + sqrt(meand^2 + 4*meandsq) )/2
[1] 10.3463
> (-meand - sqrt(meand^2 + 4*meandsq) )/2
[1] -5.2953
```

The second solution is the one we found with the numerical search. What about the other one? Is it a minimum? Maximum? Saddle point? The second derivative test will tell us. The second derivative is

$$\ell''(\theta) = \frac{n}{\theta^2} - \frac{3\sum_{i=1}^n d_i^2}{\theta^4} + \frac{2\sum_{i=1}^n d_i}{\theta^3}.$$

Substituting A.27 into this does not promise to be much fun, so we will be content with a numerical answer for this particular data set. Call the first root `t1` and the second one (our MLE) `t2`.

```
> t1 <- (-meand + sqrt(meand^2 + 4*meandsq) )/2 ; t1
[1] 10.3463
> t2 <- (-meand - sqrt(meand^2 + 4*meandsq) )/2 ; t2
[1] -5.2953
> n <- length(D)
> # Now calculaate second derivative at t1 and t2
> n/t1^2 - 3*sum(D^2)/t1^4 + 2*sum(D)/t1^3
[1] -0.7061484
> n/t2^2 - 3*sum(D^2)/t2^4 + 2*sum(D)/t2^3
[1] -5.267197
```

The second derivative is negative in both cases; they are both local maxima! Which peak is higher?

```
> loglike1(t1)
[1] -224.9832
> loglike1(t2)
[1] -155.4413
```

So the maximum we found is higher, which makes sense because it's within the range of the data. But we only found it because we started searching near the correct answer.

Let's plot the log likelihood function, and see what this thing looks like. We know that because the natural log function goes to minus infinity as its (positive) argument approaches zero, the log likelihood plunges to $-\infty$ at $\theta = 0$. A plot would look like a giant icicle and we would not be able to see any detail where it matters. So we will zoom in by limiting the range of the $y$ axis. Here is the R code.

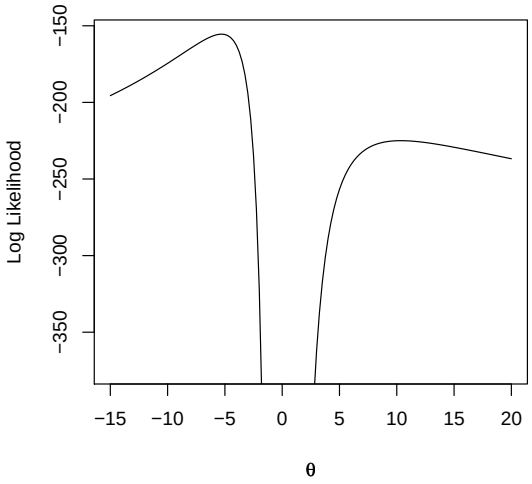
```
Theta <- seq(from=-15,to=20,by=0.25); Theta <- Theta[Theta!=0]
Loglike <- loglike1(Theta)
# Check where to break off the icicle
max(Loglike); Loglike[Theta==-3];  Loglike[Theta==3]

plot(Theta,Loglike,type='l',xlim=c(-15,20),ylim=c(-375,-155),
    xlab=expression(theta),ylab="Log Likelihood")
    # This is how you get Greek letters.
```

Here is the picture. You can see the local maxima around $\theta = -5$ and $\theta = 10$, and also that the one for negative $\theta$ is a higher.

Presumably we would have reached the bad answer if we had started the search in a bad place. Let's try starting the search at $\theta = +3$.

```
> nlm(loglike2,3)
$minimum
[1] 283.7589
```

Figure A.1: Log Likelihood for Example A.6.1




```
$estimate
[1] 64.83292

$gradient
[1] 0.701077

$code
[1] 4

$iterations
[1] 100
```

What happened?! The answer is way off, nowhere near the positive root of 10.3463. And the `minimum` (of *minus* the log likelihood) is over 283, when it would have been 224.9832 at $\theta = 10.3463$.

What happened was that the slope of the function was very steep at our starting value of $\theta = 3$, so `nlm` took a huge step in a positive direction. It was too big, and landed in a nearly flat place. Then `nlm` wandered around until it ran out of its default number of iterations (notice `iterations`=100). The exit `code` of 4 means maximum number of iterations exceeded.

It should be better if we start close to the answer, say at $\theta = 8$.

```
> nlm(loglike2,8)
$minimum
[1] 224.9832
```

```
$estimate
[1] 10.34629

$gradient
[1] -4.120564e-08

$code
[1] 1

$iterations
[1] 6
```

That's better. The moral of this story is clear. Good starting are *very* important.

Now let us look at an example of a multi-parameter problem where an explicit formula for the MLE is impossible, and numerical methods are required.

**Example A.6.2** *The Gamma Distribution*

Let $D_1, \ldots, D_n$ be a random sample from a Gamma distribution with parameters $\alpha > 0$ and $\beta > 0$. The probability density function is

$$f(x; \alpha, \beta) = \frac{1}{\beta^\alpha \Gamma(\alpha)} e^{-x/\beta} x^{\alpha-1}$$

for $x > 0$, and zero otherwise. Here is a random sample of size $n = 50$. For this example, the data are simulated using `R`, with known parameter values $\alpha = 2$ and $\beta = 3$. The seed for the random, number generator is set so the pseudo-random numbers can be recovered if necessary.

```
> set.seed(3201); alpha=2; beta=3
> D <- round(rgamma(50,shape=alpha, scale=beta),2); D
 [1] 20.87 13.74  5.13  2.76  4.73  2.66 11.74  0.75 22.07 10.49  7.26  5.82 13.08
[14]  1.79  4.57  1.40  1.13  6.84  3.21  0.38 11.24  1.72  4.69  1.96  7.87  8.49
[27]  5.31  3.40  5.24  1.64  7.17  9.60  6.97 10.87  5.23  5.53 15.80  6.40 11.25
[40]  4.91 12.05  5.44 12.62  1.81  2.70  3.03  4.09 12.29  3.23 10.94
> mean(D); alpha*beta
[1] 6.8782
[1] 6
> var(D); alpha*beta^2
[1] 24.90303
[1] 18
```

The parameter vector $\boldsymbol{\theta} = (\alpha, \beta)$, and the parameter space $\Theta$ is the first quadrant of $\mathbb{R}^2$.

$$\Theta = \{(\alpha, \beta) : \alpha > 0, \beta > 0\}$$

The log likelihood is

$$\begin{aligned}\ell(\alpha,\beta) &= \ln\prod_{i=1}^{n}\frac{1}{\beta^{\alpha}\Gamma(\alpha)}e^{-d_i/\beta}d_i^{\alpha-1}\\ &= \ln\left(\beta^{-n\alpha}\,\Gamma(\alpha)^{-n}\exp(-\frac{1}{\beta}\sum_{i=1}^{n}d_i)\left(\prod_{i=1}^{n}d_i\right)^{\alpha-1}\right)\\ &= -n\alpha\ln\beta - n\ln\Gamma(\alpha) - \frac{1}{\beta}\sum_{i=1}^{n}d_i + (\alpha-1)\sum_{i=1}^{n}\ln d_i.\end{aligned}$$

The next step would be to partially differentiate the log likelihood with respect to $\alpha$ and $\beta$, set both partial derivatives to zero, and solve two equations in two unknowns. But even if you are confident that the gamma function is differentiable (it is), you will be unable to solve the equations. It has to be done numerically.

Define an `R` function for the minus log likelihood. Notice the `lgamma` function, a direct numerical approximation of $\ln\Gamma(\alpha)$. The plan is to numerically minimize the minus log likelihood function over all $(\alpha,\beta)$ pairs, for this particular set of data values.

```
> # Gamma minus log likelihood: alpha=a, beta=b
> gmll <- function(theta,datta)
+      {
+       a <- theta[1]; b <- theta[2]
+       n <- length(datta); sumd <- sum(datta); sumlogd <- sum(log(datta))
+       gmll <- n*a*log(b) + n*lgamma(a) + sumd/b - (a-1)*sumlogd
+      gmll
+      } # End function gmll
```

Where should the numerical search start? One approach is to start at reasonable estimates of $\alpha$ and $\beta$ — estimates that can be calculated directly rather than by a numerical approximation. As in Example A.6.1, Method of Moments estimators are a convenient, high-quality choice.

For a gamma distribution, $E(D)=\alpha\beta$ and $Var(D)=\alpha\beta^2$. So,

$$\alpha=\frac{E(D)^2}{Var(D)} \quad \text{and} \quad \beta=\frac{Var(D)}{E(D)}.$$

Replacing population moments by sample moments and writing $\widetilde{\alpha}$ and $\widetilde{\beta}$ for the resulting Method of Moments estimators, we obtain

$$\widetilde{\alpha}=\frac{\overline{D}^2}{S_D^2} \quad \text{and} \quad \widetilde{\beta}=\frac{S_D^2}{\overline{D}},$$

where $\overline{D}$ is the sample mean and $S_D^2$ is the sample variance. For these data, the Method of Moments estimates are reasonably close to the correct values of $\alpha=2$ and $\beta=3$, but they are not perfect. Parameter estimates are not the same as parameters!

```
> momalpha <- mean(D)^2/var(D); momalpha
[1] 1.899754
> mombeta <- var(D)/mean(D); mombeta
[1] 3.620574
```

Now for the numerical search. This time, we will request that the `nlm` function return the *Hessian* at the place where the search stops. The Hessian is defined as follows. Suppose we are minimizing a function $g(\theta_1, \ldots, \theta_k)$ – say, a minus log likelihood. The Hessian is a $k \times k$ matrix of mixed partial derivatives. It may be written in terms of its $(i, j)$ element s

$$\mathbf{H} = \left[ \frac{\partial^2 g}{\partial \theta_i \partial \theta_j} \right]. \tag{A.28}$$

In the following, notice how the `nlm` function assumes that the first argument of the function being minimized is a vector of arguments over which we should minimize, and any other arguments (in this case, the name of the data vector) can be specified by name in the `nlm` function call.

```
> gammasearch = nlm(gmll,c(momalpha,mombeta),hessian=T,datta=D); gammasearch
$minimum
[1] 142.0316

$estimate
[1] 1.805930 3.808674

$gradient
[1] 2.847002e-05 9.133932e-06

$hessian
         [,1]      [,2]
[1,] 36.68932 13.127271
[2,] 13.12727  6.222282

$code
[1] 1

$iterations
[1] 6

> eigen(gammasearch$hessian)$values
[1] 41.565137  1.346466
```

The `nlm` object `gammasearch` is a linked list. The item `minimum` is the value of the minus log likelihood function where the search stops. The item `estimate` is the point at which

the search stops, so $\widehat{\alpha} = 1.805930$ and $\widehat{\beta} = 3.808674$. The `gradient` is

$$\left(-\frac{\partial \ell}{\partial \alpha}, -\frac{\partial \ell}{\partial \beta}\right)^\top.$$

Besides being the direction of steepest decrease, it's something that should be zero at the MLE. And indeed it is, give or take a bit of numerical inaccuracy.

The Hessian at the stopping place is in `gammasearch$hessian`. The Hessian is the matrix of mixed partial derivatives defined by

$$\mathbf{H} = \left[\frac{\partial^2(-\ell)}{\partial \theta_i \partial \theta_j}\right]. \tag{A.29}$$

The rules about Hessian matrices are

- If the second derivatives are continuous, $\mathbf{H}$ is symmetric.
- If the gradient is zero at a point and $|\mathbf{H}| \neq 0$
  - If $\mathbf{H}$ is positive definite, there is a local minimum at the point.
  - If $\mathbf{H}$ is negative definite, there is a local maximum at the point.
  - If $\mathbf{H}$ has both positive and negative eigenvalues, the point is a saddle point.

The `eigen` command returns a linked list; one item is an array of the eigenvalues, and the other is the eigenvectors in the form of a matrix. Since for real symmetric matrices, positive definite is equivalent to all positive eigenvalues, it is convenient to check the eigenvalues to determine whether the numerical search has located a minimum. In this case it has. Finally, `code=1` means normal termination of the search, and `iterations=6` means the function took 6 steps downhill to reach its target.

It is very helpful to have the true parameter values $\alpha = 2$ and $\beta = 3$ for this example. $\widehat{\alpha} = 1.8$ seems pretty close, while and $\widehat{\beta} = 3.8$ seems farther off. This is a reminder of how informative confidence intervals and tests can be.

## A.6.5 The Invariance Principle

The Invariance Principle of maximum likelihood estimation says that *the MLE of a function is that function of the MLE.* An example comes first, followed by formal details.

**Example A.6.3** *Parameterizing in Terms of Odds Rather than Probability*

Let $D_1, \ldots, D_n$ be a random sample from a Bernoulli distribution (1=Yes, 0=No) with parameter $\theta, 0 < \theta < 1$. The parameter space is $\Theta = (0, 1)$, and the likelihood function is

$$L(\theta) = \prod_{i=1}^{n} \theta^{d_i}(1-\theta)^{1-d_i} = \theta^{\sum_{i=1}^n d_i}(1-\theta)^{n-\sum_{i=1}^n d_i}.$$

Differentiating the log likelihood with respect to $\theta$, setting the derivative to zero and solving yields the usual estimate $\widehat{\theta} = \overline{d}$, the sample proportion.

Now suppose that instead of the probability, we write this model in terms of the *odds* of $D_i = 1$, a re-parameterization that is often useful in categorical data analysis. Denote the odds by $\theta'$. The definition of odds is

$$\theta' = \frac{\theta}{1-\theta} = g(\theta). \tag{A.30}$$

As $\theta$ ranges from zero to one, $\theta'$ ranges from zero to infinity. So there is a new parameter space: $\theta' \in \Theta' = (0, \infty)$.

To write the likelihood function in terms of $\theta'$, first solve for $\theta$, obtaining

$$\theta = \frac{\theta'}{1+\theta'} = g^{-1}(\theta').$$

The likelihood in terms of $\theta'$ is then

$$\begin{aligned} L(g^{-1}(\theta')) &= \theta^{\sum_{i=1}^n d_i}(1-\theta)^{n-\sum_{i=1}^n d_i} \\ &= \left(\frac{\theta'}{1+\theta'}\right)^{\sum_{i=1}^n d_i} \left(1 - \frac{\theta'}{1+\theta'}\right)^{n-\sum_{i=1}^n d_i} \\ &= \left(\frac{\theta'}{1+\theta'}\right)^{\sum_{i=1}^n d_i} \left(\frac{1+\theta'-\theta'}{1+\theta'}\right)^{n-\sum_{i=1}^n d_i} \\ &= \frac{\theta'^{\sum_{i=1}^n d_i}}{(1+\theta')^n}. \end{aligned}$$

Note how re-parameterization changes the functional form of the likelihood function. The general formula is $L'(\theta') = L(g^{-1}(\theta'))$. For this example,

$$L'(\theta') = \frac{\theta'^{\sum_{i=1}^n d_i}}{(1+\theta')^n}. \tag{A.31}$$

At this point one could differentiate the log of (A.31) with respect to $\theta'$, set the derivative to zero, and solve for $\theta'$. The point of the invariance principle is that this is unnecessary. The maximum likelihood estimator of $g(\theta)$ is $g(\widehat{\theta})$, so one need only look at (A.30) and write

$$\widehat{\theta'} = \frac{\widehat{\theta}}{1-\widehat{\theta}} = \frac{\overline{d}}{1-\overline{d}}\ .$$

It is often convenient to parameterize a statistical model in more than one way. The invariance principle can save a lot of work in practice, because it says that you only have to maximize the likelihood function once. It is useful theoretically too.

In Example A.6.3, the likelihood function has only one maximum and the function $g$ linking $\theta'$ to $\theta'$ is one-to-one, which is why we can write $g^{-1}$. This is the situation where the invariance principle is clearest and most useful. Here is a proof.

Let the parameter $\theta \in \Theta$, and re-parameterize by $\theta' = g(\theta)$. The new parameter space is $\Theta' = \{\theta' : \theta' = g(\theta), \theta \in \Theta\}$. The function $g : \Theta \to \Theta'$ is one-to-one, meaning that there exists a function $g^{-1}$ such that $g^{-1}(g(\theta)) = \theta$ for all $\theta \in \Theta$. Suppose the likelihood function $L(\theta)$ has a unique maximum at $\widehat{\theta} \in \Theta$, so that for all $\theta \in \Theta$ with $\theta \neq \widehat{\theta}$, $L(\widehat{\theta}) > L(\theta)$. For every $\theta \in \Theta$,

$$L(\theta) = L(g^{-1}(g(\theta))) = L(g^{-1}(\theta')) = L'(\theta')$$

Maximizing $L'(\theta')$ over $\theta' \in \Theta'$ yields $\widehat{\theta'}$ satisfying $L'(\widehat{\theta'}) \geq L'(\theta')$ for all $\theta' \in \Theta'$. The invariance principle says $\widehat{\theta'} = g(\widehat{\theta})$.

Let $\theta_0 = g^{-1}(\widehat{\theta'})$ so that $g(\theta_0) = \widehat{\theta'}$. The objective is to show that this value $\theta_0 \in \Theta$ equals $\widehat{\theta}$. Suppose on the contrary that $\theta_0 \neq \widehat{\theta}$. Then because the maximum of $L(\theta)$ over $\Theta$ is unique, $L(\widehat{\theta}) > L(\theta_0)$. Therefore,

$$\begin{aligned} & L(g^{-1}(g(\widehat{\theta}))) > L(g^{-1}(g(\theta_0))) \\ \Rightarrow \quad & L'(g(\widehat{\theta})) > L'(g(\theta_0)) \\ \Rightarrow \quad & L'(g(\widehat{\theta})) > L'(\widehat{\theta'}). \end{aligned}$$

Since $g(\widehat{\theta}) \in \Theta'$, this contradicts $L'(\widehat{\theta'}) \geq L'(\theta')$ for all $\theta' \in \Theta'$, showing $\widehat{\theta} = \theta_0$. Not leaving anything to the imagination, we then have $g(\widehat{\theta}) = g(\theta_0) = \widehat{\theta'}$.

This concludes the proof, but it may be useful to establish the "obvious" fact that uniqueness of the maximum over $\Theta$ implies uniqueness of the maximum over $\Theta'$. If $\widehat{\theta'_1}$ and $\widehat{\theta'_2}$ are two points in $\Theta'$ with $L'(\widehat{\theta'_1}) \geq L'(\theta')$ and $L'(\widehat{\theta'_2}) \geq L'(\theta')$ for all $\theta' \in \Theta'$, the preceding argument shows that $g(\widehat{\theta}) = \widehat{\theta'_1}$ and $g(\widehat{\theta}) = \widehat{\theta'_2}$. Because function values are unique, this can only happen if $\widehat{\theta'_1} = \widehat{\theta'_2}$

### Exercises A.6.4

A.6.1) For each of the following distributions, derive a general expression for the Maximum Likelihood Estimator (MLE). Carry out the second derivative test to make sure you have a maximum. (What is the relationship of this to the Hessian?) Then use the data to calculate a numerical estimate.

(a) $p(x) = \theta(1-\theta)^x$ for $x = 0, 1, \ldots$, where $0 < \theta < 1$. Data: `4, 0, 1, 0, 1, 3, 2, 16, 3, 0, 4, 3, 6, 16, 0, 0, 1, 1, 6, 10`. Answer: 0.2061856

(b) $f(x) = \frac{\alpha}{x^{\alpha+1}}$ for $x > 1$, where $\alpha > 0$. Data: `1.37, 2.89, 1.52, 1.77, 1.04, 2.71, 1.19, 1.13, 15.66, 1.43` Answer: 1.469102

(c) $f(x) = \frac{\tau}{\sqrt{2\pi}} e^{-\frac{\tau^2 x^2}{2}}$, for $x$ real, where $\tau > 0$. Data: `1.45, 0.47, -3.33, 0.82, -1.59, -0.37, -1.56, -0.20` Answer: 0.6451059

(d) $f(x) = \frac{1}{\theta} e^{-x/\theta}$ for $x > 0$, where $\theta > 0$. Data: `0.28, 1.72, 0.08, 1.22, 1.86, 0.62, 2.44, 2.48, 2.96` Answer: 1.517778

A.6.2) The univariate normal density is

$$f(y|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(y-\mu)^2}{\sigma^2}}$$

(a) Show that the univariate normal likelihood may be written

$$L(\mu,\sigma^2) = (2\pi\sigma^2)^{-n/2} \exp -\frac{n}{2\sigma^2}\left\{\widehat{\sigma}^2 + (\overline{y}-\mu)^2\right\},$$

where $\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2$. Hint: Add and subtract $\overline{y}$.

(b) How does this expression allow you to see *without differentiating* that the MLE of $\mu$ is $\overline{y}$?

A.6.3) Let $X_1, \ldots, X_5$ be a random sample from a Gamma distribution with parameters $\alpha > 0$ and $\beta = 1$. That is, the density is

$$f(x;\alpha) = \frac{1}{\Gamma(\alpha)} e^{-x} x^{\alpha-1}$$

for $x > 0$, and zero otherwise.

The five data values are 2.06, 1.08, 0.96, 1.32, 1.53. Find an approximate numerical value of the maximum likelihood estimate of $\alpha$. Your final answer is one number. For this question you will hand in a one-page printout. On the back, you will write a brief explanation of what you did.

A.6.4) For each of the following distributions, try to derive a general expression for the Maximum Likelihood Estimator (MLE). Then, use R's `nlm` function to obtain the MLE numerically for the data supplied for the problem. The data are in a separate HTML document, because it saves a lot of effort to copy and paste rather than typing the data in by hand, and PDF documents can contain invisible characters that mess things up. NOTE! Put them here as well as in assignment HTML document.

(a) $f(x) = \frac{1}{\pi[1+(x-\theta)^2]}$ for $x$ real, where $-\infty < \theta < \infty$.

```
-3.77  -3.57 4.10 4.87  -4.18  -4.59  -5.27  -8.33 5.55  -4.35  -0.55 5.57
-34.78 5.05 2.18 4.12  -3.24 3.78  -3.57 4.86
```

For this one, try at least two different starting values and *plot the minus log likelihood function!*

(b) $f(x) = \frac{1}{2}e^{-|x-\theta|}$ for $x$ real, where $-\infty < \theta < \infty$.

```
3.36 0.90 2.10 1.81 1.62 0.16 2.01 3.35 4.75 4.27 2.04
```

(c) $f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$ for $0 < x < 1$, where $\alpha > 0$ and $\beta > 0$.

```
0.45 0.42 0.38 0.26 0.43 0.24 0.32 0.50 0.44 0.29 0.45 0.29 0.29 0.32 0.30
0.32 0.30 0.38 0.43 0.35 0.32 0.33 0.29 0.20 0.46 0.31 0.35 0.27 0.29 0.46
0.43 0.37 0.32 0.28 0.20 0.26 0.39 0.35 0.35 0.24 0.36 0.28 0.32 0.23 0.25
0.43 0.30 0.43 0.33 0.37
```

If you are getting a lot of warnings, maybe it's because the numerical search is leaving the parameter space. If so and if you are using R, try `help(nlminb)`.

For each distribution, be able to state (briefly) why differentiating the log likelihood and setting the derivative to zero does not work. For the computer part, bring to the quiz one sheet of printed output for each of the 3 distributions. The three sheets should be separate, because you may hand only one of them in. Each printed page should show the following, *in this order*.

- Definition of the function that computes the likelihood, or log likelihood, or minus log likelihood or whatever.
- How you got the data into R – probably a `scan` statement.
- Listing of the data for the problem.
- The `nlm` statement and resulting output.

A.6.5) Let $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{X}$ is an $n \times p$ matrix of known constants, $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown constants, and $\boldsymbol{\epsilon}$ is multivariate normal with mean zero and covariance matrix $\sigma^2\mathbf{I}_n$, with $\sigma^2 > 0$ an unknown constant.

(a) What is the distribution of $\mathbf{Y}$? There is no need to show any work.

(b) Assuming that the columns of $\mathbf{X}$ are linearly independent, show that the maximum likelihood estimate of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top Y$. Don't use derivatives. The trick is to add and subtract $\hat{\boldsymbol{\beta}}$, distribute the expected value, and simplify. Does your answer apply for any value of $\sigma^2$? Why or why not?

(c) Given the MLE of $\boldsymbol{\beta}$, find the MLE of $\sigma^2$. Show your work. This time you may differentiate.

### A.6.6 Interval Estimation and Testing

All the tests and confidence intervals here are based on large-sample approximations, primarily the Central Limit Theorem. See Section A.5 for basic definitions and results. They are valid as the sample size $n \to \infty$, but frequently perform well for samples that are only fairly large. How big is big enough? This is a legitimate question, and the honest answer is that it depends upon the distribution of the data. In practice, people often just apply these tools almost regardless of the sample size, because nothing better is available. Some do it with their eyes closed, some squint, and some have their eyes wide open.

The basic result comes from the research of Abraham Wald (give a source) in the 1950s. *As the sample size n increases, the distribution of the maximum likelihood estimator* $\widehat{\boldsymbol{\theta}}_n$ *approaches a multivariate normal* with expected value $\boldsymbol{\theta}$ and variance-covariance matrix $\mathbf{V}_n(\boldsymbol{\theta})$. It is quite remarkable that anyone could figure this out, given that it includes cases like the Gamma, where no closed-form expressions for the maximum likelihood estimators are possible. The theorem in question is not true for every distribution, but it is true if the distribution of the data is not too strange. The precise meaning of "not too

strange" is captured in a set of technical conditions called *regularity conditions*. Volume 2 of *Kendall's advanced theory of statistics* [63] is a good textbook source for the details.

If $\boldsymbol{\theta}$ is a $k \times 1$ matrix, then $\mathbf{V}_n(\boldsymbol{\theta})$ is a $k \times k$ matrix, called the *asymptotic covariance matrix* of the estimators. It's not too surprising that it depends on the parameter $\boldsymbol{\theta}$, and it also depends on the sample size $n$. Using the asymptotic covariance matrix, it is possible to construct a variety of useful tests and confidence intervals.

### Fisher Information

The fact that $\mathbf{V}_n(\boldsymbol{\theta})$ depends on the unknown parameter will present no problem; substituting $\widehat{\boldsymbol{\theta}}_n$ for $\boldsymbol{\theta}$ yields an *estimated* asymptotic covariance matrix. So consider the form of the matrix $\mathbf{V}$.

Think of a one-parameter maximum likelihood problem, where we differentiate the log likelihood, set the derivative to zero and solve for $\theta$; the solution is $\widehat{\theta}$. The log likelihood will be concave down at $\widehat{\theta}$, but the exact way it looks will depend on the distribution as well as the sample size. In particular, it could be almost flat at $\widehat{\theta}$, or it could be nearly a sharp peak, with extreme downward curvature. In the latter case, clearly the log likelihood is more informative about $\theta$. It contains more information. One of the many good ideas of R. A. F. Fisher was that the second derivative reflects curvature, and and can be viewed as a measure of the information provided by the sample data. It is called the *Fisher Information* in his honour.

Now with increasing sample size, nearly all log likelihood functions acquire more and more downward curvature at the MLE. This makes sense – more data provide more information. But how about the information from just one observation? If you look at the second derivative of the log likelihood function,

$$\frac{\partial^2 \ell}{\partial \theta^2} = \frac{\partial^2}{\partial \theta^2} \ln \prod_{i=1}^{n} f(d_i;\theta) = \sum_{i=1}^{n} \frac{\partial^2}{\partial \theta^2} \ln f(d_i;\theta),$$

you see that it is the sum of $n$ quantities. Each observation is contributing a piece to the downward curvature. But how much? Well, it depends on the particular data value $x_i$. But the data are a random sample, so in fact the contribution is a random quantity: $\frac{\partial^2}{\partial \theta^2} \ln f(X_i;\theta)$. How about the information one would *expect* an observation to contribute? Okay, take the expected value. Finally, note that because the curvature is down at the MLE, the quantity we are discussing is negative. But we want to call this "information," and it would be nicer if it were a positive number, so higher values meant more information. Okay, multiply by $-1$. This leads to the definition of the Fisher Information in a single observation:

$$I(\theta) = E\left[-\frac{\partial^2}{\partial \theta^2} \ln f(D_i;\theta)\right]. \tag{A.32}$$

The information is the same for $i = 1, \ldots, n$, and the Fisher Information in the entire sample is just $nI(\theta)$.

It was clear that Fisher was onto something good, because for many problems where the variance of $\widehat{\theta}$ can be calculated exactly, it is one divided by the Fisher Information. Subsequently Cramér and Rao discovered the *Cramér-Rao Inequality*, which says that for *any* statistic $T$ that is an unbiased estimator of $\theta$,

$$Var(T) \geq \frac{1}{nI(\theta)}.$$

That's impressive, because to have a small variance is a great property in an estimator; it means precise estimation. The Cramér-Rao inequality tells us that in terms of variance, one cannot do better than an unbiased estimator whose variance equals hte reciprocal (inverse) of the Fisher Information, and many MLEs do that. Subsequently, Wald[15] showed that under some regularity conditions, the variances of maximum likelihood estimators in general attain the Cramér-Rao lower bound as $n \to \infty$. Thus, to learn the asymptotic variance of $\widehat{\theta}$, you do not need an explicit formula for $\widehat{\theta}$. All you need is the Fisher Information. Also, in terms of variance nothing can beat maximum likelihood estimation, at least for large samples. So if the distribution of the data is known so you can write down the likelihood, it is difficult to justify any method of estimation other than maximum likelihood.

Calculating the expected value in (A.32) is often not too hard because taking the log and differentiating twice results in some simplification; it's a source of many fun homework problems. But still it can be a chore, especially for multiparameter problems, which will be taken up shortly. For larger sample sizes, the Law of Large Numbers (Section A.5) guarantees that the expected value can be approximated quite well by a sample mean, so that

$$I(\theta) = E\left[-\frac{\partial^2}{\partial\theta^2}\ln f(D_1;\theta)\right] \approx \frac{1}{n}\sum_{i=1}^{n} -\frac{\partial^2}{\partial\theta^2}\ln f(D_i;\theta).$$

This is sometimes called the *observed* Fisher Information.

Multiplying the observed Fisher Information by $n$ to get the approximate information in the entire sample yields

$$\sum_{i=1}^{n} -\frac{\partial^2}{\partial\theta^2}\ln f(D_i;\theta) = \frac{\partial^2}{\partial\theta^2}\sum_{i=1}^{n} -\ln f(D_i;\theta) = \frac{\partial^2}{\partial\theta^2}\left(-\ln\prod_{i=1}^{n} f(D_i;\theta)\right).$$

That's just the second derivative of the minus log likelihood.

The parameter $\theta$ is unknown, so to get the *estimated* Fisher Information in the whole sample, substitute $\widehat{\theta}$. The result is

$$\frac{\partial^2}{\partial\theta^2}\left(-\ln\prod_{i=1}^{n} f(D_i;\widehat{\theta})\right).$$

That's the second derivative of minus the log likelihood, evaluated at the maximum likelihood estimate. And, it's a function of the sample data that is not a function of any

[15] Need a reference

unknown parameters; in other words it is a statistic. If you have already carried out the second derivative test to check that you really had a maximum, all you need to do to estimate the variance of $\widehat{\theta}$ is take the reciprocal of the second derivative and multiply by $-1$. It is truly remarkable how neatly this all works out.

Generalization to the multivariate case is very natural. Now the parameter is $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)^\top$ and the Fisher Information *Matrix* is a $k \times k$ matrix of (expected) mixed partial derivatives, defined by

$$\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}) = \left[ -E\left( \frac{\partial^2}{\partial\theta_i \partial\theta_j} f(\mathbf{D}_1; \boldsymbol{\theta}) \right) \right], \tag{A.33}$$

where the boldface $\mathbf{D}_i$ is an acknowledgement that the data might also be multivariate.

To estimate the Fisher information matrix, one may simply put a hat on $\boldsymbol{\theta}$ in A.33. If calculating the expected values is too much of a pain, one may replace the expected value by a sample mean as well as replacing $\boldsymbol{\theta}$ with $\widehat{\boldsymbol{\theta}}$. The result is

$$\boldsymbol{\mathcal{J}}(\widehat{\boldsymbol{\theta}}) = \left[ \frac{\partial^2}{\partial\theta_i \partial\theta_j} \left( -\ln \prod_{q=1}^{n} f(\mathbf{D}_q; \widehat{\boldsymbol{\theta}}) \right) \right] = \left[ \frac{\partial^2}{\partial\theta_i \partial\theta_j} \left( -\ell(\widehat{\boldsymbol{\theta}}) \right) \right]. \tag{A.34}$$

$\boldsymbol{\mathcal{I}}(\widehat{\boldsymbol{\theta}})$ is sometimes loosely called the "expected" Fisher information, and $\boldsymbol{\mathcal{J}}(\widehat{\boldsymbol{\theta}})$ is sometimes called the "observed" Fisher information, even though it would be more accurate to call it the estimated observed Fisher information. They are both excellent large-sample estimates of $\boldsymbol{\mathcal{I}}(\boldsymbol{\theta})$ in (A.33).

In the one-dimensional case, one divided by the estimated Fisher Information is the (estimated) asymptotic variance of the maximum likelihood estimator. *In the multiparameter case, the estimated Fisher Information is a matrix, and the corresponding estimated asymptotic variance-covariance matrix is its inverse.* Assume that the true Fisher information matrix is being estimated by $\boldsymbol{\mathcal{J}}(\widehat{\boldsymbol{\theta}})$, and denote the estimated asymptotic covariance matrix by $\widehat{\mathbf{V}}_n$. In that case we have

$$\widehat{\mathbf{V}}_n = \boldsymbol{\mathcal{J}}(\widehat{\boldsymbol{\theta}}_n)^{-1}. \tag{A.35}$$

Now comes the really good part. Comparing Formula (A.34) for the Fisher Information to Formula (A.29) for the Hessian, we see that they are exactly the same. And *the Hessian evaluated at $\widehat{\boldsymbol{\theta}}$ is a by-product of the numerical search for the MLE*[16].

So to get a good estimate of the asymptotic covariance matrix, minimize minus the log likelihood, tell the software to give you the Hessian, and calculate its inverse by computer. The theoretical story may be a bit long here, but what you have to do in practice is quite simple.

Continuing with the Gamma distribution Example A.6.2, the Hessian is

[16] At least for generic numerical minimization routines like `R`'s `nlm`. Some specialized methods like iterative proportional fitting of log-linear models and Fisher scoring (iteratively re-weighted least squares) for generalized linear models maximize the likelihood indirectly and do not require calculation of the Hessian.

```
> gammasearch$hessian
         [,1]      [,2]
[1,] 36.68932 13.127271
[2,] 13.12727  6.222282
```

and the asymptotic covariance is just

```
> Vhat = solve(gammasearch$hessian); V
           [,1]       [,2]
[1,]  0.1111796 -0.2345577
[2,] -0.2345577  0.6555638 .
```

The diagonal elements of $\widehat{\mathbf{V}}$ are the estimated variances of the sampling distributions of $\widehat{\alpha}$ and $\widehat{\beta}$ respectively, and their square roots are the standard errors.

```
> SEalphahat = sqrt(Vhat[1,1]); SEbetahat = sqrt(Vhat[2,2])
```

In general, let $\theta$ denote an element of the parameter vector, let $\widehat{\theta}$ be its maximum likelihood estimator, and let the standard error of $\widehat{\theta}$ be written $S_{\widehat{\theta}}$. Then Wald's Central Limit Theorem for maximum likelihood estimators tells us that

$$Z = \frac{\widehat{\theta} - \theta}{S_{\widehat{\theta}}} \tag{A.36}$$

has an approximate standard normal distribution. In particular, for the Gamma example

$$Z_1 = \frac{\widehat{\alpha} - \alpha}{S_{\widehat{\alpha}}} \quad \text{and} \quad Z_2 = \frac{\widehat{\beta} - \beta}{S_{\widehat{\beta}}}$$

may be treated as standard normal.

### Confidence Intervals

These quantities may be used to produce both tests and confidence intervals. For example, a 95% confidence interval for the parameter $\theta$ is obtained as follows.

$$\begin{aligned} 0.95 &\approx Pr\{-1.96 \leq Z \leq 1.96\} \\ &= Pr\left\{-1.96 \leq \frac{\widehat{\theta} - \theta}{S_{\widehat{\theta}}} \leq 1.96\right\} \\ &= Pr\left\{\widehat{\theta} - 1.96\, S_{\widehat{\theta}} \leq \theta \leq \widehat{\theta} + 1.96\, S_{\widehat{\theta}}\right\} \end{aligned}$$

This could also be written $\widehat{\theta} \pm 1.96\, S_{\widehat{\theta}}$ .

If you are used to seeing confidence intervals with a $\sqrt{n}$ and wondering where it went, recall that $S_{\overline{X}} = \frac{S}{\sqrt{n}}$. The $\sqrt{n}$ is also present in the confidence interval for $\theta$, but it is embedded in $S_{\widehat{\theta}}$.

Here are the 95% confidence intervals for the Gamma distribution example:

```
> alphahat = gammasearch$estimate[1]; betahat = gammasearch$estimate[2]
> Lalpha = alphahat - 1.96*SEalphahat; Ualpha = alphahat + 1.96*SEalphahat
> Lbeta = betahat - 1.96*SEbetahat; Ubeta = betahat + 1.96*SEbetahat
> cat("\nEstimated alpha = ",round(alphahat,2),"  95 percent CI from ",
+     round(Lalpha,2)," to ",round(Ualpha,2), "\n\n")

Estimated alpha =  1.81   95 percent CI from  1.15  to  2.46

> cat("\nEstimated beta = ",round(betahat,2),"  95 percent CI from ",
+     round(Lbeta,2)," to ",round(Ubeta,2), "\n\n")

Estimated beta =  3.81   95 percent CI from  2.22  to  5.4
```

Notice that while the parameter estimates may not seem very accurate, the 95% confidence intervals do include the true parameter values $\alpha = 2$ and $\beta = 3$.

### $Z$-tests

The standard normal variable in (A.36) can be used to form a $Z$-test of $H_0 : \theta = \theta_0$ using

$$Z = \frac{\widehat{\theta} - \theta_0}{S_{\widehat{\theta}}}.$$

So for example, suppose the data represent time intervals between events occurring in time, and we wonder whether the events arise from a Poisson process. In this case the distribution of times would be exponential, which means $\alpha = 1$. To test this null hypothesis at the 0.05 level,

```
> Z = (alphahat-1)/SEalphahat; Z
[1] 2.417046
> pval = 2*(1-pnorm(abs(Z))); pval # Two-sided test
[1] 0.01564705
```

So, the null hypothesis is rejected, and because the value is positive, the conclusion is that the true value of $\alpha$ is greater than one[17].

[17]The following basic question arises from time to time. Suppose a null hypothesis is rejected in favour of a two-sided alternative. Are we then "allowed" to look at the sign of the test statistic and conclude that $\theta < \theta_0$ or $\theta > \theta_0$, or must we just be content with saying $\theta \neq \theta_0$? The answer is that directional conclusions are theoretically justified as well as practically desirable. Think of splitting up the two-sided level $\alpha$ test (call it the *overall test*) into two one-sided tests with significance level $\alpha/2$. The null hypotheses of these tests are $H_{0,a} : \theta \leq \theta_0$ and $H_{0,b} : \theta \geq \theta_0$. Exactly one of these null hypotheses will be rejected if and only if the null hypothesis of the overall test is rejected, so the set of two one-sided tests is fully equivalent to the overall two-sided test. And directional conclusions from the one-sided tests are clearly justified.

On a deeper level, notice that the null hypothesis of the overall test is the intersection of the null hypotheses of the one-sided tests, and its critical region (rejection region) is the union of the critical

When statistical software packages display this kind of large-sample $Z$-test, they usually just divide $\widehat{\theta}$ by its standard error, testing the null hypothesis $H_0 : \theta = 0$. For parameters like regression coefficients, this is usually a good generic choice.

### A.6.7 Wald Tests

The approximate multivariate normality of the MLE can be used to construct a larger class of hypothesis tests for *linear* null hypotheses. A linear null hypothesis sets a collection of linear combinations of the parameters to zero. Suppose $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)^\top$ is a $k \times 1$ vector. A linear null hypothesis can be written

$$H_0 : \mathbf{L}\boldsymbol{\theta} = \mathbf{h},$$

where $\mathbf{L}$ is an $r \times k$ matrix of constants, with rank $r$, $r \leq k$. As an example let $\boldsymbol{\theta} = (\theta_1, \ldots \theta_7)^\top$, and the null hypothesis is

$$\theta_1 = \theta_2, \quad \theta_6 = \theta_7, \quad \frac{1}{3}(\theta_1 + \theta_2 + \theta_3) = \frac{1}{3}(\theta_4 + \theta_5 + \theta_6)\,.$$

This may be expressed in the form $\mathbf{L}\boldsymbol{\theta} = \mathbf{h}$ as follows:

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Recall from Section A.4 of this appendix that if $\mathbf{X} \sim N_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and $\mathbf{L}$ is an $r \times k$ constant matrix of rank $r$, then

$$\mathbf{C}\mathbf{X} \sim N_r(\mathbf{L}\boldsymbol{\mu}, \mathbf{L}\boldsymbol{\Sigma}\mathbf{L}^\top)$$

and

$$(\mathbf{C}\mathbf{X} - \mathbf{L}\boldsymbol{\mu})^\top (\mathbf{L}\boldsymbol{\Sigma}\mathbf{C}^\top)^{-1} (\mathbf{L}\mathbf{X} - \mathbf{L}\boldsymbol{\mu}) \sim \chi^2(r).$$

Similar facts hold asymptotically — that is approximately, as the sample size $n$ approaches infinity. Because (approximately) $\widehat{\boldsymbol{\theta}}_n \sim N_k(\boldsymbol{\theta}, \widehat{\mathbf{V}}_n)$,

$$\mathbf{L}\widehat{\boldsymbol{\theta}}_n \sim N_r(\mathbf{L}\boldsymbol{\theta}, \mathbf{L}\widehat{\mathbf{V}}_n\mathbf{L}^\top)$$

regions of the one-sided tests. This makes the two one-sided tests a set of *union-intersection multiple comparisons*, which are always simultaneously protected against Type I error at the significance level of the overall test. Performing the two-sided test and then following up with a one-sided test is very much like following up a statistically significant ANOVA with Scheffeé tests. Indeed, Scheffé tests are another example of union-intersection multiple comparisons. See [30] for details.

and

$$(\mathbf{L}\widehat{\boldsymbol{\theta}}_n - \mathbf{L}\boldsymbol{\theta})^\top (\mathbf{C}\widehat{\mathbf{V}}_n\mathbf{L}^\top)^{-1}(\mathbf{L}\widehat{\boldsymbol{\theta}}_n - \mathbf{L}\boldsymbol{\theta}) \sim \chi^2(r).$$

So, if $H_0 : \mathbf{L}\boldsymbol{\theta} = \mathbf{h}$ is true, we have the Wald test statistic

$$W_n = (\mathbf{L}\widehat{\boldsymbol{\theta}}_n - \mathbf{h})^\top (\mathbf{C}\widehat{\mathbf{V}}_n\mathbf{L}^\top)^{-1}(\mathbf{L}\widehat{\boldsymbol{\theta}}_n - \mathbf{h}) \sim \chi^2(r), \tag{A.37}$$

where again,

$$\widehat{\mathbf{V}}_n = \boldsymbol{\mathcal{J}}(\widehat{\boldsymbol{\theta}})^{-1} = \left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\left(-\ell(\widehat{\boldsymbol{\theta}})\right)\right]^{-1}.$$

Here is a test of $H_0 : \alpha = \beta$ for the Gamma distribution example. A little care must be taken to ensure that the matrices in (A.37) are the right size.

```
> #  H0: C theta = 0 is that alpha = beta <=> alpha-beta=0
> # Name C is used by R
> CC = rbind(c(1,-1)); is.matrix(CC); dim(CC)
[1] TRUE
[1] 1 2
> thetahat = as.matrix(c(alphahat,betahat)); dim(thetahat)
[1] 2 1
> W = t(CC%*%thetahat) %*% solve(CC%*%Vhat%*%t(CC)) %*% CC%*%thetahat
> W = as.numeric(W) # it was a 1x1 matrix
> pval2 = 1-pchisq(W,1)
> cat("Wald Test:  W = ", W, ", p = ", pval2, "\n")
Wald Test:  W =  3.245501 , p =  0.07161978
```

We might as well define a function to do Wald tests in general. The function returns a pair of quantities, the Wald test statistic and the $p$-value.

```
> WaldTest = function(C,thetahat,h=0) # H0: C theta = h
+     {
+     WaldTest = numeric(2)
+     names(WaldTest) = c("W","p-value")
+     dfree = dim(C)[1]
+     W = t(C%*%thetahat-h) %*% solve(C%*%Vhat%*%t(C)) %*% (C%*%thetahat-h)
+     W = as.numeric(W)
+     pval = 1-pchisq(W,dfree)
+     WaldTest[1] = W; WaldTest[2] = pval
+     WaldTest
+     } # End function WaldTest
```

Here is the same test of $H_0 : \alpha = \beta$ done immediately above, just to test out the function. Notice that the default value of $\mathbf{h}$ in $H_0 : \mathbf{L}\boldsymbol{\theta} = \mathbf{h}$ is zero, so it does not have to be specified. The matrix `CC` has already been created, and the computed values are the same as before, naturally.

```
> WaldTest(CC,as.matrix(c(alphahat,betahat)))
         W    p-value
3.24550127 0.07161978
```

Here is a test of $H_0 : \alpha = 2, \beta = 3$, which happen to be the true parameter values. The null hypothesis is not rejected.

```
> C2 = rbind(c(1,0),
+           c(0,1) )
> WaldTest(C2,as.matrix(c(alphahat,betahat)),c(2,3))
        W   p-value
1.3305497 0.5141322
```

Finally, here is a test of $H_0 : \alpha = 1$, which was done earlier with a $Z$-test.

```
> WaldTest(t(c(1,0)),as.matrix(c(alphahat,betahat)),1)
         W    p-value
5.84210645 0.01564708
> Z; pval
[1] 2.417045
[1] 0.01564708
> Z^2
[1] 5.842106
```

The results of the Wald and $Z$ tests are identical, with $W_n = Z^2$. In general, suppose the matrix $\mathbf{L}$ in $H_0 : \mathbf{L}\boldsymbol{\theta} = \mathbf{h}$ has just a single row, and that row contains one 1 in position $j$ and all the rest zeros. Take a look at Formula (A.37) for the Wald test statistic. Pre-multiplying by $\mathbf{L}$ in $\mathbf{C}\widehat{\mathbf{V}}_n$ picks out row $j$ of $\widehat{\mathbf{V}}_n$, and post-multiplying by $\mathbf{L}^\top$ picks out column $j$ of the result, so that $\mathbf{C}\widehat{\mathbf{V}}_n\mathbf{L}^\top = \widehat{v}_{j,j}$, and inverting it puts it in the denominator. In the numerator, $(\mathbf{L}\widehat{\boldsymbol{\theta}}_n - \mathbf{h})^\top(\mathbf{L}\widehat{\boldsymbol{\theta}}_n - \mathbf{h}) = (\widehat{\theta}_j - \theta_{j,0})^2$, so that $W_n = Z^2$. Thus, squaring a large-sample $Z$-test gives a Wald chisquare test with one degree of freedom.

### A.6.8 Likelihood Ratio Tests

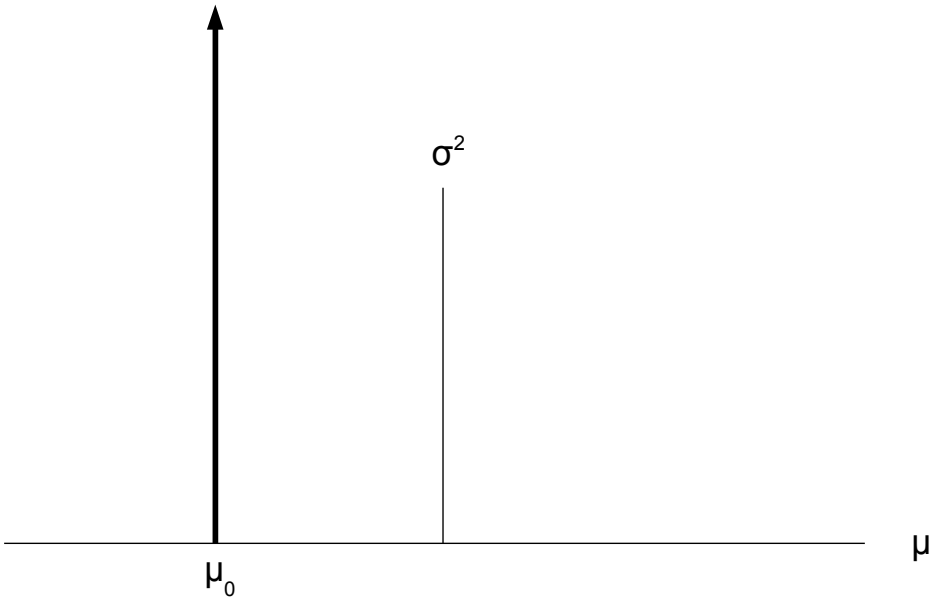
Likelihood ratio tests fall into two categories, exact and large-sample. The main examples of exact likelihood ratio tests include are the standard $F$-tests and $t$-tests associated with regression and the analysis of variance for normal data. Here, we concentrate on the large-sample likelihood ratio tests.

Consider the following hypothesis-testing framework. The data are $D_1, \ldots, D_n$. The distribution of these independent and identically distributed random variables depends on the parameter $\theta$, and we are testing a null hypothesis $H_0$.

$$
\begin{array}{l}
D_1, \ldots, D_n \overset{i.i.d.}{\sim} P_\theta,\ \theta \in \Theta, \\
H_0 : \theta \in \Theta_0 \text{ v.s. } H_A : \theta \in \Theta \cap \Theta_0^c,
\end{array}
$$

For example, let $D_1, \ldots, D_n \overset{i.i.d.}{\sim} N(\mu, \sigma^2)$. The null hypothesis is $H_0 : \mu = \mu_0$ v.s. versus $H_A : \mu \neq \mu_0$. The full parameter space is $\Theta = \{(\mu, \sigma^2) : -\infty < \mu < \infty, \sigma^2 > 0\}$ and the restricted parameter space is $\Theta_0 = \{(\mu, \sigma^2) : \mu = \mu_0, \sigma^2 > 0\}$. The full and restricted parameter spaces are shown in Figure A.2.

Figure A.2: Full versus reduced parameter spaces for $H_0 : \mu = \mu_0$ versus $H_A : \mu \neq \mu_0$




In general, the data have likelihood function

$$L(\theta) = \prod_{i=1}^{n} f(d_i; \theta),$$

where $f(d_i; \theta)$ is the density or probability mass function evaluated at $d_i$. Let $\widehat{\theta}$ denote the usual Maximum Likelihood Estimate (MLE). That is, it is the parameter value for which the likelihood function is greatest, over all $\theta \in \Theta$. Let $\widehat{\theta}_0$ denote the *restricted* MLE. The restricted MLE is the parameter value for which the likelihood function is greatest, over all $\theta \in \Theta_0$. This MLE is *restricted* by the null hypothesis $H_0 : \theta \in \Theta_0$. It should be clear that $L(\widehat{\theta}_0) \leq L(\widehat{\theta})$, so that the *likelihood ratio*.

$$\lambda = \frac{L(\widehat{\theta}_0)}{L(\widehat{\theta})} \leq 1.$$

The likelihood ratio will equal one if and only if the overall MLE $\widehat{\theta}$ is located in $\Theta_0$. In this case, there is no reason to reject the null hypothesis.

Suppose that the likelihood ratio is strictly less than one. If it's a *lot* less than one, then the data are a lot less likely to have been observed under the null hypothesis than

under the alternative hypothesis, and the null hypothesis is questionable. This is the basis of the likelihood ratio tests.

If $\lambda$ is small (close to zero), then $\ln(\lambda)$ is a large negative number, and $-2\ln\lambda$ is a large positive number.

Tests will be based on

$$\begin{aligned} G^2 &= -2\ln\left(\frac{\max_{\theta\in\Theta_0} L(\theta)}{\max_{\theta\in\Theta} L(\theta)}\right) \\ &= -2\ln\left(\frac{L(\widehat{\theta}_0)}{L(\widehat{\theta})}\right) \\ &= -2\ln L(\widehat{\theta}_0) - [-2\ln L(\widehat{\theta})] \\ &= 2\left(-\ell(\widehat{\theta}_0) - [-\ell(\widehat{\theta})]\right). \end{aligned} \tag{A.38}$$

Thus, the test statistic $G^2$ is the *difference* between two $-2$ log likelihood functions. This means that to carry out a test, you can minimize $-\ell(\theta)$ twice, first over all $\theta \in \Theta$, and then over all $\theta \in \Theta_0$. The test statistic is the difference between the two minimum values, multiplied by two.

If the null hypothesis is true, then the test statistic $G$ has, if the sample size is large, an approximate chisquare distribution, with degrees of freedom equal to the difference of the *dimension* of $\Theta$ and $\Theta_0$. For example, if the null hypothesis is that 4 elements of $\theta$ equal zero, then the degrees of freedom are equal to 4. If the null hypothesis imposes $r$ linearly independent linear restrictions on $\theta$ (as in $H_0 : \mathbf{L}\boldsymbol{\theta} = \mathbf{h}$), then the degrees of freedom equal $r$, the number or rows in $\mathbf{L}$. Another way to obtain the degrees of freedom is by counting the equal signs in the null hypothesis.

The $p$-value associated with the test statistic $G^2$ is $Pr\{X > G^2\}$, where $X$ is a chisquare random variable with $r$ degrees of freedom. If $p < \alpha$, we reject $H_0$ and call the results "statistically significant." The standard choice is $\alpha = 0.05$.

Many null hypotheses are linear statements of the form $H_0 : \mathbf{L}\boldsymbol{\theta} = \mathbf{h}$, but some are not.

**Example A.6.4** *A Non-linear Null Hypothesis*

Suppose you wanted to test $H_0 : \sigma^2 = \mu^2$ based on a normal random sample. The restricted MLE is fairly easy to find numerically (see Example A.6.1), and it seems like the degrees of freedom should equal one because the null hypothesis has one equals sign. Can this be justified formally?

The original proof published in 1938 by Wilks [70] applies to linear null hypotheses, and if you look at high-level textbooks like the *Advanced Theory of Statistics* [63], you will find only Wilks' proof, without modification. A way around this that often works is to use the Invariance Principle of Section A.6.5. Suppose the null hypothesis is that one or more non-linear functions of $\theta$ equal zero. If you can, make those functions part of a function that is one-to-one, and then re-parameterize. Your null hypothesis is now a

linear null hypothesis in the new paraameter space. Wilks' theorem applies, and you are done. Furthermore, you don't have to literally re-parameterize. A glance at the proof of the Invariance Principle confirms that the likelihood ratio test statistic is the same under the original and re-parameterized models. Thus, the degrees of freedon equals he number of equals signs in the null hypothesis, period.

For Example A.6.4, let $\theta'_1 = \sigma^2 - \mu^2$ and $\theta'_2 = \mu$. The function is one-to-one, because $\mu = \theta'_2$ and $\sigma^2 = \theta'_1 + \theta'^2_2$. The null hypothesis is $H_0 : \theta'_1 = 0$. That's is a linear null hypothesis, so by Wilks' Theorem, the test statistic has a chi-squared distribution with $df = 1$.

Sometimes this lovely trick does not work. In a regression, it is easy to test the null hypothesis that $\beta_1$ and $\beta_2$ are both zero; this is a linear null hypothesis. But suppose that you want to test the null hypothesis that $\beta_1$ *or* $\beta_2$ (or maybe both) are equal to zero. This is reasonable and attractive, because the alternative is that they are both non-zero, and it would be nice to have a single test for this. The null hypothesis is $H_0 : \beta_1\beta_2 = 0$, which is non-linear. Furthermore, any function that yields $\theta'_1 = \beta_1\beta_2 = 0$ can't be one-to-one, because recovering $\beta_1$ or $\beta_2$ would potentially involve dividing by zero. Thus, while it would be perfectly possible to obtain the restricted MLE $\widehat{\theta}_0$ numerically and calculate the likelihood ratio statistic, its distribution under the null hypothesis is mysterious (to me, anyway). So, transforming a non-linear null hypothesis into a linear one by a one-to-one re-parameterization is a method that often works, but not always.

To illustrate the likelihood ratio tests, consider (one last time) the Gamma distribution Example A.6.2. For comparison, the likelihood ratio method will be used test the same three null hypotheses that were tested earlier using Wald tests. They are

- $H_0 : \alpha = 1$
- $H_0 : \alpha = \beta$
- $H_0 : \alpha = 2, \beta = 3$

For $H_0 : \alpha = 1$, the restricted parameter space is $\Theta_0 = \{(\alpha, \beta) : \alpha = 1, \beta > 0\}$. Because the Gamma distribution with $\alpha = 1$ is exponential, the restricted MLE is $\widehat{\theta}_0 = (1, \overline{d})$. It is more informative, though, to use numerical methods.

To maximize the likelihood function (or minimize minus the log likelihood) over $\Theta_0$, it might be tempting to impose the restriction on $\theta$, simplify the log likelihood, and write the code for a new function to minimize. But this strategy is *not* recommended. It's time consuming, and mistakes are possible. In the `R` work shown below, notice how the function `gmll1` is just a "wrapper" for the unrestricted minus log likelihood function `gmll`. It is a function of $\beta$ (and the data, of course), but all it does is call `gmll` with $\alpha$ set to one and $\beta$ free to vary.

```
> gmll1 <- function(b,datta) # Restricted gamma minus LL with alpha=1
+     { gmll1 <- gmll(c(1,b),datta)
+       gmll1
+     } # End of function gmll1
```

```
> mean(D) # Resticted MLE of beta, just to check
[1] 6.8782
```

The next step is to invoke the nonlinear minimization function `nlm`. The second argument is a (vector of) starting value(s). Starting the search at $\beta = 1$ turns out to be unfortunate.

```
> gsearch1 <- nlm(gmll1,1,datta=D); gsearch1
$minimum
[1] 282.6288

$estimate
[1] 278.0605

$gradient
[1] 0.1753689

$code
[1] 4

$iterations
[1] 100
```

The answer `g1search$estimate=278.0605` is way off the correct answer of $\overline{d} = 6.8782$, it took 100 steps, and the exit code of 4 means the function ran out of the default number of iterations. Starting at the unrestricted $\widehat{\beta}$ works better.

```
> gsearch1 <- nlm(gmll1,betahat,datta=D); gsearch1
$minimum
[1] 146.4178

$estimate
[1] 6.878195

$gradient
[1] -1.768559e-06

$code
[1] 1

$iterations
[1] 7
```

That's better. Good starting values are important! Now the test statistic is easy to calculate.

```
> Gsq = 2 * (gsearch1$minimum-gammasearch$minimum); pval = 1-pchisq(Gsq,df=1)
> Gsq; pval
[1] 8.772448
[1] 0.003058146
```

Let us carry out the other two tests, and then compare the Wald and likelihood ratio test results together in a table.

For $H_0 : \alpha = \beta$, the restricted parameter space is $\Theta_0 = \{(\alpha, \beta) : \alpha = \beta > 0\}$.

```
> gmll2 <- function(ab,datta) # Restricted gamma minus LL with alpha=1
+     { gmll2 <- gmll(c(ab,ab),datta)
+       gmll2
+     } # End of function gmll2
> abstart = (alphahat+betahat)/2
> gsearch2 <- nlm(gmll2,abstart,datta=D); gsearch2
Warning messages:
1: NaNs produced in: log(x)
2: NA/Inf replaced by maximum positive value
$minimum
[1] 144.1704

$estimate
[1] 2.562369

$gradient
[1] -4.991384e-07

$code
[1] 1

$iterations
[1] 4

> Gsq = 2 * (gsearch2$minimum-gammasearch$minimum); pval = 1-pchisq(Gsq,df=1)
> Gsq; pval
[1] 4.277603
[1] 0.03861777
```

This seems okay; it only took 4 iterations and the exit code of 1 is a clean bill of health. But the warning messages are a little troubling. Probably they just indicate that the search tried a negative parameter value, outside the parameter space. The `R` function `nlminb` does minimization with bounds. Let's try it.

```
> gsearch2b <- nlminb(start=abstart,objective=gmll2,lower=0,datta=D); gsearch2b
$par
```

```
[1] 2.562371

$objective
[1] 144.1704

$convergence
[1] 0

$message
[1] "relative convergence (4)"

$iterations
[1] 5

$evaluations
function gradient
       7        8
```

Since `nlminb` gives almost the same restricted $\widehat{\alpha} = \widehat{\beta} = 2.5624$ (and no warnings), the warning messages from `nlm` were probably nothing to worry about.

Finally, for $H_0 : \alpha = 2, \beta = 3$ the restricted parameter space $\Theta_0$ is a single point and no optimization is necessary. All we need to do is calculate the minus log likelihood there.

```
> Gsq = 2 * (gmll(c(2,3),D)-gammasearch$minimum); pval = 1-pchisq(Gsq,df=1)
> Gsq; pval
[1] 2.269162
[1] 0.1319713
```

The top panel of Table A.1 shows the Wald and likelihood ratio tests that have been done on the Gamma distribution data. But this is $n = 50$, which is not a very large sample. In the lower panel, the same tests were done for a sample of $n = 200$, formed by adding another 150 cases to the original data set. The results are typical; the $\chi^2$ values are much closer except where they are far out on the tails, and both test lead to the same conclusions (though not always to the truth).

Like the Wald tests, likelihood ratio tests are very flexible and are distributed approximately as chi-square under the null hypothesis for large samples. In fact, they are *asymptotically equivalent* under $H_0$, meaning that if the null hypothesis is true, the difference between the likelihood ratio statistic and the Wald statistic goes to zero in probability as the sample size approaches infinity.

Since the Wald and likelihood ratio tests are equivalent, does it matter which one you use? The answer is that usually, Wald tests and likelihood ratio tests lead to the same conclusions and their numerical values are close. But the tests are only equivalent as $n \to \infty$. When there is a meaningful difference, the likelihood ratio tests usually perform better, especially in terms of controlling Type I error rate for relatively small sample sizes.

Table A.1: Tests on data from a gamma distribution with $\alpha = 2$ and $\beta = 3$

| $n = 50$ | | | | |
|---|---|---|---|---|
| | Wald | | Likelihood Ratio | |
| $H_0$ | $\chi^2$ | $p$-value | $\chi^2$ | $p$-value |
| $\alpha = 1$ | 5.8421 | 0.0156 | 8.7724 | 0.0031 |
| $\alpha = \beta$ | 3.2455 | 0.0762 | 4.2776 | 0.0386 |
| $\alpha = 2, \beta = 3$ | 1.3305 | 0.5141 | 2.2692 | 0.1320 |
| $n = 200$ | | | | |
| $\alpha = 1$ | 34.1847 | 5.01e-09 | 58.2194 | 2.34e-14 |
| $\alpha = \beta$ | 0.9197 | 0.3376 | 0.9664 | 0.3256 |
| $\alpha = 2, \beta = 3$ | 1.5286 | 0.4657 | 1.2724 | 0.2593 |

Table A.2: Wald versus likelihood ratio: Type I error in 10,000 simulated datasets

| | $n$ | | | | |
|---|---|---|---|---|---|
| **Test** | 50 | 100 | 250 | 500 | 1000 |
| Wald | 1180 | 1589 | 1362 | 0749 | 0556 |
| Likelihood Ratio | 0330 | 0391 | 0541 | 0550 | 0522 |

Table A.2 below contains the most extreme example I know. For a particular structural equation model with normal data (details don't matter for now), ten thousand data sets were randomly generated so that the null hypothesis was true. This was done for several sample sizes: $n = 50, 100, 250, 500$ and $1,000$. Using each of the 50,000 resulting data sets, the null hypothesis was tested with a Wald test and a likelihood ratio test at the $\alpha = 0.05$ significance level. If the asymptotic results held, we would expect both tests to reject $H_0$ 500 times at each sample size.

So for this deliberately nasty example, the Wald test requires $n = 1,000$ before it settles down to something like the theoretical 0.05 significance level. The likelihood ratio test needs $n = 250$, and for smaller sample sizes it is conservative, with a Type I error rate somewhat *lower* than 0.05[18]. In general, when the Wald and likelihood ratio tests have a contest of this sort, it is usually a draw. When there is a winner, it is always the likelihood ratio test, but the margin of victory is seldom as large as this.

### Exercises A.6.8

A.6.1) Let $Y_1, \ldots, Y_n$ be a random sample from a distribution with density $f(y) = \frac{1}{\theta} e^{-\frac{y}{\theta}}$ for $y > 0$, where the parameter $\theta > 0$. We are interested in testing $H_0 : \theta = \theta_0$.

[18] This suggests that the power will not be wonderful for smaller sample sizes, in this example. But keeping Type I error rates below 0.05 is the first priority.

(a) What is $\Theta$?

(b) What is $\Theta_0$?

(c) What is $\Theta_1$?

(d) Derive a general expression for the large-sample likelihood ratio statistic $G^2 = -2 \log \frac{\ell(\widehat{\widehat{\theta}})}{\ell(\widehat{\theta})}$.

(e) A sample of size $n = 100$ yields $\overline{Y} = 1.37$ and $S^2 = 1.42$. One of these quantities is unnecessary and just provided to irritate you. Well, actually it's a mild substitute for reality, which always provides you with a huge pile of information you don't need. Anyway, we want to test $H_0 : \theta = 1$. You can do this with a calculator. When I did it a long time ago I got $G^2 = 11.038$.

(f) At $\alpha = 0.05$, the critical value of chisquare with one degree of freedom is 3.841459. Do you reject $H_0$? Answer Yes or No.

A.6.2) The label on the peanut butter jar says peanuts, partially hydrogenated peanut oil, salt and sugar. But we all know there is other stuff in there too. In the United States, the Food and Drug administration requires that a shipment of peanut butter be rejected if it contains an average of more than 8 rat hairs per pound (well, I'm not sure if it's exactly 8, but let's pretend). There is very good reason to assume that the number of rat hairs per pound has a Poisson distribution with mean $\lambda$, because it's easy to justify a Poisson process model for how the hairs get into the jars. We will test $H_0 : \lambda = \lambda_0$.

(a) What is $\Theta$?

(b) What is $\Theta_0$?

(c) What is $\Theta_1$?

(d) Derive a general expression for the large-sample likelihood ratio statistic.

(e) We sample 100 1-pound jars, and observe a sample mean of $\overline{Y} = 8.57$. Should we reject the shipment? We want to test $H_0 : \lambda = 8$. What is the value of $G^2$? You can do this with a calculator. When I did it a long time ago I got $G^2 = 3.97$.

(f) Do you reject $H_0$ at $\alpha = 0.05$? Answer Yes or No.

(g) Do you reject the shipment of peanut butter? Answer Yes or No.

A.6.3) The normal distribution has density

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}.$$

Find an explicit formula for the MLE of $\theta = (\mu, \sigma^2)$. This example is in practically every mathematical statistics textbook, so the full solution is available. But please try it yourself first.

A.6.4) Write an `R` function that performs a large-sample likelihood ratio test of $H_0 : \sigma^2 = \sigma_0^2$ for data from a single normal random sample. The function should take the sample data and $\sigma_0^2$ as input, and return 3 values: $G^2$, the degrees of freedom, and the $p$-value. Run your function on the data in `var.dat`, testing $H_0 : \sigma^2 = 2$; see link to the data on the course web page.

For this question, you need to bring a printout with a listing of your function (showing how it is defined), and also part of an R session showing execution of the function, and the resulting output.

A.6.5) For $k$ samples from independent normal distributions, the usual one-way analysis of variance tests equality of means assuming equal variances. Now you will construct a large-sample likelihood ratio test for equality of means, except that you will *not* assume equal variances. Write an `R` function to do it.

Input to the function should be the sample data, in the form of a matrix. The first column should contain group membership (the explanatory variable). It is okay to assume that the unique values in this column are the integers from 1 to $k$. The second column should contain values of the normal random variates – the response variable.

The function should return 3 values: $G^2$, the degrees of freedom, and the $p$-value. Run your function on the sample in `kars.dat`; see link to the data on the course web page. This data set shows country of origin and gas mileage for a sample of automobiles.

A.6.6) Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be a random sample from a multivariate normal population with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$. Using the MLEs

$$\widehat{\boldsymbol{\mu}} \;=\; \overline{\mathbf{X}} \text{ and } \widehat{\boldsymbol{\Sigma}} \;=\; \frac{1}{n}\sum_{i=1}^{n}(\mathbf{X}_i - \overline{\mathbf{X}})(\mathbf{X}_i - \overline{\mathbf{X}})^\top,$$

derive the large-sample likelihood ratio test $G^2$ for testing whether the components of the random vectors $\mathbf{X}_i$ are independent. That is, we want to test whether $\boldsymbol{\Sigma}$ is diagonal. It is okay to use material from the class notes without proof.

A.6.7) Using `R`, write a program to compute the test you derived in the preceding question. Your program should return 3 values: $G^2$, the degrees of freedom, and the $p$-value. Run it on the sample in `fourvars.dat`; see link to the data on the course web page. Bring a printout listing your program and illustrating the run on `fourvars.dat`. Of course it would be nice if your program were general, but it is not required. Note that for this problem, numerical maximum likelihood is not needed. Both your restricted and your unrestricted MLEs can and should be in explicit form.

### A.6.9 The Bootstrap

Sometimes, the distribution of a statistic or vector of statistics can be tough to figure out. You may not be able to do it at all. Or, maybe you could get an asymptotic answer using

the multivariate delta method, but it would be a big job requiring extensive paper and pencil calculations followed by careful programming. The bootstrap, due to Efron [22], is a computer-intensive method that can yield fairly automatic answers in such situations.

Let $\mathbf{x} = (X_1, \ldots, X_n)$ be a random sample from some distribution $F$. Let $T = T(\mathbf{x})$ be a statistic or vector of statistics. We need to know the distribution of $T$; an approximate answer will be good enough. You should not turn up your nose at the word "approximate." Bootstrap solutions are approximate in the same sense that a consistent estimator is approximate.

The name "bootstrap" comes from the saying "Pull yourself up by your bootstraps." Figure A.3 shows a pair of boots[19]. The little loops at the back of the boots are the

Figure A.3: A pair of boots with bootstraps

bootstraps; if you hook your fingers in the loops, it's easier to pull your boots on. Pulling yourself up by your bootstraps is physically impossible, but it's a metaphor for getting the job done with the resources you have available, even though it may seem impossible.

To appreciate the statistical bootstrap, recall how the idea of a *sampling distribution* is introduced in an elementary statistics class. One does not terrorize the students by referring to functions of a random variable. Instead, the sampling distribution is described as follows. Imagine drawing repeated random samples from the same population. Either the sampling is with replacement, or the population is so large that the distinction between with and without replacement makes no difference. For each sample, calculate the statistic. Make a relative frequency histogram of the values of the statistic. As the number of samples increases, the histogram gets closer and closer to the sampling distribution of the statistic.

So, select a random sample from the population. If the sample size is large, the sample is similar to the population. Sample repeatedly from the sample with replacement; this is called *resampling*. Calculate the statistic for every bootstrap sample. A histogram of the resulting values approximates the shape of the sampling distribution of the statistic.

[19]This photograph was taken by Tarquin. It is licensed under a Creative Commons Attribution - ShareAlike 3.0 Unported License. For more information, see the entry at the wikimedia site.

To visualize re-sampling, think of writing the $n$ sample data values on marbles, putting the marbles in a jar, and drawing $n$ marbles with replacement. Naturally, there will be some repeats; don't worry about it. In many applications, you will be re-sampling *vectors* of data values, like $x_1$, $x_2$, $x_3$ and $x_4$. In such cases, keep the values from a given individual together[20]. Think of $n$ strings of beads, with four beads on each string. You randomly sample strings of beads. Of course, in practice all this is done by computer using pseudo-random number generation, but the physical analogy may be helpful as a way of understanding the process.

More formally, let $\mathbf{x} = (X_1, \ldots, X_n)$ be a random sample from some distribution $F$, possibly a multivariate distribution. $T = T(\mathbf{x})$ is a statistic or a vector of statistics. Form a "bootstrap sample" $\mathbf{x}^*$ by sampling $n$ values from $\mathbf{x}$ *with replacement*. Repeat this process $B$ times, obtaining $\mathbf{x}_1^*, \ldots, \mathbf{x}_B^*$. Calculate the statistic (or vector of statistics) for each bootstrap sample, obtaining $T_1^*, \ldots, T_B^*$. The relative frequencies of $T_1^*, \ldots, T_B^*$ approximate the sampling distribution of $T$.

It works because the empirical distribution converges to the true distribution function.

$$\widehat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} I\{X_i \leq x\} \stackrel{a.s.}{\rightarrow} E(I\{X_i \leq x\}) = F(x)$$

Resampling from $\mathbf{x}$ with replacement is the same as simulating a random variable whose distribution is the empirical distribution function $\widehat{F}(x)$. Suppose the distribution function of $T$ is a nice smooth function of $F$. Then as $n \to \infty$ and $B \to \infty$, bootstrap sample moments and quantiles[21] of $T_1^*, \ldots, T_B^*$ converge to the corresponding moments and quantiles of the distribution of $T$. If the distribution of $\mathbf{x}$ is discrete and supported on a finite number of points, the technical issues are modest. For continuous distributions with unbounded support it's more challenging, but the conclusions still hold.

### Estimating the covariance matrix of a vector of statistics

In structural equation modeling, it is quite common to have a vector of estimators that are known to be consistent and asymptotically multivariate normal. An asymptotic variance-covariance matrix is available provided that the observable data are multivariate normal, but the normality assumption is either doubtful or demonstrably false. So constructing tests and confidence intervals is not routine.

There are two main ways this situation can emerge. In the first scenario, the statistics in question are nice explicit functions of the sample variance-covariance matrix of the observable data. Even when the data are not normally distributed, Theorem A.1 on page 564 establishes that the joint distribution of the sample variances and covariances is

[20] Well, if you were interested in testing independence of $x_1$ and $x_2$ from $x_3$ and $x_4$, you could put the $(x_1, x_2)$ pairs in one jar and the $(x_3, x_4)$ pairs in another jar, and draw independently from the two jars to assemble a set of four values. This is an example of *bootstrapping under the null hypothesis*, a very nice way to construct tests that make no assumptions about the distribution of the data.

[21] The $q$ quantile of a distribution is the point with $q$ of the distribution at or below it, where $0 \leq q \leq 1$. Quantiles are like percentiles.

asymptotically multivariate normal, and then by the multivariate delta method, differentiable functions of those variances and covariances are approximately multivariate normal too. The asymptotic variances and covariances of the sample variances and covariances – and functions of them – are actually available and can be estimated consistently, but it's a big, unpleasant chore.

In the other scenario, the statistics in question are MLEs, but they are MLEs based on the assumption that the observable data are multivariate normal – an assumption that is questionable or worse. The good news is that by Theorem 5.1 and the "Corollary to Huber's corollary" (Expression 5.4 on page 432) in Chapter 5, these pseudo-MLEs are consistent and have an asymptotic distribution that is multivariate normal. The bad news is that the normal-theory estimates of the asymptotic variance-covariance matrix are incorrect in general, though some exceptions are given in Chapter 5. Again, estimating the right variance-covariance matrix is not out of the question, but it's a big job involving mathematical calculations and computer coding that might never be needed again.

It's a lot easier using the bootstrap. The bootstrap provides a good picture of the sampling distribution of that vector of statistics. The only feature of the sampling distribution that matters is their variance-covariance matrix. Proceed as follows. Draw $B$ bootstrap samples from the sample data, and for each one calculate the vector of statistics. Assemble the results into a sort of data file, with $B$ rows, and one column for each statistic. Calculate the sample variance-covariance matrix of that. The result is an excellent approximation of the asymptotic variance-covariance matrix that's needed for tests and confidence intervals.

Here is an example. In the United States, admission to university is sometimes based partly on the Scholastic Aptitude Test, or SAT. In the old days there were two subtests, Verbal and Math. The data file `openSAT.data.txt`[22] has Verbal score, Math score and first-year grade point average for a sample of 200 students. We first read the data and look at the correlation matrix.

```
> sat = read.table("https://www.utstat.toronto.edu/brunner/openSEM/data/openSAT.data.txt")
> head(sat)
  VERBAL MATH  GPA
1    578  567 2.68
2    474  653 2.51
3    546  657 1.95
4    664  686 2.81
5    600  619 2.79
6    488  738 2.36
> cor(sat)
          VERBAL      MATH       GPA
VERBAL 1.0000000 0.2751041 0.3224927
MATH   0.2751041 1.0000000 0.1941086
GPA    0.3224927 0.1941086 1.0000000
```

These correlations are not too impressive, but remember that the students were admitted largely on the basis of having high SAT scores, so this is an example of how restricted

[22]This is a reconstructed data set based on a Minitab data set. I believe the Minitab data set is a cleaned-up version of real data from Penn State University.

range can weaken an observed correlation. Verbal score appears to be more highly correlated with GPA than Math score, but is the difference statistically significant? This is a meaningful but non-standard question.

By Theorem A.1 and the multivariate delta method, the asymptotic distribution of the sample correlation coefficients is multivariate normal and centered on the true correlations. For a Wald test and a confidence interval, all we need is an estimate of the covariance matrix.

Now we'll follow the recipe. Put the row numbers in a "jar." Sample from the jar with replacement, putting the rows into a bootstrap data set. Calculate the correlations. Do this $B$ times, saving the results in an array that will be called `bootdata`.

```
> # Bootstrap the correlations
> n = dim(sat)[1] # Sample size is the number of rows in the data file
> set.seed(9999) # Set random number seed so results can be duplicated.
> jar = 1:n; B = 1000
> bootdata = matrix(NA,B,3)
> colnames(bootdata) = c('Verbal-Math','Verbal-GPA','Math-GPA')
> for(j in 1:B)
+
+     rowz = sample(jar,size=n,replace=TRUE)
+     xstar = sat[rowz,]
+     kor = cor(xstar)
+     bootdata[j,1] = kor[1,2] # Correlation of Verbal with Math
+     bootdata[j,2] = kor[1,3] # Correlation of Verbal with GPA
+     bootdata[j,3] = kor[2,3] # Correlation of   Math with GPA
+      # Next bootstrap sample
> head(bootdata)
     Verbal-Math Verbal-GPA  Math-GPA
[1,]   0.3020368  0.3171977 0.2320282
[2,]   0.3589208  0.2834930 0.2247893
[3,]   0.1572560  0.3590254 0.2988522
[4,]   0.1989407  0.3582051 0.0998772
[5,]   0.3165621  0.3644107 0.2394445
[6,]   0.2808987  0.2934830 0.1626899
```

The estimated covariance matrix we need is just the sample covariance matrix of these bootstrapped statistics.

```
> Vhat = var(bootdata); Vhat # Asymptotic covariance matrix
             Verbal-Math   Verbal-GPA    Math-GPA
Verbal-Math 0.0044099830 0.0002516633 0.001059281
Verbal-GPA  0.0002516633 0.0037209355 0.001182263
Math-GPA    0.0010592808 0.0011822628 0.004240506
```

To test for difference between the two correlations, we'll use the `Wtest` function. The present application isn't quite a Wald test strictly speaking, but the theory applies.

```
> # Now use it
> # Test H0: Corr(Verbal,GPA) = Corr(Math,GPA)
> source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/Wtest.txt")
```

```
> # function(L,Tn,Vn,h=0) # H0: L theta = h
> LL = cbind(0,1,-1)
> estcorr = c(corsat[1,2],corsat[1,3],corsat[2,3])
> Wtest(L=LL,Tn=estcorr,Vn=Vhat)
         W          df     p-value
2.94491891 1.00000000 0.08614802
```

So the difference between is not statistically significant at the 0.05 level. How about a confidence interval?

```
> # 95 percent CI for Corr(Verbal,GPA) - Corr(Math,GPA)
> estdiff = corsat[1,3]-corsat[2,3]; estdiff # Estimated difference between correlations
[1] 0.128384
> sediff = as.numeric(sqrt( LL %*% Vhat %*% t(LL) ))
> CI = c(estdiff - 1.96*sediff, estdiff + 1.96*sediff); round(CI,4)
[1] -0.0182  0.2750
```

Observe that the confidence interval includes zero, which must happen since the hypothesis of zero difference was not rejected. It absolutely *must* happen because squaring the $z$ statistic corresponding to the confidence interval yields the Wald chi-square.

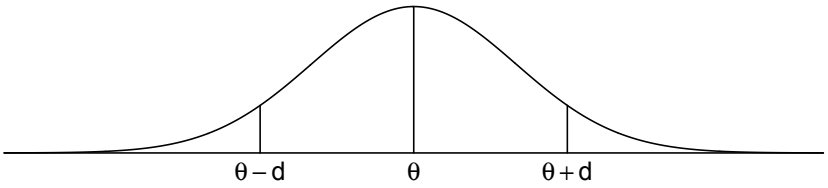
**Bootstrapping MLEs** In structural equation modeling it is common practice to estimate the model parameters with normal theory maximum likelihood, even if there is no particular reason to believe that the data are normally distributed. Fortunately, almost regardless of the distribution of the sample data, the resulting estimators are consistent by Theorem 5.1, and have asymptotically normal distributions by Corollary 5.4 on page 432. The normal theory estimates of the variances and covariances of the estimators might not be correct (see Chapter 5), but that problem is neatly solved by bootstrapping the pseudo-MLE's and estimating their variance-covariance matrix, exactly as in the example above. In `lavaan`, the `se="bootstrap"` option does the trick. Here are a couple of examples.

```
boot = lavaan(fullmod, data=X, se="bootstrap")
fit3 = cfa(model3,data=simdat, se="bootstrap")
```

**Quantile Bootstrap Confidence Intervals** An alternative to normal-theory confidence intervals are the quantile confidence intervals, which use more information about the exact shape of the sampling distribution out on the tails. Suppose $T_n$ is a consistent estimator of $\theta$, and the distribution of $T_n$ is approximately symmetric around $\theta$. Then the lower $(1-\alpha)100\%$ confidence limit for $\theta$ is the $\alpha/2$ sample quantile of $T_1^*, \ldots, T_B^*$, and the upper limit is the $1-\alpha/2$ sample quantile. For example, the 95% confidence interval ranges from the 2.5th to the 97.5th percentile of $T_1^*, \ldots, T_B^*$.

Symmetry is a requirement that is often ignored when computing quantile bootstrap intervals. The distribution of $T_n$ symmetric about $\theta$ means for all $d > 0$, $P\{T_n > \theta + d\} = P\{T_n < \theta - d\}$. See Figure A.4.

Figure A.4: A symmetric distribution




Select $d$ so that $P\{T_n > \theta + d\} = P\{T_n < \theta - d\}$ equals $\alpha/2$. Then

$$\begin{aligned} 1-\alpha &= P\{\theta - d < T_n < \theta + d\} \\ &= P\{T_n - d < \theta < T_n + d\} \end{aligned}$$

To use this result, an estimate of $d$ is required.

There are two natural estimates. Letting $Q_{\alpha/2}$ denote the true $\alpha/2$ quantile of the distribution of $T_n$,

$$1-\alpha = P\{\theta - d < T_n < \theta + d\} = P\{Q_{\alpha/2} < T_n < Q_{1-\alpha/2}\}.$$

The estimates should satisfy

$$\begin{array}{rclcrcl} \widehat{\theta} - \widehat{d}_1 & = & \widehat{Q}_{\alpha/2} & \Rightarrow & \widehat{d}_1 & = & T_n - \widehat{Q}_{\alpha/2} \\ \widehat{\theta} + \widehat{d}_2 & = & \widehat{Q}_{1-\alpha/2} & \Rightarrow & \widehat{d}_2 & = & \widehat{Q}_{1-\alpha/2} - T_n, \end{array}$$

where $T_n$ has been used to estimate $\theta$, and $\widehat{Q}_{\alpha/2}$ and $\widehat{Q}_{1-\alpha/2}$ are the bootstrap quantiles.

Then, take $1-\alpha = P\{T_n - d < \theta < T_n + d\}$ and plug in the estimates of $d_1$ and $d_2$. Using $\widehat{d}_1$ on the left yields

$$T_n - \widehat{d}_1 = T_n - (T_n - \widehat{Q}_{\alpha/2}) = \widehat{Q}_{\alpha/2}$$

Using $\widehat{d}_2$ on the right yields

$$T_n + \widehat{d}_2 = T_n + (\widehat{Q}_{1-\alpha/2} - T_n) = \widehat{Q}_{1-\alpha/2},$$

so that the $(1-\alpha)100\%$ bootstrap quantile confidence interval is

$$\left(\widehat{Q}_{\alpha/2}, \widehat{Q}_{1-\alpha/2}\right). \tag{A.39}$$

There are indications that the coverage of this interval can approach $1-\alpha$ faster with increasing sample size than a confidence interval based on the central limit theorem. See Chapter 22 of Efron and Tibshirani [23].

To test hypotheses like $H_0 : \theta = \theta_0$, one can simply check whether the $(1-\alpha)100\%$ quantile confidence interval for $\theta$ includes $\theta_0$, and reject the null hypothesis at significance level $\alpha$ if it does.

**Justifying the Assumption of Symmetry** All this depends on the statistic $T_n$ having a distribution that is approximately symmetric. When the distribution of the estimator is not symmetric about the parameter being estimated, quantile confidence intervals are unjustified and often quite inaccurate. Ignoring this point has led to confusion ans suspicion about the bootstrap, especially among non-statisticians. So how does one justify the assumption of symmetry, particularly when the distribution of $T_n$ is elusive? The easiest answer is asymptotic normality. Smooth functions of asymptotic normals are asymptotically normal, and this includes maximum likelihood estimators as well as functions of the sample moments. Of course the normal distribution is symmetric, and this justifies the use of quantile confidence intervals. Here is an illustration using the SAT data.

```
> # Now a quantile confidence interval
> difcorr = bootdata[,2]-bootdata[,3]
> difcorr = sort(difcorr)
> # 0.025 * 1000 = 25, so go midway between number 25 and number 26,
> # And midway between number 974 and 975
> LowerQuant = (difcorr[25]+difcorr[26])/2
> UpperQuant = (difcorr[974]+difcorr[975])/2
> qCI = c(LowerQuant,UpperQuant) # 95% Quantile interval
> round(qCI,4)
[1] -0.0281  0.2704
```

This confidence interval is very similar to the one directly based on asymptotic normality. Again, it provides no evidence that the correlation between Verbal SAT and first-year GPA is different from the correlation between Math SAT and first-year GPA.

# Appendix B

# Symbolic Mathematics with Sagemath

## B.1 Introduction to Sagemath

### What is Sagemath, and why use it?

Sagemath is free, open source mathematics software. Lots of software can carry out numerical calculations, and so can Sagemath. What makes Sagemath special is that it can also do *symbolic* computation. That is, it is able to manipulate symbols as well as numbers.

If you think about it, you will realize that a lot of the "mathematics" you do in your statistics courses does not really require much mathematical thinking. Sometimes, all you are really doing is pushing symbols around. You might have to do something like partially differentiate a log likelihood function with respect to several variables, set all the expressions to zero and solve the resulting equations. To do this you need to know some rules, apply them accurately, and pay attention to detail. This kind of "thinking" is something that computers do a lot better than humans. So particularly for big, complicated tasks, why not let a computer do the grunt work? Symbolic mathematics software is designed for this purpose.

There are several commercial products that do symbolic math. The best known are Mathematica (http://www.wolfram.com) and Maple (http://www.maplesoft.com). There are also quite a few free, open source alternatives that are developed and maintained by volunteers. Sagemath is one of them. What makes Sagemath really special is that in addition to its own core capabilities, it incorporates and more or less unifies quite a few of the other mathematical programs using a single convenient interface. After all, why not? They are free and open source, so there are no legal obstacles (like copyrights) to prevent the Sagemath programmers from sending a particular task to the program that does it best[1].

It's all accomplished with Python scripts. In fact, Sagemath is largely a set of sophisticated Python functions. So if you know the Python programming language, you have a

[1] A by-product of this approach is that if you download a copy of Sagemath, you'll see that it's *huge*. This is because you're really downloading six or seven complete applications.

huge head start in learning Sagemath. If you want to do something in Sagemath and you can figure out how to do it in Python, try it. Probably the Python code will work.

### Reference Materials

This appendix is intended to be more or less complete. For further information and documentation, see the Sagemath project home page at http://www.sagemath.org. Another useful source of information is the Wikipedia article:

http://en.wikipedia.org/wiki/Sage_(mathematics_software)

## A Guided tour

To follow this tour actively by trying things out as you read about them, you will need access to Sagemath, either on your computer or on a server. For more information, see Section B.3: Using Sagemath on your Computer.

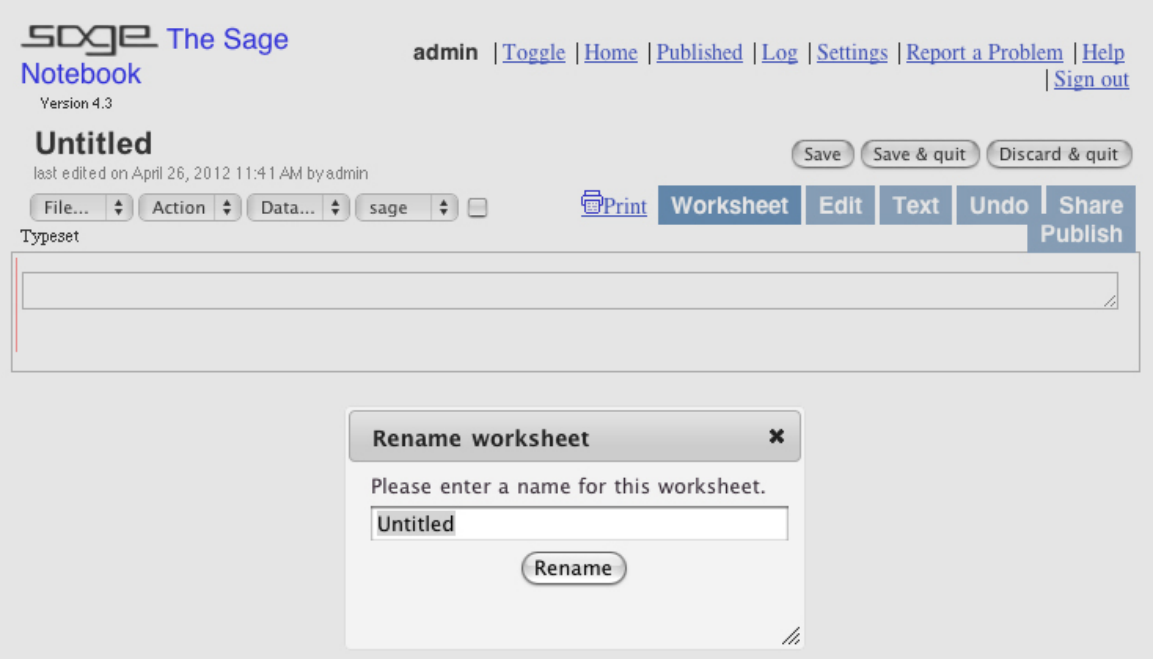
### The Interface

Sagemath has a browser interface. So, whether the software resides on a remote server or you have downloaded and installed your own free copy as described in Section B.3, you type your input and see your output using an ordinary Web browser like Firefox.

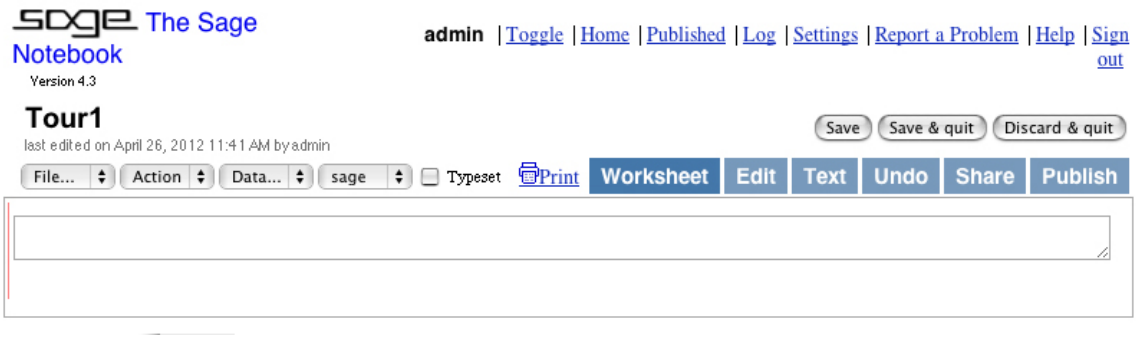
Sagemath also has a text-only interface, in which the output as well as input is in plain text format. Many mathematicians who use Sagemath prefer the simplicity of plain text, and most Sagemath documentation uses plan text. But a great strength of Sagemath, and our main reason for using it, is that we can manipulate and view the results of calculations using Greek symbols. This capability depends on the browser interface, so we'll stick exclusively to that.

When you first start up Sagemath, you'll see the Sagemath *Notebook* with a list of your active *Worksheets*. You can save your worksheets and go back to them later. It's great, but right now you don't have any worksheets. Your screen looks roughly like this:

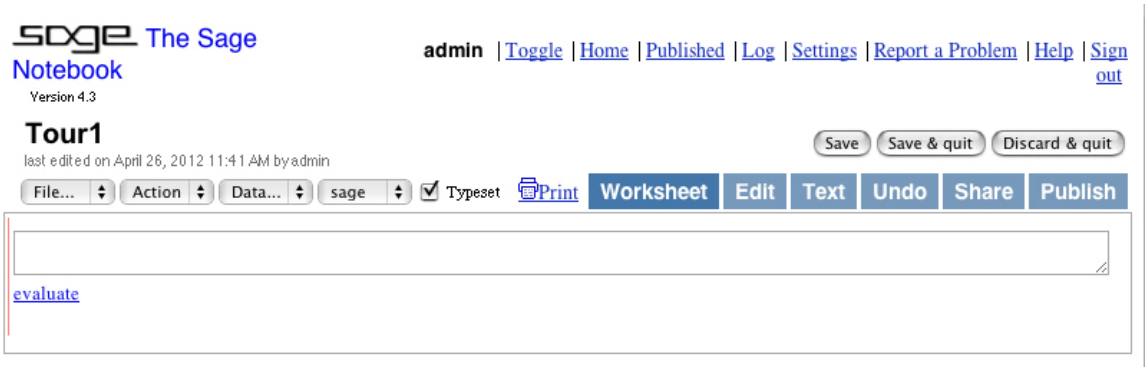

Click on "New Worksheet." A new window opens. It looks like this:



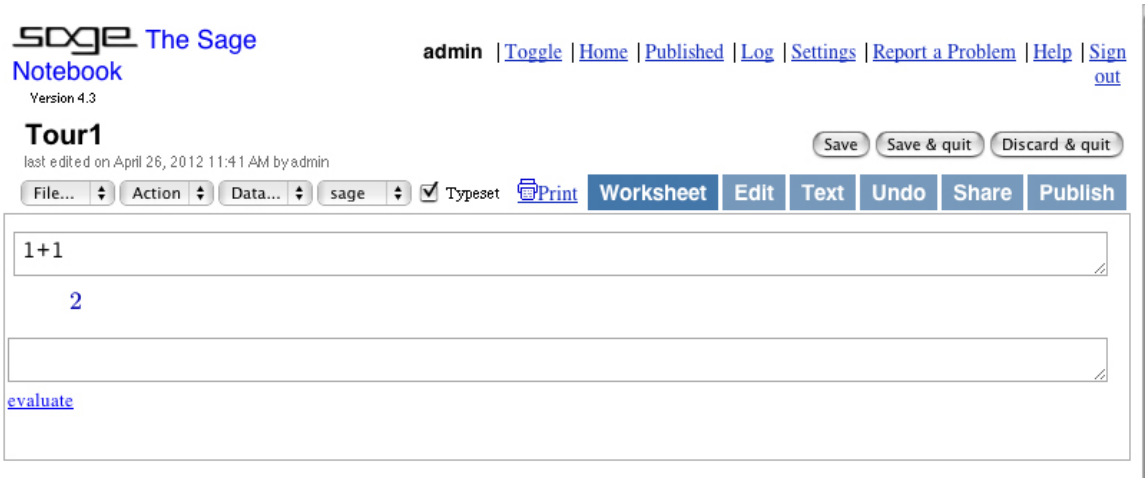

Type in an informative name and click Rename. I called mine Tour1, because we're on a guided tour of Sagemath. Now the browser window looks like something like this:



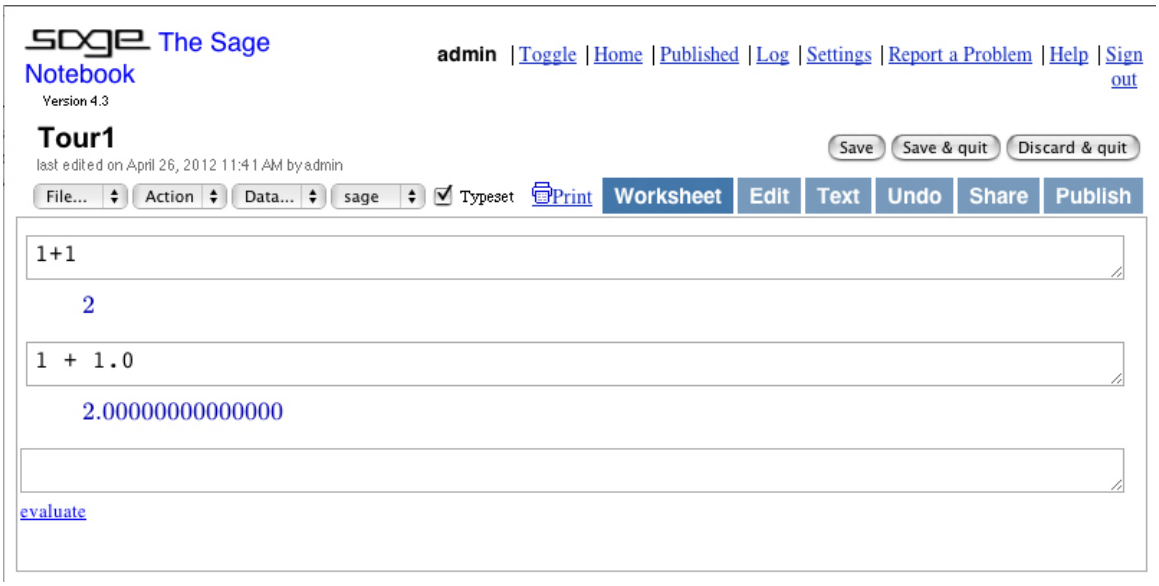

You definitely want to check the "Typeset" box, so you can see nice Greek letters. Now, the way it works is that you type (or paste) your commands into the upper box and Sagemath writes the output in the box below it. As soon as you click in the upper box, the underlined word evaluate appears below. It looks like this.



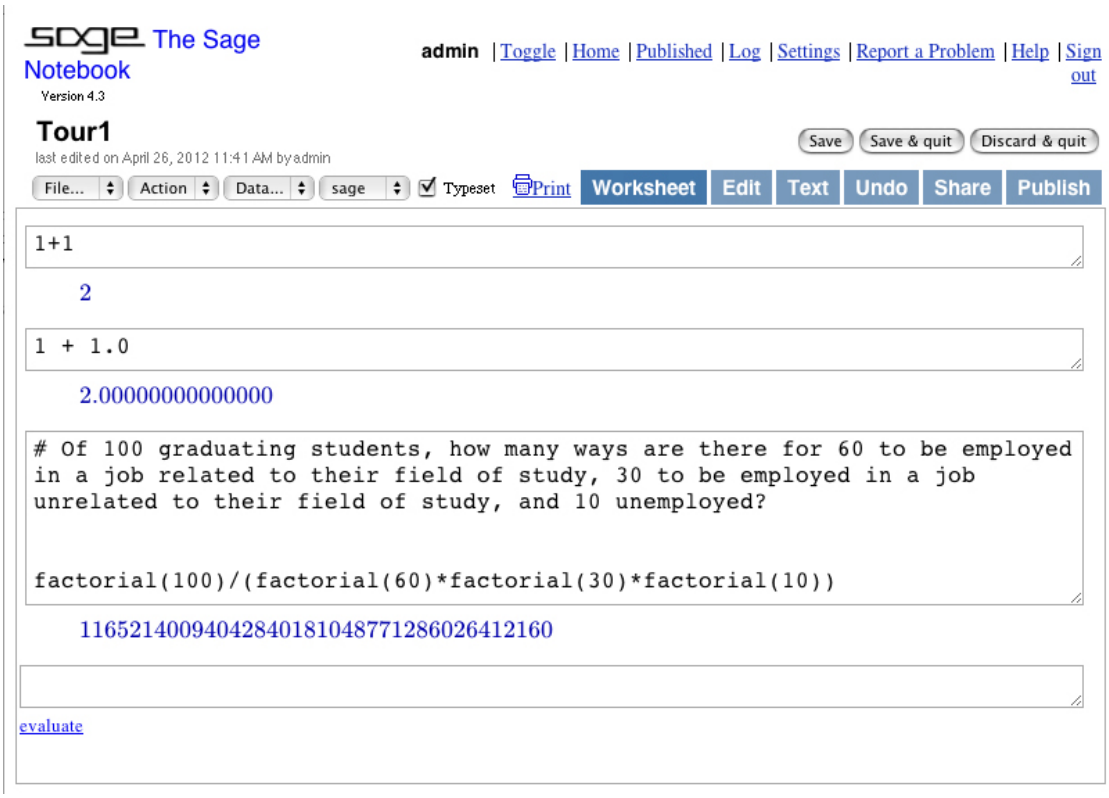
Now you type your input, which in this case is numerical as well as mathematically profound. Pressing the Enter (or Return) key just lets you type another line of input. To execute the command(s), click evaluate. An alternative to clicking evaluate is to hold down the Shift key and press Enter. Here is the result.




Notice that now there's another box for your next set of input. Here's a variation on $1 + 1 = 2$.




In the first case, `Sagemath` was doing integer arithmetic. In the second case, part of the input was interpreted as real-valued because it had a decimal point. Integer plus real is real, so `Sagemath` converted the 1 to 1.0 and did a floating-point calculation. This kind of "dynamic typing" is a virtue that `Sagemath` shares with Python. `Sagemath` is very good at integer arithmetic. In the next example, everything following # is a comment.




For comparison, this is how the calculation goes in `R`.

```
> prod(1:100)/(prod(1:60)*prod(1:30)*prod(1:10))
[1] 1.165214e+37
```

The whole thing is a floating point calculation, and `R` returns the answer in an imprecise scientific notation.

Exact integer arithmetic is nice, but it's not why we're using `Sagemath`. Let's calculate the third derivative $\frac{\partial^3}{\partial x^3}\left(\frac{e^{4x}}{1+e^{4x}}\right)$. This is something you could do by hand, but would you want to?


You can see how the worksheet grows. At any time, you can click on the Save button if you like what you have. You can also print it just as you would any other Web page.

You can edit the contents of an input box by clicking in the box. When you do, evaluate appears beneath the box. Click on it, and the code in the box is executed. You can re-do all the calculations in order by choosing Evaluate All from the Action menu (upper left). When you quit Sagemath and come back to a worksheet later, you may want to Evaluate All so all the objects you've defined – like $f(x)$ above – are available. When you're done (for the present), click the Save & Quit button. If you click Discard & Quit, all the material since the last Save will be lost; sometimes this is what you want. When you Save & Quit, you see something like this:

Click on Sign out (upper right) and you're done. Next time you run `Sagemath` the worksheet will be available. You can double-click on it to work on it some more, or start a new one.

The guided tour will resume now, but without continuing to illustrate the interface. Instead, the input will be given in a typewriter typeface `like this`, and then the output will given, usually in typeset form[2].

### Limits, Integrals and Derivatives (Plus a little plotting and solving)

Now we return to the Tour1 worksheet and choose Evaluate All from the Action menu. Then

```
f(x)
```

and clicking on evaluate yields

$$\frac{e^{(4\,x)}}{\left(e^{(4\,x)}+1\right)}$$

This really looks like a cumulative distribution function. Is it? Let's try $\lim_{x\to-\infty} f(x)$.

```
limit(f(x),x=-Infinity);limit(f(x),x=Infinity)
```

evaluate

0
1

Okay! So it's a distribution function. Notice the two commands on the same line, separated by a semi-colon. Without the semi-colon, only the last item is displayed. An alternative to the semi-colon is the `show command`:

```
show(limit(f(x),x=-Infinity))
show(limit(f(x),x=Infinity))
```

evaluate

0

1

The (single) derivative of $f(x)$ is a density.

[2]In case you are interested in how this works, `Sagemath` uses the open source LaTeX typesetting system to produce output in mathematical script. The LaTeX code produced by `Sagemath` is available. So, in the Tour1 worksheet, if I enter `f(x)` in the input box, I get nice-looking mathematical output (see above). Then if I type `print(latex(_))` in the next input box, I get the LaTeX code for the preceding expression. Since this book is written in LaTeX, I can directly paste in the machine-generated LaTeX code without having to typeset it myself. My code might be a bit cleaner and more human-readable, but this is very convenient.

```
derivative(f(x),x)
```

evaluate

$$4\,\frac{e^{(4\,x)}}{\left(e^{(4\,x)}+1\right)}-4\,\frac{e^{(8\,x)}}{\left(e^{(4\,x)}+1\right)^2}$$

Here is another way to get the same thing.

```
# Another way
f(x).derivative(x)
```

evaluate

$$\frac{4\,e^{(4\,x)}}{e^{(4\,x)}+1}-\frac{4\,e^{(8\,x)}}{\left(e^{(4\,x)}+1\right)^2}$$

This second version of the syntax is more like Python, and makes it clear that the derivative is an *attribute*, or *method* associated with the object $f(x)$. Many tasks can be requested either way, but frequently only the second form (object followed by a dot, followed by the attribute) is available. It is preferable from a programming perspective.

The expression for $f'(x)$ could and should be simplified. `Sagemath` has a `simplify` command that does nothing in this case and in many others, because `simplify` is automatically applied before any expression is displayed. But `factor` does the trick nicely.

```
g(x) = factor(f(x).derivative(x)); g(x)
```

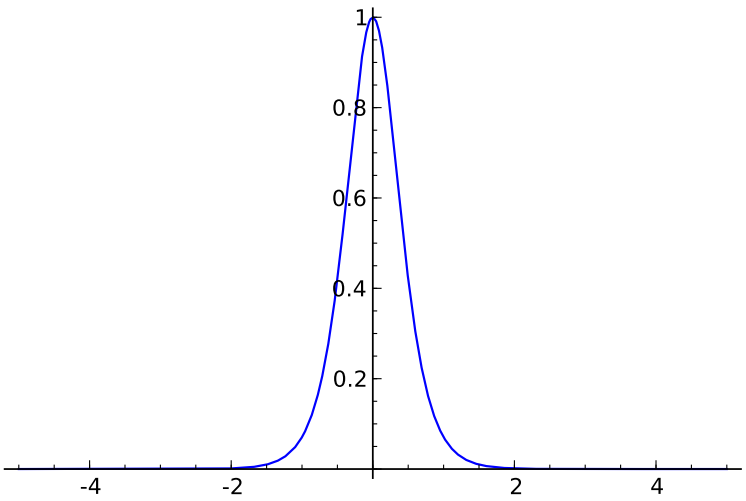
evaluate

$$\frac{4\,e^{(4\,x)}}{\left(e^{(4\,x)}+1\right)^2}$$

Want to see what it looks like? Plotting functions is straightforward.

```
plot(g(x),x,-5,5)
```

evaluate



It's easy to add labels and so on to make the plot look nicer, but that's not the point here. The objective was just to take a quick look to see what's going on.

Actually, the picture is a bit surprising. It *looks* like the density is symmetric around $x = 0$, which would make the median and the mean both equal to zero. But the formula for $g(x)$ above does not suggest symmetry. Well, it's easy to verify that the median is zero.

```
f(0)
```

evaluate

$\frac{1}{2}$

How about symmetry? The first try is unsuccessful, because the answer is not obviously zero (though it is). But then `factor` works.

```
g(x)-g(-x)
```

evaluate

$$\frac{4\,e^{(4\,x)}}{\left(e^{(4\,x)}+1\right)^2} - \frac{4\,e^{(-4\,x)}}{\left(e^{(-4\,x)}+1\right)^2}$$

```
factor(g(x)-g(-x))
```

evaluate

0

Is this right? Yes. To see it, just multiply numerator and denominator of $g(-x)$ by $e^{8x}$. `Sagemath` does not show its work, but it's a lot less likely to make a mistake than you are. And even if you're the kind of person who likes to prove everything, `Sagemath` is handy because it can tell you what you should try to prove.

Clearly, the number 4 in $f(x)$ is arbitrary, and could be any positive number. So we'll replace 4 with $\theta$. Now `Sagemath`, like most software, will usually complain if you try to use variables that have not been defined yet. So we have to declare $\theta$ as a symbolic variable, using a `var` statement. The variable $x$ is the only symbolic variable that does not have to be declared. It comes pre-defined as symbolic[3].

```
var('theta')
F(x) = exp(theta*x)/(1+exp(theta*x)); F(x)
```

evaluate

$\frac{e^{(\theta x)}}{e^{(\theta x)}+1}$

Is $F(x)$ a distribution function? Let's see.

[3]In Mathematica, all variables are symbolic by default unless they are assigned a numeric value. I wish `Sagemath` did this too, but I'm not complaining. `Sagemath` has other strengths that Mathematica lacks.

```
limit(F(x),x=-Infinity)
```

evaluate

```
Traceback (click to the left of this block for traceback)
...
Is  theta  positive, negative, or zero?
```

This is how error messages are displayed. You can click on the blank space to the left of the error message for more information, but in this case it's unnecessary. Sagemath asks a very good question about $\theta$. Well, actually, the question is asked by the excellent open-source calculus program Maxima, and Sagemath relays the question. In Maxima, you could answer the question interactively through the console and the calculation would proceed, but this capability is not available in Sagemath. The necessary information can be provided non-interactively. Go back into the box and edit the text.

```
assume(theta>0)
F(x).limit(x=-oo); F(x).limit(x=oo)
```

evaluate

0

1

Notice how two small letter o characters can be used instead of typing out Infinity. Now we'll differentiate $F(x)$ to get the density. It will be called $f(x)$, and that will *replace the existing definition* of $f(x)$.

```
f(x) = factor(F(x).derivative(x)); f(x)
```

evaluate

$$\frac{\theta e^{(\theta x)}}{\left(e^{(\theta x)}+1\right)^2}$$

Of course this density is also symmetric about zero, just like the special case with $\theta = 4$. It's easy to verify.

```
factor(f(x)-f(-x))
```

evaluate

0

Symmetry of the density about zero implies that the expected value is zero, because the expected value is the physical balance point. Direct calculation confirms this.

```
# Expected value
integrate(x*f(x),x,-oo,oo)
```

evaluate

0

It would be nice to calculate the variance too, but the variance emerges in terms of an obscure function called the `polylog`. The calculation will not be shown.

This distribution (actually, a version of the logistic distribution) is a good source of cute homework problems because the parameter $\theta$ has to be estimated numerically. So, for the benefit of some lucky future students, let's figure out how to simulate a random sample from $F(x)$. First, we'll add a location parameter, because two-parameter problems are more fun. The following definition rubs out the previous $F(x)$.

```
# Add a location parameter
var('mu')
F(x) = exp(theta*(x-mu))/(1+exp(theta*(x-mu))); F(x)
```

evaluate

$$\frac{e^{(-(\mu-x)\theta)}}{e^{(-(\mu-x)\theta)}+1}$$

I can't control the order of variables in `Sagemath` output. It looks alphabetical, with the `m` in `mu` coming before $x$.

It's well known that if $U$ is a random variable with a uniform density on the interval $(0, 1)$ and $F(x)$ is the cumulative distribution function of a continuous random variable, then if you transform $U$ with the *inverse* of $F(x)$, the result is a random variable with distribution function $F(x)$. Symbolically,

$$F^{-1}(U) = X \sim F(x)$$

Of course this is something you could do by hand, but it's so fast and easy with `Sagemath`:

```
# Inverse of cdf
var('X U')
solve(F(X)==U,X) # Solve F(X)=U for X
```

evaluate

$$\left[X = \frac{\mu\theta+\log\left(-\frac{U}{U-1}\right)}{\theta}\right]$$

It might be a bit better to write this as

$$X = \mu + \frac{1}{\theta}\log\left(\frac{U}{1-U}\right),$$

but what `Sagemath` gives us is quite nice. A few technical comments are in order. First, the double equal sign in `F(X)==U` indicates a *logical* relation. For example,

```
1==4
```

evaluate

False

Second, the `solve` returns a *list* of solutions. `Sagemath` uses brackets to indicate a list. In this case, there is only one solution so the list contains only one element. It's element *zero* in the list, not element one. Like Python, `Sagemath` starts all lists and array indices with element zero. It's a hard-core computer science feature, and mildly irritating for the ordinary user. Here's how one can extract element zero from the list of solutions.

```
solve(F(X)==U,X)[0]
```

evaluate

$$X = \frac{\mu\theta + \log\left(-\frac{U}{U-1}\right)}{\theta}$$

The equals sign in that last expression is actually a double equals. If you're going to use something like that solution in later calculations, it can matter. In `Sagemath`, the underscore character always refers to the output of the preceding command. It's quite handy. The `print` function means "Please don't typeset it."

```
print(_)
```

evaluate

```
X == (mu*theta + log(-U/(U - 1)))/theta
```

Just for completeness, here's how that inverse function could be used to simulate data from $F(x)$ in `R`.

```
> n = 20; mu = -2; theta = 4
> U = runif(n)
> X = mu + log(U/(1-U))/theta; X
 [1] -1.994528 -2.455775 -2.389822 -2.996261 -1.477381 -2.422011 -1.855653
 [8] -2.855570 -2.358733 -1.712423 -2.075641 -1.908347 -2.018621 -2.019441
[15] -1.956178 -2.015682 -2.846583 -1.727180 -1.726458 -2.207717
```

Random number generation is available from within `Sagemath` too, and in fact `R` is one of the programs incorporated in `Sagemath`, but to me it's more convenient to use `R` directly – probably just because I'm used to it.

You have to declare most variables (like $\theta$, $\mu$, $X$, $U$ and so on) before you can use them, but there are exceptions. The pre-defined symbolic variable $x$ is one. Here is another.

```
pi
```

evaluate

$$\pi$$

Is that really the ratio of a circle's circumference to its diameter, or just the Greek letter?

```
cos(pi)
```

evaluate

$-1$

That's pretty promising. Evaluate it numerically.

```
n(pi) # Could also say pi.n()
```

evaluate

3 : 14159265358979

```
gamma(1/2)
```

evaluate

$\sqrt{\pi}$

So it's really $\pi$. Let's try using `pi` in the normal distribution.

```
# Normal density
var('mu, sigma')
assume(sigma>0)
f(x) = 1/(sigma*sqrt(2*pi)) * exp(-(x-mu)^2/(2*sigma^2)); f(x)
```

evaluate

$$\frac{\sqrt{2}e^{\left(-\frac{(\mu-x)^2}{2\,\sigma^2}\right)}}{2\,\sqrt{\pi}\sigma}$$

```
# Integrate the density
integrate(f(x),x,-oo,oo)
```

evaluate

1

Calculate the expected value.

```
# E(X)
integrate(x*f(x),x,-oo,oo)
```

evaluate

$\mu$

Obtain the variance directly.

```
# E(X-mu)^2
integrate((x-mu)^2*f(x),x,-oo,oo)
```

evaluate

$\sigma^2$

Calculate the moment-generating function and use it to get $E(X^4)$.

```
# Moment-generating function M(t) = E(e^{Xt})
var('t')
M(t) = integrate(exp(x*t)*f(x),x,-oo,oo); M(t)
```

evaluate

$$e^{\left(\frac{1}{2}\,\sigma^2 t^2+\mu t\right)}$$

```
# Differentiate four times, set t=0
derivative(M(t),t,4)(t=0)
```

evaluate

$$\mu^4 + 6\,\mu^2\sigma^2 + 3\,\sigma^4$$

Discrete distributions are easy to work with, too. In the geometric distribution, a coin with $Pr\{\text{Head}\} = \theta$ is tossed repeatedly, and $X$ is the number of tosses required to get the first head. Notice that two separate `assume` statements are required to establish $0 < \theta < 1$. All the commands work as expected, but only the output from the last one is displayed.

```
# Geometric
var('theta')
assume(0<theta); assume(theta<1)
p(x) = theta*(1-theta)^(x-1); p(x)
p(x).sum(x,1,oo)                  # Sum the pmf
(x*p(x)).sum(x,1,oo)              # Expected value
((x-1/theta)^2*p(x)).sum(x,1,oo) # Variance
```

evaluate

$$-\frac{\theta-1}{\theta^2}$$

In the next example, the parameter $\lambda$ of the Poisson distribution must be treated specially because it has a specific advanced programming meaning and the word is reserved. It can still be used as a symbol if it is assigned to a variable *and* used with an underscore as illustrated. Lambdas with subscripts present no problems. In fact, `lambda_` can be viewed as a $\lambda$ with an invisible subscript.

```
# Poisson - lambda has a special meaning. But if you assign
# it to a variable and define it WITH AN UNDERSCORE you can
# still use it as a symbol.
L = var('lambda_')
p(x) = exp(-L) * L^x / factorial(x) ; p(x)
```

evaluate

$\frac{\lambda^x e^{(-\lambda)}}{x!}$

```
p(x).sum(x,0,oo)       # Sums nicely to one
(x*p(x)).sum(x,0,oo)   # Expected value
```

evaluate

$\lambda$

Here is some sample code for the Gamma distribution. Note the use of `full_simplify` on ratios of gamma functions.

```
# Gamma
var('alpha beta')
assume(alpha>0); assume(beta>0)
assume(alpha,'noninteger'); assume(beta,'noninteger')
f(x) = 1/(beta^alpha*gamma(alpha)) * exp(-x/beta) * x^(alpha-1)
integrate(f(x),x,0,oo) # Equals one
integrate(x*f(x),x,0,oo) # E(X)
```

evaluate

$\frac{\beta\Gamma(\alpha+1)}{\Gamma(\alpha)}$

```
_.full_simplify() # Underscore refers to the preceding expression.
```

evaluate

$\alpha\beta$

Now for the the moment-generating function. When I first tried it `Sagemath` asked "`Is beta*t-1 positive, negative, or zero?`" Because the moment-generating function only needs be defined in a neighbourhood of zero. I said `assume(beta*t<1)`, which is equivalent to $t < \frac{1}{\beta}$. In this way, `Sagemath` makes us specify the *radius of convergence* of the moment-generating function, but only when the radius of convergence is not the whole real line. `Sagemath` may be just a calculator, but it's a very smart calculator. It helps keep us mathematically honest. You have to love it.

```
# Moment-generating function
var('t'); assume(beta*t<1)
M(t) = integrate(exp(x*t)*f(x),x,0,oo).full_simplify(); M(t)
derivative(M(t),t,2)(t=0).full_simplify() # Lovely
```

evaluate

$(\alpha^2 + \alpha)\beta^2$

Here is some sample code for the Binomial distribution. Only the input is given.

```
# Binomial
```

```
var('n theta')
assume(n,'integer'); assume(n>-1)
assume(0<theta); assume(theta<1)
p(x) = factorial(n)/(factorial(x)*factorial(n-x)) * theta^x * (1-theta)^(n-x)
p(x).sum(x,0,n)                                    # Adds to one
(x*p(x)).sum(x,0,n).full_simplify()                # E(X)
(x^2*p(x)).sum(x,0,n).full_simplify()              # E(X^2)
((x-n*theta)^2*p(x)).sum(x,0,n).full_simplify() # cov(X) directly
```

### Maxima and Minima in Several Variables (Maximum Likelihood)

The standard way to derive maximum likelihood estimators is to partially differentiate the log likelihood with respect to each parameter, set the resulting expressions to zero, and solve for the parameters. This task is routine with Sagemath, except for one part. The "one part" is actually a nasty clerical chore that a symbolic math program like Sagemath *should* be able to do for us. Writing the likelihood function as

$$L(\theta) = \prod_{i=1}^{n} f(x_i|\theta),$$

the task is to carry out the multiplication, using the fact that multiplication is addition of exponents. The result is often an expression in the parameter $\theta$ and a a set of (sufficient) *statistics* – that is, functions of the sample data that could be calculated without knowing any of the parameters. I'm not insisting this step cannot be done with Sagemath, only that I've tried hard, I can't do it with Mathematica either, and other knowledgeable users[4] can't seem to make Sagemath do it either.

**The Univariate Normal Distribution** For the normal distribution, one version of the calculation goes like this.

$$\begin{aligned} L(\mu,\sigma) &= \prod_{i=1}^{n}\left(= \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right) \\ &= \frac{1}{\sigma^n(2\pi)^{n/2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2} \\ &= \frac{1}{\sigma^n(2\pi)^{n/2}} e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(x_i^2-2x_i\mu+\mu^2\right)} \\ &= \frac{1}{\sigma^n(2\pi)^{n/2}} e^{-\frac{1}{2\sigma^2}\left(\sum_{i=1}^{n}x_i^2-2\mu\sum_{i=1}^{n}x_i+n\mu^2\right)} \end{aligned}$$

[4]Somebody is a statistician in New Zealand who uses Sagemath in her classes. I have not asked her directly, but in the material she posts online she simplifies likelihood functions by hand, just as I am forced to do here.

This is not actually the best way to do the calculation. Better is to add and subtract $\overline{x}$ in the exponent. But this way requires a bit less insight (or experience), and leads to a more complicated problem that illustrates Sagemath's power. Continuing, the minus log likelihood function is

$$-\ell(\mu, \sigma) = n \log \sigma + \frac{n}{2} \log 2\pi + \frac{1}{2\sigma^2} \left( \left( \sum_{i=1}^{n} x_i^2 \right) - 2\mu \left( \sum_{i=1}^{n} x_i \right) + n\mu^2 \right).$$

Notice how the likelihood has been simplified to an expression that depends on the sample data only through a two-dimensional sufficient statistic[5]. This is what we need to minimize over the pair $(\mu, \sigma)$. In the Sagemath code, $\sum_{i=1}^{n} x_i$ will be denoted by $s_1$ and $\sum_{i=1}^{n} x_i^2$ will be denoted by $s_2$.

```
# Minus Log Likelihood for univariate normal
# s1 is sum of x, s2 is sum of x^2
var('mu sigma s1 s2 n')
mLL = n*log(sigma) + n/2 * log(2*pi) + 1/(2*sigma^2) * (s2 - 2*mu*s1 + n*mu^2)
mLL
```

evaluate

$$\frac{1}{2}\, n \log\left(2\,\pi\right) + n \log\left(\sigma\right) + \frac{\mu^2 n - 2\,\mu s_1 + s_2}{2\,\sigma^2}$$

Now partially differentiate the minus log likelihood with respect to $\mu$ and $\sigma$, set the derivates to zero, and solve.

```
d1 = derivative(mLL,mu); d2 = derivative(mLL,sigma)
eq = [d1==0,d2==0]; eq
```

evaluate

$$\left[\frac{\mu n - s_1}{\sigma^2} = 0, \frac{n}{\sigma} - \frac{\mu^2 n - 2\,\mu s_1 + s_2}{\sigma^3} = 0\right]$$

```
# Solution is a list of lists
sol1 = solve(eq,[mu,sigma]); sol1
```

evaluate

$$\left[\left[\mu = \frac{s_1}{n}, \sigma = -\frac{\sqrt{n s_2 - s_1^2}}{n}\right], \left[\mu = \frac{s_1}{n}, \sigma = \frac{\sqrt{n s_2 - s_1^2}}{n}\right]\right]$$

Notice that there is only one solution for $\mu$; it's $\mu = \frac{s_1}{n} = \overline{x}$. But there are two solutions for $\sigma$; they simplify to plus and minus the sample standard deviation (with $n$ rather than $n - 1$ in the denominator).

Of course we discard the negative solution because it's outside the parameter space, but this illustrates a feature of Sagemath that can be easy to forget. It doesn't know as

[5] The fact that the sufficient statistic has the same dimension as the parameter suggests that we will live happily ever after.

much about the problem as you do. Not only does it not know that variances can't be negative, it does not know that the quantity under the square root sign has to be positive, or even that all the symbols represent real numbers rather than complex numbers. I tried playing around with `assume`, but to no avail. There were always two solutions. It's easy enough to get the one we want. It's element one in the list of lists – the second one.

```
# Extract the second list of solutions
sol1[1]
```

evaluate

$$\left[\mu = \frac{s_1}{n}, \sigma = \frac{\sqrt{ns_2 - s_1^2}}{n}\right]$$

Later, it will be handy to evaluate the parameter vector at the vector of MLEs. So, this time, get the solution in the form of a dictionary (exactly like a Python dictionary). Actually, `solve` returns a *list* of dictionaries, and we want the second one.

```
# This time, get the solutions in the form of a LIST of dictionaries.
#  Save item one, the second one. (Indices begin with zero, not one.)
mle = solve(eq,[mu,sigma],solution_dict=True)[1]; mle
```

evaluate

$$\left\{\sigma : \frac{\sqrt{ns_2 - s_1^2}}{n}, \mu : \frac{s_1}{n}\right\}$$

```
# Refer to the elements of a dictionary using the keys.
mle[mu]  # MLE of mu
```

evaluate

$$\frac{s_1}{n}$$

For this particular case, it's not hard to show by elementary methods that the likelihood function attains its maximum at the sample mean and standard deviation, rather than a minimum or saddle point. But the general method is of interest. For a function $g(\theta_1, \ldots, \theta_t)$, define the *Hessian* as the $t \times t$ matrix of mixed partial derivatives whose $i, j$ element is

$$\frac{\partial^2 g}{\partial \theta_i \partial \theta_j}. \tag{B.1}$$

If the eigenvalues of the Hessian are all positive at a critical point, the function is concave up there. If they are all negative, it's concave down. If some are positive and some are negative, it's a saddle point.

In `Sagemath`, functions have a built-in Hessian attribute, but unfortunately, it applies to *all* symbolic variables. So `mLL.hessian()` returns a $5 \times 5$ matrix, corresponding to $(\mu, n, s_1, s_2, \sigma)$, in alphabetical order. And `mLL.hessian([mu,sigma])` (which is natural, and similar to expressions that work with gradients and Jacobians) yields

`TypeError: hessian() takes no arguments (1 given)`. So we'll construct the Hessian from scratch. Start by making an empty matrix that will be filled with partial derivates. It's critical that the matrix be of the right *type* (symbolic). Also, note that a lot of burdensome High School algebra is avoided by the quiet use of `factor` in the calculations below.

```
# H will be hessian of MINUS log likelihood
H = identity_matrix(SR,2); H # SR is the Symbolic Ring
```

evaluate

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

```
# Fill it with mixed partial derivatives
H[0,0] = derivative(mLL,mu,2); H[0,1] = derivative(mLL,[mu,sigma])
H[1,0] = H[0,1]              ; H[1,1] = derivative(mLL,sigma,2)
H = factor(H); H
```

evaluate

$$\begin{pmatrix} \frac{n}{\sigma^2} & -2\,\frac{(\mu n - s_1)}{\sigma^3} \\ -2\,\frac{(\mu n - s_1)}{\sigma^3} & \frac{(3\,\mu^2 n - n\sigma^2 - 6\,\mu s_1 + 3\,s_2)}{\sigma^4} \end{pmatrix}$$

```
# Evaluate at mle
hmle = factor(H(mle)); hmle
```

evaluate

$$\begin{pmatrix} \frac{n^3}{(n s_2 - s_1^2)} & 0 \\ 0 & 2\,\frac{n^3}{(n s_2 - s_1^2)} \end{pmatrix}$$

```
# Function is concave up at critical point iff all eigenvalues > 0 there.
hmle.eigenvalues()
```

evaluate

$$\left[\frac{n^3}{(n s_2 - s_1^2)}, 2\,\frac{n^3}{(n s_2 - s_1^2)}\right]$$

The denominator of both eigenvalues equals

$$n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2 = n\sum_{i=1}^{n}(x_i - \overline{x})^2,$$

so both eigenvalues are positive and the minus log likelihood is concave up at the MLE.

**The Multinomial Distribution** The multinomial distribution is based on a statistical experiment in which one of $k$ outcomes occurs, with probability $\theta_j, j = 1, \ldots, k$, where $\sum_{j=1}^{k} \theta_j = 1$. For example, consumers might be asked to smell six perfumes, and indicate which one they like most. The probability of preferring perfume $j$ is $\theta_j$, for $j = 1, \ldots, 6$.

The likelihood function may be written in terms of multinomial random vectors made up of $k$ indicators random variables: For case $i$, $x_{ij} = 1$ if event $j$ occurs, and zero otherwise. $\sum_{j=1}^{k} x_{ij} = 1$. The likelihood function is

$$\begin{aligned} L(\boldsymbol{\theta}) &= \prod_{i=1}^{n} \theta_1^{x_{i,1}} \theta_2^{x_{i,2}} \cdots \theta_k^{x_{i,k}} \\ &= \theta_1^{\sum_{i=1}^{n} x_{i,1}} \theta_2^{\sum_{i=1}^{n} x_{i,2}} \cdots \theta_k^{\sum_{i=1}^{n} x_{i,k}}. \end{aligned}$$

Using $x_j$ to represent the sum $\sum_{i=1}^{n} x_{i,j}$, the likelihood may be expressed in a non-redundant way in terms of $k-1$ parameters and $k-1$ sufficient statistics, as follows:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \theta_1^{x_1} \theta_2^{x_2} \cdots \theta_k^{x_k} \\ &= \theta_1^{x_1} \cdots \theta_{k-1}^{x_{k-1}} \left(1 - \sum_{j=1}^{k-1} \theta_j\right)^{n - \sum_{j=1}^{k-1} x_j}. \end{aligned}$$

Here's an example with $k = 6$ (six perfumes).

```
# Multinomial Maximum likelihood - 6 categories
var('theta1 theta2 theta3 theta4 theta5  x1 x2 x3 x4 x5 n')
theta = [theta1, theta2, theta3, theta4, theta5]
LL = x1*log(theta1) + x2*log(theta2) + x3*log(theta3) +
x4*log(theta4)  + x5*log(theta5) +
(n-x1-x2-x3-x4-x5)*log(1-theta1-theta2-theta3-theta4-theta5)
LL
```

evaluate

$$(n - x_1 - x_2 - x_3 - x_4 - x_5) \log(-\theta_1 - \theta_2 - \theta_3 - \theta_4 - \theta_5 + 1) + x_1 \log(\theta_1) + x_2 \log(\theta_2) + x_3 \log(\theta_3) + x_4 \log(\theta_4) + x_5 \log(\theta_5)$$

Instead of calculating all five partial derivatives, it's easier to request the gradient – which is the same thing. Then we loop through the element of the gradient list, setting each derivative to zero, displaying the equation, and appending it to a list of equations that need to be solved. Notice the use of the colon (:) and indentation for looping. `Sagemath` shares this syntax with `Python`.

```
# Gradient is zero at MLE. It's a tuple, not a list.
gr = LL.gradient(theta)
# Setting the derivatives to zero ...
eq = [] # Start with empty list
for a in gr :
    equation = (a==0)
    show(equation)  # Display the equation
    eq.append(equation) # Append equation to list eq.
```

evaluate

$$\frac{n-x_1-x_2-x_3-x_4-x_5}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} + \frac{x_1}{\theta_1} = 0$$

$$\frac{n-x_1-x_2-x_3-x_4-x_5}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} + \frac{x_2}{\theta_2} = 0$$

$$\frac{n-x_1-x_2-x_3-x_4-x_5}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} + \frac{x_3}{\theta_3} = 0$$

$$\frac{n-x_1-x_2-x_3-x_4-x_5}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} + \frac{x_4}{\theta_4} = 0$$

$$\frac{n-x_1-x_2-x_3-x_4-x_5}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} + \frac{x_5}{\theta_5} = 0$$

Now we will solve for $\theta_1, \ldots, \theta_5$. While it's true that the Sagemath calculation is specific to $k = 6$ categories, the list of equations to solve makes the pattern clear, and points the way to a general solution. Here is the specific solution:

```
# Get the solutions in the form of a LIST of dictionaries.
# Dictionary items are not in any particular order.
# Save item zero, the first dictionary.
ThetaHat = solve(eq,theta,solution_dict=True)[0]
ThetaHat  # The mean (vector)
```

evaluate

$$\left\{\theta_3 : \frac{x_3}{n}, \theta_2 : \frac{x_2}{n}, \theta_1 : \frac{x_1}{n}, \theta_5 : \frac{x_5}{n}, \theta_4 : \frac{x_4}{n}\right\}$$

So for $j = 1, \ldots, 5$, the MLE is $\widehat{\theta}_j = \frac{\sum_{i=1}^{n} x_{ij}}{n} = \overline{x}_j$, or the sample proportion. There's little doubt that this is really where the likelihood function achieves its maximum, and not a minimum or saddle point. But it's instructive to check. Here is the Hessian of the minus log likelihood.

```
# Is it really the maximum?
 # H will be hessian of MINUS log likelihood
 H = identity_matrix(SR,5) # SR is the Symbolic Ring
for i in interval(0,4) :
    for j in interval(0,i) :
        H[i,j] = derivative(-LL,[theta[i],theta[j]])
        H[j,i] = H[i,j] # It's symmetric
H
```

evaluate

$$\left(\begin{array}{ccccc}
\frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2}+\frac{x_1}{\theta_1^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} \\
\frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2}+\frac{x_2}{\theta_2^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} \\
\frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2}+\frac{x_3}{\theta_3^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} \\
\frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2}+\frac{x_4}{\theta_4^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} \\
\frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & \frac{n-x_1-x_2-x_3-x_4-x_5}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2}+\frac{x_5}{\theta_5^2}
\end{array}\right)$$

All its eigenvalues should be positive at the critical point where the derivates simultaneously equal zero.

```
# Evaluate at critical point
Hmle = factor(H(ThetaHat)); Hmle
```

evaluate

$$\left(\begin{array}{ccccc}
\frac{(n-x_2-x_3-x_4-x_5)n^2}{(n-x_1-x_2-x_3-x_4-x_5)x_1} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} \\
\frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{(n-x_1-x_3-x_4-x_5)n^2}{(n-x_1-x_2-x_3-x_4-x_5)x_2} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} \\
\frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{(n-x_1-x_2-x_4-x_5)n^2}{(n-x_1-x_2-x_3-x_4-x_5)x_3} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} \\
\frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{(n-x_1-x_2-x_3-x_5)n^2}{(n-x_1-x_2-x_3-x_4-x_5)x_4} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} \\
\frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{(n-x_1-x_2-x_3-x_4)n^2}{(n-x_1-x_2-x_3-x_4-x_5)x_5}
\end{array}\right)$$

```
# Concave up iff all eigenvalues > 0
Hmle.eigenvalues()
```

evaluate

```
Traceback (click to the left of this block for traceback)
...
ArithmeticError: could not determine eigenvalues exactly using symbolic
matrices; try using a different type of matrix via self.change_ring(),
if possible
```

It seems that Sagemath cannot solve for the eigenvalues symbolically. A *numerical* solution for a particular set of sample data would be routine. But there is another way out. A real symmetric matrix has all positive eigenvalues if and only if it's positive definite.

And *Sylvester's Criterion*[6] is a necessary and sufficient condition for a real symmetric matrix to be positive definite. A *minor* of a matrix is the determinant of a square submatrix that is formed by deleting selected rows and columns from the original matrix. The *principal minors* of a square matrix are the determinants of the upper left $1 \times 1$ matrix, the upper left $2 \times 2$ matrix, and so on. Sylvester's Criterion says that the matrix is positive definite if and only if all the principal minors are positive.

Here, there are five determinants to evaluate, one of which is just the upper left matrix element. We'll do it in a loop. The `submatrix(h,i,j,k)` attribute returns the submatrix starting in row $h$ and column $i$, consisting of $j$ rows and $k$ columns. As usual, index numbering starts with zero. For full documentation, try something like `Hmle.submatrix?`

```
Hmle.submatrix(0,0,2,2) # Upper left 2x2, just to see
```

evaluate

$$\begin{pmatrix} \frac{(n-x_2-x_3-x_4-x_5)n^2}{(n-x_1-x_2-x_3-x_4-x_5)x_1} & \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} \\ \frac{n^2}{n-x_1-x_2-x_3-x_4-x_5} & \frac{(n-x_1-x_3-x_4-x_5)n^2}{(n-x_1-x_2-x_3-x_4-x_5)x_2} \end{pmatrix}$$

```
# Calculate and display determinants
for j in interval(1,5) :
    show(Hmle.submatrix(0,0,j,j).determinant().factor())
```

evaluate

$$\frac{(n-x_2-x_3-x_4-x_5)n^2}{(n-x_1-x_2-x_3-x_4-x_5)x_1}$$

$$\frac{(n-x_3-x_4-x_5)n^4}{(n-x_1-x_2-x_3-x_4-x_5)x_1x_2}$$

$$\frac{(n-x_4-x_5)n^6}{(n-x_1-x_2-x_3-x_4-x_5)x_1x_2x_3}$$

$$\frac{(n-x_5)n^8}{(n-x_1-x_2-x_3-x_4-x_5)x_1x_2x_3x_4}$$

$$\frac{n^{11}}{(n-x_1-x_2-x_3-x_4-x_5)x_1x_2x_3x_4x_5}$$

Assuming the sample size is large enough so that there's at least one observation in each category, these quantities are obviously all positive. You can also see that while `Sagemath` performs calculations that are very specific to the problem at hand, the answers can reveal regular patters that could be exploited in something like a proof by induction. And the effort involved is tiny, compared to doing it by hand.

Incidentally, the `submatrix` function can be used to obtain Hessians a bit more easily. Recall that `Sagemath` functions have a `hessian` attribute, but it's calculated with respect to *all* the variables, which is never what you want for likelihood calculations. But the rows and columns are in alphabetical order, which in the present case is $n, \theta_1, \ldots, \theta_5, x_1, \ldots, x_5$.

[6] The Wikipedia has a nice article on this, including a formal proof. See http://www.en.wikipedia.org/.

So the $5 \times 5$ Hessian we want is easy to extract. Check and see if it's what we calculated earlier in a double loop.

```
-LL.hessian().submatrix(1,1,5,5) == H
```

evaluate

True

Ho Ho!

### Fisher Information

There are many places in mathematical Statistics where `Sagemath` can save a lot of tedious calculation. One of these is in conjunction with *Fisher Information* (See Appendix A for some discussion). For a model with parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_t)'$, the Fisher information matrix is a $t \times t$ matrix $I(\boldsymbol{\theta})$ whose $(i, j)$ element is

$$-E\left(\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log f(X|\boldsymbol{\theta})\right).$$

This is the information about $\boldsymbol{\theta}$ in a single observation. The information in $n$ independent and identically distributed observations is $n\, I(\boldsymbol{\theta})$. Under some regularity conditions that amount to smoothness of the functions involved, the vector of MLEs is approximately multivariate normal for large samples, with mean $\boldsymbol{\theta}$ and covariance matrix $(n\, I(\boldsymbol{\theta}))^{-1}$. This is a source of large-sample tests and confidence intervals.

**The Univariate Normal Distribution** Comparing the formula for the Fisher Information to Expression (B.1), it is clear that the Fisher information is just the expected value of the Hessian of the minus log density[7]. We'll start by calculating the Hessian. The last line says "Take minus the log of $f(X)$, calculate the Hessian, extract the $2 \times 2$ matrix with upper left entry $(1, 1)$, and factor it. Then put the result in `h`; display `h`." In this case and many others, factoring yields a lot of simplification.

```
# Normal
var('mu, sigma, X, n'); assume(sigma>0)
f(x) = 1/(sigma*sqrt(2*pi)) * exp(-(x-mu)^2/(2*sigma^2))
# Extract lower right 2x2 of Hessian of minus log density
# That is, of Hessian with respect to X, mu, sigma.
# X is alphabetically first because it's capitalized.
h = -log(f(X)).hessian().submatrix(1,1,2,2).factor(); h
```

evaluate

[7] The Hessian reflects *curvature* of the function. Fisher's insight was that the greater the curvature of the log likelihood function at the true parameter value, the more information the data provide about the parameter. Further discussion of the connection between the Hessian and the Fisher Information may be found in Appendix A.

$$\begin{pmatrix} \frac{1}{\sigma^2} & \frac{2\,(X-\mu)}{\sigma^3} \\ \frac{2\,(X-\mu)}{\sigma^3} & \frac{3\,X^2-6\,X\mu+3\,\mu^2-\sigma^2}{\sigma^4} \end{pmatrix}$$

Now take the expected value. In the lower right we'll directly integrate, though it could also be done by substituting in known quantities and then simplifying. The other cells can be done by inspection.

```
# Fisher information in one observation is expected h
info = h
info[0,1]=0; info[1,0]=0 # Because E(X)=mu
info[1,1] = integrate(info[1,1]*f(X),X,-oo,oo)
info
```

evaluate

$$\begin{pmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{2}{\sigma^2} \end{pmatrix}$$

That's the Fisher Information in one observation. To get the asymptotic (approximate, for large $n$) covariance matrix, multiply by $n$ and invert the matrix.

```
# Fisher info in n observations is n * info in one observation.
# MLEs are asymptotically multivariate normal with mean theta
# and variance-covariance matrix the inverse of the Fisher info.
avar = (n*info).inverse(); avar
```

evaluate

$$\begin{pmatrix} \frac{\sigma^2}{n} & 0 \\ 0 & \frac{\sigma^2}{2\,n} \end{pmatrix}$$

That's a standard example that can be done by hand, though perhaps it's a little unusual because the model is parameterized in terms of the standard deviation rather than the variance. This next one, however, would be fearsome to do by hand.

**The Multinomial Distribution** We'll stay with the case of six categories. Now, because the MLE equals the sample mean vector in this case, the multivariate Central Limit Theorem (see Appendix A) can be used directly without going through the Fisher Information. We'll do it this way first, because it's a good way to check `Sagemath`'s final answer.

The multivariate Central Limit Theorem says that if $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are i.i.d. random vectors with expected value vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Then $\sqrt{n}(\overline{\mathbf{X}}_n - \boldsymbol{\mu})$ converges in distribution to a multivariate normal with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$. That is, for large $n$, $\overline{\mathbf{X}}_n$ has a distribution that is approximately multivariate normal, with mean $\boldsymbol{\mu}$ and covariance matrix $\frac{1}{n}\boldsymbol{\Sigma}$.

Here, each of the i.i.d. random vectors is filled with $k - 1 = 5$ zeros and possibly a single 1 , with the number 1 indicating which event occurred. If all five entries of

$\mathbf{X}_i$ equal zero, then the sixth event occurred. The marginal distributions are Bernoulli, so $E(X_{i,j}) = \theta_j$ and $\boldsymbol{\mu} = (\theta_1, \ldots, \theta_5)'$. The variances are $Var(X_{i,j}) = \theta_j(1-\theta_j)$, for $j = 1, \ldots, 5$. Since, $Pr\{X_{i,j}X_{i,m} = 0\}$ for $j \neq m$, $E(X_{i,j}X_{i,m}) = 0$, and

$$\begin{aligned} Cov(X_{i,j}X_{i,m}) &= E(X_{i,j}X_{i,m}) - E(X_{i,j})E(X_{i,m}) \\ &= -\theta_j\theta_m. \end{aligned}$$

So by the Central Limit Theorem, the asymptotic mean of the MLE is $\boldsymbol{\mu} = (\theta_1, \ldots, \theta_5)'$, and the asymptotic covariance matrix is

$$\frac{1}{n}\boldsymbol{\Sigma} = \begin{pmatrix} \frac{\theta_1(1-\theta_1)}{n} & -\frac{\theta_1\theta_2}{n} & -\frac{\theta_1\theta_3}{n} & -\frac{\theta_1\theta_4}{n} & -\frac{\theta_1\theta_5}{n} \\ -\frac{\theta_1\theta_2}{n} & \frac{\theta_2(1-\theta_2)}{n} & -\frac{\theta_2\theta_3}{n} & -\frac{\theta_2\theta_4}{n} & -\frac{\theta_2\theta_5}{n} \\ -\frac{\theta_1\theta_3}{n} & -\frac{\theta_2\theta_3}{n} & \frac{\theta_3(1-\theta_3)}{n} & -\frac{\theta_3\theta_4}{n} & -\frac{\theta_3\theta_5}{n} \\ -\frac{\theta_1\theta_4}{n} & -\frac{\theta_2\theta_4}{n} & -\frac{\theta_3\theta_4}{n} & \frac{\theta_4(1-\theta_4)}{n} & -\frac{\theta_4\theta_5}{n} \\ -\frac{\theta_1\theta_5}{n} & -\frac{\theta_2\theta_5}{n} & -\frac{\theta_3\theta_5}{n} & -\frac{\theta_4\theta_5}{n} & \frac{\theta_5(1-\theta_5)}{n} \end{pmatrix} \tag{B.2}$$

To compare this to what we get from the likelihood approach, first calculate the Hessian of the minus log probability mass function.

```
# Multinomial - 6 categories again
var('theta1 theta2 theta3 theta4 theta5  X1 X2 X3 X4 X5  n')
Lp = X1*log(theta1) + X2*log(theta2) + X3*log(theta3)
+ X4*log(theta4) + X5*log(theta5) + (1-X1-X2-X3-X4-X5)
* log(1-theta1-theta2-theta3-theta4-theta5)
h = -Lp.hessian().submatrix(5,5,5,5); h
```

evaluate

$$\begin{pmatrix} -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} + \frac{X_1}{\theta_1^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} \\ -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} + \frac{X_2}{\theta_2^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X}{(\theta_1+\theta_2+\theta_3} \\ -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} + \frac{X_3}{\theta_3^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X}{(\theta_1+\theta_2+\theta_3} \\ -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X}{(\theta_1+\theta_2+\theta_3} \\ -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X_3+X_4+X_5-1}{(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1)^2} & -\frac{X_1+X_2+X}{(\theta_1+\theta_2+\theta_3} \end{pmatrix}$$

Sometimes, `Sagemath` output runs off the right side of the screen and you have to scroll to see it all. In this document, it just gets chopped off. But you can still see that all the $X_j$ quantities appear in the numerator, and taking the expected values would be easy by hand.

```
# Computing expected values is just substituting theta_j for X_j
info = h(X1=theta1,X2=theta2,X3=theta3,X4=theta4,X5=theta5)
info
```

evaluate

$$\begin{pmatrix} -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1}+\frac{1}{\theta_1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} \\ -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1}+\frac{1}{\theta_2} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} \\ -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1}+\frac{1}{\theta_3} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} \\ -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1}+\frac{1}{\theta_4} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} \\ -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} & -\frac{1}{\theta_1+\theta_2+\theta_3+\theta_4+\theta_5-1} \end{pmatrix}$$

The asymptotic covariance matrix is obtained by multiplying by $n$ and taking the inverse. Inverting the matrix by hand is possible, but it would be a brutal experience. With `Sagemath`, it takes a few seconds, including the typing.

```
# Asymptotic covariance matrix
avar = (n*info).inverse().factor(); avar
```

evaluate

$$\begin{pmatrix} \frac{(\theta_1-1)\theta_1}{n} & \frac{\theta_1\theta_2}{n} & \frac{\theta_1\theta_3}{n} & \frac{\theta_1\theta_4}{n} & \frac{\theta_1\theta_5}{n} \\ \frac{\theta_1\theta_2}{n} & \frac{(\theta_2-1)\theta_2}{n} & \frac{\theta_2\theta_3}{n} & \frac{\theta_2\theta_4}{n} & \frac{\theta_2\theta_5}{n} \\ \frac{\theta_1\theta_3}{n} & \frac{\theta_2\theta_3}{n} & \frac{(\theta_3-1)\theta_3}{n} & \frac{\theta_3\theta_4}{n} & \frac{\theta_3\theta_5}{n} \\ \frac{\theta_1\theta_4}{n} & \frac{\theta_2\theta_4}{n} & \frac{\theta_3\theta_4}{n} & \frac{(\theta_4-1)\theta_4}{n} & \frac{\theta_4\theta_5}{n} \\ \frac{\theta_1\theta_5}{n} & \frac{\theta_2\theta_5}{n} & \frac{\theta_3\theta_5}{n} & \frac{\theta_4\theta_5}{n} & \frac{(\theta_5-1)\theta_5}{n} \end{pmatrix}$$

This is the same as Expression B.2, which came from the Central Limit Theorem. It's an unqualified success.

### Taylor Expansions

There are many versions of Taylor's Theorem. Here is a useful one. Let the $n$th derivative $f^{(n)}$ of the function $f(x)$ be continuous in $[a, b]$ and differentiable in $(a, b)$, with $x$ and $x_0$ in $(a, b)$. Then there exists a point $\xi$ between $x$ and $x_0$ such that

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)(x - x_0)^2}{2!} + \ldots + \frac{f^{(n)}(x_0)(x - x_0)^n}{n!} \\ &+ \frac{f^{(n+1)}(\xi)(x - x_0)^{n+1}}{(n + 1)!} \end{aligned} \tag{B.3}$$

where $R_n = \frac{f^{(n+1)}(\xi)(x-x_0)^{n+1}}{(n+1)!}$ is called the *remainder term*. If $R_n \to 0$ as $n \to \infty$, the resulting infinite series is called the *Taylor Series* for $f(x)$.

In certain styles of applied statistics, when people are having trouble with a function, they approximate it by just taking the first two or three terms of a Taylor expansion, and discarding the remainder. Sometimes, the approximation can be quite helpful. Consider, for example, a simple[8] logistic regression in which a linear model for the log odds of $Y = 1$ leads to

$$Pr\{Y = 1|X = x\} = E(Y|X = x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}.$$

Under this model, what is the covariance between $X$ and $Y$? It's easy to wonder, but not easy to calculate. Suppose $X$ has a distribution with expected value $\mu$ and variance $\sigma^2$.

[8] One explanatory variable.

Perhaps $X$ is normal. Let's use the formula $Cov(X,Y) = E(XY) - E(X)E(Y)$, and try double expectation. That is,

$$\begin{aligned} E[Y] &= E[E(Y|X)] \\ &= \int_{-\infty}^{\infty} E(Y|X=x)\, f(x)\, dx \\ &= \int_{-\infty}^{\infty} \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}\, f(x)\, dx. \end{aligned} \tag{B.4}$$

If $X$ is normal, I certainly can't do this integral. I have tried many times and failed. `Sagemath` can't do it either. Details are omitted.

Let's approximate $g(X) = E(Y|X)$ with the first few terms of a Taylor series. Then it's easier to work with. Note that you can find out what atributes the function $g$ has with `print(dir(g))`, and then get details about the `taylor` attribute with `g.taylor?` .

```
# Cov(X,Y) for logistic regression (Taylor)

var('X beta0 beta1 mu sigma')
g = exp(beta0 + beta1*X)/(1+exp(beta0 + beta1*X))
# print(dir(g))
# g.taylor?
t1 = g.taylor(X,mu,2); t1 # Expand function of X about mu, degree 2 (3 terms)
```

evaluate

$$\frac{(X-\mu)\beta_1 e^{(\beta_1\mu+\beta_0)}}{2\,e^{(\beta_1\mu+\beta_0)}+e^{(2\,\beta_1\mu+2\,\beta_0)}+1} + \frac{(X-\mu)^2\left(\beta_1^2 e^{(\beta_1\mu+\beta_0)}-\beta_1^2 e^{(2\,\beta_1\mu+2\,\beta_0)}\right)}{2\left(3\,e^{(\beta_1\mu+\beta_0)}+3\,e^{(2\,\beta_1\mu+2\,\beta_0)}+e^{(3\,\beta_1\mu+3\,\beta_0)}+1\right)} + \frac{e^{(\beta_1\mu+\beta_0)}}{e^{(\beta_1\mu+\beta_0)}+1}$$

Taking the expected value with respect to $X$ will cause the first term to disappear, and replace $(X-\mu)^2$ with $\sigma^2$ in the second term. We'll integrate with respect to the normal distribution, but that's just for convenience. Any distribution with expected value $\mu$ and variance $\sigma^2$ would yield the same result.

```
# Use normal to take expected value Just a convenience :
f = 1/(sigma*sqrt(2*pi)) * exp(-(X-mu)^2/(2*sigma^2))
assume(sigma>0)
EY = (t1*f).integrate(X,-oo,oo).factor(); EY
```

evaluate

$$-\frac{\left(\beta_1^2\sigma^2 e^{(\beta_1\mu+\beta_0)}-\beta_1^2\sigma^2-4\,e^{(\beta_1\mu+\beta_0)}-2\,e^{(2\,\beta_1\mu+2\,\beta_0)}-2\right)e^{(\beta_1\mu+\beta_0)}}{2\left(e^{(\beta_1\mu+\beta_0)}+1\right)^3}$$

That's pretty messy, but maybe there will be some simplification when we calculate $Cov(X,Y) = E(XY) - E(X)E(Y)$. First we need an approximation of $E(XY)$.

```
# Double expectation for E(XY) - First, approximate XE(Y|X)
t2 = (X*g).taylor(X,mu,2); t2 # Looks pretty hairy
EXY = (t2*f).integrate(X,-oo,oo).factor(); EXY
```

evaluate

$$-\frac{\left(\beta_1^2\mu\sigma^2 e^{(\beta_1\mu+\beta_0)}-\beta_1^2\mu\sigma^2-2\,\beta_1\sigma^2 e^{(\beta_1\mu+\beta_0)}-2\,\beta_1\sigma^2-4\,\mu e^{(\beta_1\mu+\beta_0)}-2\,\mu e^{(2\,\beta_1\mu+2\,\beta_0)}-2\,\mu\right)e^{(\beta_1\mu+\beta_0)}}{2\left(e^{(\beta_1\mu+\beta_0)}+1\right)^3}$$

```
# Finally, approximate the covariance
Cov = (EXY-mu*EY).factor(); Cov
```

evaluate

$$\frac{\beta_1\sigma^2 e^{(\beta_1\mu+\beta_0)}}{\left(e^{(\beta_1\mu+\beta_0)}+1\right)^2}$$

Well, you have to admit that's nice! Some of the intermediate steps were fiercely complicated, but the final result is clean and simple. Sagemath has saved us a lot of unpleasant work. Furthermore, the result makes sense because the sign of the covariance is the same as the sign of $\beta_1$, as it should be.

However, *we really don't know if it's a good approximation or not.* That's right. Taylor expansions are more accurate closer to the point about which you expand the function, and they are more accurate the more terms you take. Beyond that, it's generally unknown, unless you have more information (like perhaps the remainder you've discarded approaches zero as the sample size increases, or something).

So we need to investigate it a bit more, and the easiest thing to do is to try some numerical examples. With specific numbers for the parameters, Sagemath will be able to calculate $E(Y)$ and $E(XY)$ by numerical integration. First, we'll try $\mu = 0, \sigma = 2, \beta_0 = 0, \beta_1 = 1$. The approximation is

```
# Example 1, with  mu=0,beta0=0,sigma=2,beta1=1
Cov(mu=0,beta0=0,sigma=2,beta1=1)
```

evaluate

1

The calculation of $Cov(X,Y) = E(XY)$ by double expectation is similar to (B.4).

$$\begin{aligned} E[XY] &= E[E(XY|X)] \\ &= \int_{-\infty}^{\infty} E(XY|X=x)\, f(x)\, dx \\ &= \int_{-\infty}^{\infty} E(xY|X=x)\, f(x)\, dx \\ &= \int_{-\infty}^{\infty} x\, E(Y|X=x)\, f(x)\, dx \\ &= \int_{-\infty}^{\infty} x\, \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}\, f(x)\, dx. \end{aligned} \tag{B.5}$$

In the material below, the result of `show(EXY1)` tells us that $E(XY)$, though it's simplified a bit, is an integral that Sagemath cannot take any farther, even with specific numerical

values. Then, `EXY1.n()` says please evaluate it numerically. The numerical evaluation attribute, in the case of an integral, is a sophisticated numerical integration algorithm.

```
# This will be the covariance, since mu=0
EXY1 = (X*g*f)(mu=0,beta0=0,sigma=2,beta1=1).integrate(X,-oo,oo)
show(EXY1)
EXY1.n()
```

evaluate

$$\frac{\sqrt{2}\int_{-\infty}^{+\infty}\frac{Xe^{\left(-\frac{1}{8}X^2+X\right)}}{e^X+1}\,dX}{4\sqrt{\pi}}$$

0.605705509602159

That's not too promising. Is the approximation really this bad? While `Sagemath` is extremely accurate compared to almost any human being, mistakes in the input can cause big problems. Typos are the main source of trouble, but misunderstandings are possible too, and the results can be even worse. So, when a result is a bit surprising like this, it's important to cross-check it somehow. Let's try a simulation with `R`. The idea is to first simulate a large collection of $X$ values from a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 2$, calculate $Pr\{Y = 1|X_i\}$, using $\beta_0 = 0$ and $\beta_1 = 1$. Finally, generate binary $Y$ values using those probabilities, and calculate the sample covariance. By the Strong Law of Large Numbers, the probability equals one that the sample covariance approaches the true covariance as $n \to \infty$, like an ordinary limit. So with a very large $n$, we'll get a good approximation of $Cov(X, Y)$. Is it closer to 1, or 0.606? Here is the `R` calculation, without further comment.

```
> n = 100000; mu=0; beta0=0; sigma=2; beta1=1
> x = rnorm(n,mu,sigma)
> xb = beta0 + beta1*x
> p = exp(xb)/(1+exp(xb))
> y = rbinom(n,1,p)
> var(cbind(x,y))
          x         y
x 3.9687519 0.6039358
y 0.6039358 0.2499991
```

Now we can be confident that the numerical integration (and the double expectation reasoning behind it) produced correct results, and the Taylor series approximation was poor. It can easily get worse. For example, with $\mu = 1, \sigma = 10, \beta_0 = 1, \beta_1 = 1$, the Taylor series approximation of the covariance is 10.499, while the correct answer by numerical integration is 3.851.

The story has a two-part moral. Taylor series approximations are easy with `Sagemath`, but whether they are accurate enough to be useful is another matter. This point is some-

times overlooked in applied Statistics. To be clear, this is not a problem with Sagemath; the problem is with the practice of blindly linearizing everything.

To leave a better taste about Taylor series approximations, let $X_1, \ldots, X_n$ be a random sample from a Bernoulli distribution, with $Pr\{X_i = 1\} = \theta$. A quantity that is useful in categorical data analysis is the *log odds*:

$$\text{Log Odds} = \log \frac{\theta}{1-\theta},$$

where log refers to the natural logarithm.

The best estimator of $\theta$ is the sample proportion: $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. The log odds is estimated by

$$Y = \log \frac{\overline{X}}{1-\overline{X}}.$$

The variance of $\overline{X}$ is $\frac{\theta(1-\theta)}{n}$, but what is the variance of the estimated log odds $Y$? As we shall see, it's possible to give an exact answer for any given $n$, but the expression is very complicated and hard to use in later calculations.

Instead, for any statistic $T_n$ that estimates $\theta$, and any differentiable function $g(t)$ (of which $g(t) = \log \frac{t}{1-t}$ is an example), expand $g(t)$ about $\theta$, taking just the first two terms of a Taylor expansion (see Expression B.3) and discarding the remainder. Then

$$\begin{aligned} Var\left(g(T_n)\right) &\approx Var\left(g(\theta) + g'(\theta)(T_n - \theta)\right) \\ &= 0 + g'(\theta)^2 Var(T_n) + 0 \\ &= g'(\theta)^2 Var(T_n). \end{aligned} \tag{B.6}$$

The only reason for making $T_n$ a statistic that estimates $\theta$ is so it will be reasonable to expand $g(t)$ about $\theta$. Actually, $T_n$ could be any random variable and $\theta$ could be any real number, but in that case the approximation could be arbitrarily bad.

Formula (B.6) for the variance of a function is quite general. We don't need taylor; instead, we'll just use Sagemath to take the derivative, square it, multiply by the variance of $T_n$, and simplify.

```
# Variance of log odds
var('n theta')
g = log(theta/(1-theta))
vTn = theta*(1-theta)/n
v = ( g.derivative(theta)^2 * vTn ).factor(); v
```

evaluate

$$-\frac{1}{(\theta-1)n\theta}$$

Let's try a numerical example, with $\theta = 0.1$ and $n = 200$.

```
v(theta=0.1,n=200)
```

evaluate

0.0555555555555556

Is this a good approximation? We certainly can't take it for granted. Now, for any fixed $n$, the random variable $\overline{X}_n$ (also known as $T_n$) is just $\frac{X}{n}$, where $X$ is binomial with parameters $n$ and $\theta$. So,

$$\begin{aligned} Y = Y(X) &= \log \frac{\overline{X}}{1-\overline{X}} \\ &= \log \frac{X/n}{1-X/n} \\ &= \log \frac{X}{n-X}, \end{aligned}$$

and we can calculate

$$\begin{aligned} E(Y) &= \sum_{x=0}^{n} y(x) Pr\{X=x\} \\ &= \sum_{x=0}^{n} \log\left(\frac{X}{n-X}\right) Pr\{X=x\} \\ &= \sum_{x=0}^{n} \log\left(\frac{X}{n-X}\right) \binom{n}{x} \theta^x (1-\theta)^{n-x}. \end{aligned}$$

The calculation of $E(Y^2)$ is similar, and then $Var(Y) = E(Y^2) - [E(Y)]^2$.

Because we're actually going to do it (an insane proposition by hand), we notice that *the variance of the estimated log odds is not even defined* for any finite $n$. Everything falls apart for $x = 0$ and $x = n$.

Now in standard categorical data analysis, it assumed that $\theta$ is strictly between zero and one, and the sample size is large enough so that the events $X = 0$ and $X = n$ (whose probability goes to zero as $n \to \infty$ do not occur. In practice if they did occur, the statistician would move to a different technology. So, the variance we want is actually *conditional* on $1 \leq X \leq n-1$.

Adjusting $Pr\{X = x\}$ to make it a conditional probability involves dividing by $1 - Pr\{X = 0\} - Pr\{X = n\}$, which for $n = 200$ is a number extremely close to one. So will it be okay to just discard $x = 0$ and $x = n$ rather than really adjusting? Take a look at how small the probabilities are.

```
# Is it okay to just drop x=0 and x=200?
p(x) = n.factorial()/(x.factorial() * (n-x).factorial()) * theta^x * (1-theta)^(n-x)
p(0)(theta=0.1); p(200)(theta=0.1)
```

evaluate

$7.05507910865537 \times 10^{-10}$
$1.00000000000001 \times 10^{-200}$

Okay, we'll just sum from $x = 1$ to $x = n - 1$, and call it an "exact" calculation. In the Sagemath work below, note that because $n$ is so large, the binomial coefficient in $p(x)$ can be big enough to overflow the computer's memory, while at the same time the product of $\theta$ and $(1 - \theta)$ values can be small enough to underflow. To avoid the numerical inaccuracy that would come from this, $\theta$ is written as a ratio of two integers. Then inside the loop, $p(x)$ is evaluated by exact integer arithmetic and then factored, resulting in numerous cancellations so that the result is as accurate as possible before it is numerically evaluated and multiplied by the numerical version of $log\frac{x}{n-x}$. By the way, it's a *lot* faster to do it this way rather than doing the whole calculation symbolically and then numerically evaluating the final result.

```
# Calculate exactly, trying to minimize rounding error
y(x) = log(x/(n-x))
n=200; EY=0.0; EYsq=0.0
for x in interval(1,n-1) :
    EY = EY + y(x).n()*(p(x)(theta=1/10).factor().n())
    EYsq = EYsq + (y(x)^2).n()*(p(x)(theta=1/10).factor().n())
vxact = EYsq-EY^2; vxact
```

evaluate

0.0595418877731042

As a check on this, one can randomly generate a large number of Binomial$(n, \theta)$ pseudo-random numbers. Dividing each one by $n$ gives a random sample of $\overline{X}_n$ values, and then computing any function of the $\overline{X}_n$ values yields a collection of random variables that is a nice estimate of the sampling distribution of the statistic in question. With ten million Binomial$(n, \theta)$ values, this approach is used to approximate $Var\left(\log\left(\frac{\overline{X}_n}{1-\overline{X}_n}\right)\right)$.

```
> set.seed(9999)
> n = 200; theta = 0.1; m=10000000
> xbar = rbinom(m,n,theta)/n
> logodds = log(xbar/(1-xbar))
> var(logodds)
[1] 0.05955767
```

So the "exact" calculation is right, and the Taylor series approximation is pretty close. Is it a coincidence? No. By the Law of Large Numbers, the probability distribution of the sample proportion $\overline{X}_n$ becomes increasingly concentrated around $\theta$ as the sample size increases, so that within a tiny interval enclosing $\theta$, the linear approximation of $g(t)$ in (B.6) is very accurate in the neighbourhood where most of the probability distribution resides. As the sample size increases, it becomes even better, and the approximation of the variance becomes even better.

As a final note about Taylor series, Sagemath can easily calculate truncated Taylor series approximations of functions of several variables, in which derivatives are replaced by matrices of partial derivatives (Jacobians).

### Matrices and linear algebra

Sagemath is very good at matrix calculations with numbers, but Sagemath's ability to do matrix calculations with symbols is what makes it useful for structural equation modeling. The algorithm that Sagemath uses for a particular task will depend on the *ring* (a concept from Algebra) to which the matrix belongs. When the contents of a matrix are symbols, the matrix belongs to the symbolic ring, abbreviated SR. As in Python, a matrix is a list of rows, and the rows are lists of matrix elements.

```
var('alpha beta gamma delta')
A = matrix( SR, [[alpha, beta],[gamma, delta]] ); A
```

evaluate

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

Also as in Python, index numbering begins with zero, not one. This may be easy to forget.

```
A[0,1]
```

evaluate

$$\beta$$

Of course you need not be bound by this awkward convention, but in the following example you do need to remember that A[0,0] $= x_{11}$. By the way, I cannot figure out how to get nice-looking double subscripts separated by commas; I don't even know if it's possible. However, it's not a problem for small examples.

```
# Note the nice subscripts
var('x11 x12 x13 x21 x22 x23')
B = matrix(SR, [[x11, x12, x13], [x21, x22, x23]])
B
```

evaluate

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix}$$

Multiplication by a scalar does what you would hope.

```
a=2
a*A
```

evaluate

$$\begin{pmatrix} 2\,\alpha & 2\,\beta \\ 2\,\gamma & 2\,\delta \end{pmatrix}$$

Matrix multiplication also uses asterisks. Of course the matrices must be the right size or Sagemath raises an error.

```
C = A*B; C
```

evaluate

$$\begin{pmatrix} \alpha x_{11} + \beta x_{21} & \alpha x_{12} + \beta x_{22} & \alpha x_{13} + \beta x_{23} \\ \gamma x_{11} + \delta x_{21} & \gamma x_{12} + \delta x_{22} & \gamma x_{13} + \delta x_{23} \end{pmatrix}$$

Transpose, inverse, trace, determinant — all are available using a notation that quickly becomes natural if it is not already. First look at **A** again, and then the transpose.

```
show(A)
A.transpose()
```

evaluate

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

$$\begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix}$$

```
A.trace()
```

evaluate

$$\alpha + \delta$$

```
A.determinant()
```

evaluate

$$\alpha\delta - \beta\gamma$$

The following result runs off the page (Sagemath has a scrollbar) and is a reminder of Sagemath's ability to calculate expressions that are almost too complicated to look at.

```
D = C.transpose() # C is 2x3, D is 3x2
E = (C*D).inverse()   # Inverse of C*D
factor(E) # E is HUGE! This is not as bad. Factor is a good way to simplify.
```

evaluate

$$\begin{pmatrix} \frac{\gamma^2x_{11}^2+\gamma^2x_{12}^2+\gamma^2x_{13}^2+2\,\delta\gamma x_{11}x_{21}+\delta^2x_{21}^2+2\,\delta\gamma x_{12}x_{22}+\delta^2x_{22}^2+2\,\delta\gamma x_{13}x_{23}+\delta^2x_{23}^2}{(x_{12}^2x_{21}^2+x_{13}^2x_{21}^2-2\,x_{11}x_{12}x_{21}x_{22}+x_{11}^2x_{22}^2+x_{13}^2x_{22}^2-2\,x_{11}x_{13}x_{21}x_{23}-2\,x_{12}x_{13}x_{22}x_{23}+x_{11}^2x_{23}^2+x_{12}^2x_{23}^2)(\alpha\delta-\beta\gamma)^2} & -\frac{\alpha\gamma x_{11}^2+\alpha\gamma x_{12}^2+\alpha\gamma x_{13}^2+\alpha}{(x_{12}^2x_{21}^2+x_{13}^2x_{21}^2-2\,x_{11}x_1} \\ -\frac{\alpha\gamma x_{11}^2+\alpha\gamma x_{12}^2+\alpha\gamma x_{13}^2+\alpha\delta x_{11}x_{21}+\beta\gamma x_{11}x_{21}+\beta\delta x_{21}^2+\alpha\delta x_{12}x_{22}+\beta\gamma x_{12}x_{22}+\beta\delta x_{22}^2+\alpha\delta x_{13}x_{23}+\beta\gamma x_{13}x_{23}+\beta\delta x_{23}^2}{(x_{12}^2x_{21}^2+x_{13}^2x_{21}^2-2\,x_{11}x_{12}x_{21}x_{22}+x_{11}^2x_{22}^2+x_{13}^2x_{22}^2-2\,x_{11}x_{13}x_{21}x_{23}-2\,x_{12}x_{13}x_{22}x_{23}+x_{11}^2x_{23}^2+x_{12}^2x_{23}^2)(\alpha\delta-\beta\gamma)^2} & \frac{\alpha^2x_{11}^2+\alpha^2x}{(x_{12}^2x_{21}^2+x_{13}^2x_{21}^2-2\,x_{11}x_1} \end{pmatrix}$$

```
A.inverse() # Here is something we can look at without a scrollbar.
```

evaluate

$$\begin{pmatrix} \frac{1}{\alpha}+\frac{\beta\gamma}{\alpha^2\left(\delta-\frac{\beta\gamma}{\alpha}\right)} & -\frac{\beta}{\alpha\left(\delta-\frac{\beta\gamma}{\alpha}\right)} \\ -\frac{\gamma}{\alpha\left(\delta-\frac{\beta\gamma}{\alpha}\right)} & \frac{1}{\delta-\frac{\beta\gamma}{\alpha}} \end{pmatrix}$$

```
Ainverse = factor(_) # Factor the last expression.
Ainverse
```

evaluate

$$\begin{pmatrix} \frac{\delta}{\alpha\delta-\beta\gamma} & -\frac{\beta}{\alpha\delta-\beta\gamma} \\ -\frac{\gamma}{\alpha\delta-\beta\gamma} & \frac{\alpha}{\alpha\delta-\beta\gamma} \end{pmatrix}$$

That's better. Notice how Sagemath quietly assumes that $\alpha\delta \neq \beta\gamma$. This is typical behaviour, and usually what you want.
It's easy to get at the contents.

```
denominator(Ainverse[0,1])
```

evaluate

$$\alpha\delta-\beta\gamma$$

For a numerical (or partly numerical) example, just treat the matrix as a function.

```
Ainverse(alpha=1,gamma=2)
```

evaluate

$$\begin{pmatrix} -\frac{\delta}{2\,\beta-\delta} & \frac{\beta}{2\,\beta-\delta} \\ \frac{2}{2\,\beta-\delta} & -\frac{1}{2\,\beta-\delta} \end{pmatrix}$$

Recall the earlier example.

```
C
```

evaluate

$$\begin{pmatrix} \alpha x_{11}+\beta x_{21} & \alpha x_{12}+\beta x_{22} & \alpha x_{13}+\beta x_{23} \\ \gamma x_{11}+\delta x_{21} & \gamma x_{12}+\delta x_{22} & \gamma x_{13}+\delta x_{23} \end{pmatrix}$$

We had $\mathbf{D} = \mathbf{L}^\top$, so $\mathbf{D}$ is $3 \times 2$.

```
(D.nrows(),D.ncols()) # A tuple
```

evaluate

$(3, 2)$

This means $\mathbf{DC}$ is $3 \times 3$. It's awful to look at, but since the rank of a product is the minimum of the rank of the matrices being multiplied, the rank of $\mathbf{DC}$ must be two (with Sagemath's usual optimistic assumptions about symbolic functions not being equal to zero unless there is more information).

```
DC = D*C
DC.rank()
```

evaluate

2

```
A.eigenvalues() # Returns a list
```

evaluate

$$\left[\tfrac{1}{2}\,\alpha + \tfrac{1}{2}\,\delta - \tfrac{1}{2}\,\sqrt{\alpha^2 - 2\,\alpha\delta + \delta^2 + 4\,\beta\gamma},\ \tfrac{1}{2}\,\alpha + \tfrac{1}{2}\,\delta + \tfrac{1}{2}\,\sqrt{\alpha^2 - 2\,\alpha\delta + \delta^2 + 4\,\beta\gamma}\right]$$

The eigenvalues of a real symmetric matrix are real, and observe that in the last result the expression under the square root sign will be non-negative if $\mathbf{A}$ is symmetric — that is, if $\beta = \gamma$. Sagemath doesn't care about this; imaginary numbers are fine.

This is really just the basics. Sagemath's capabilities in linear algebra go much deeper, including Cholesky and Jordan decompositions, vector spaces and subspaces – the list goes on. As usual, you need to know the math to use it effectively. We have all we need for now.

**Applications to structural equation modeling** In structural equation modeling, we often find ourselves calculating the covariance matrix of the observable data as a function of the model parameters. For real-world models with lots of variables this can be a big, tedious job. It's largely a clerical task that Sagemath can do for you. Here, we'll just calculate the covariance matrices for a couple of structural equation models to illustrate how it goes. It's even easier with the sem package of Section B.2.

### Example B.1.1

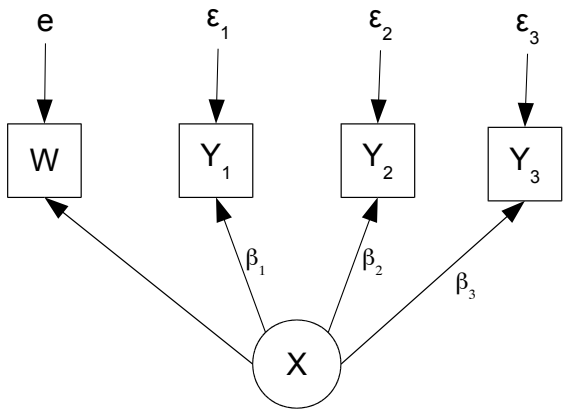
The first example is a small regression model with one latent explanatory variable and three observable response variables. A path diagram is shown in Figure B.1. Indepen-

dently for $i = 1, \ldots, n$,

$$\begin{aligned} W_i &= X_i + e_i \\ Y_{i,1} &= \beta_1 X_i + \epsilon_{i,1} \\ Y_{i,2} &= \beta_2 X_i + \epsilon_{i,2} \\ Y_{i,3} &= \beta_3 X_i + \epsilon_{i,3}, \end{aligned}$$

where $X_i$, $e_i$, $\epsilon_{i,1}$, $\epsilon_{i,2}$ and $\epsilon_{i,3}$ are all independent, $Var(X_i) = \phi$, $Var(e_i) = \omega$, $Var(\epsilon_{i,1}) = \psi_1$, $Var(\epsilon_{i,2}) = \psi_2$, $Var(\epsilon_{i,3}) = \psi_3$, all expected values are zero, and the regression coefficients $\beta_1$, $\beta_2$ and $\beta_3$ are fixed constants.

Figure B.1: Path diagram for Example B.1.1




To calculate the covariance matrix, write the model equations in matrix form as

$$\mathbf{Y}_i = \boldsymbol{\beta}\mathbf{X}_i + \boldsymbol{\epsilon}_i,$$

with $\mathbf{X}_i$ and $\boldsymbol{\epsilon}_i$ independent, $cov(\mathbf{X}_i) = \boldsymbol{\Phi}$, and $cov(\boldsymbol{\epsilon}_i) = \boldsymbol{\Psi}$. In the present case, this means

$$\begin{pmatrix} W_i \\ Y_{i,1} \\ Y_{i,2} \\ Y_{i,3} \end{pmatrix} = \begin{pmatrix} 1 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} (X_i) + \begin{pmatrix} e_i \\ \epsilon_{i,1} \\ \epsilon_{i,2} \\ \epsilon_{i,3} \end{pmatrix},$$

with $cov(X_i) = \boldsymbol{\Phi}$ equal to the $1 \times 1$ matrix $(\phi)$, and

$$cov \begin{pmatrix} e_i \\ \epsilon_{i,1} \\ \epsilon_{i,2} \\ \epsilon_{i,3} \end{pmatrix} = \boldsymbol{\Psi} = \begin{pmatrix} \omega & 0 & 0 & 0 \\ 0 & \psi_1 & 0 & 0 \\ 0 & 0 & \psi_2 & 0 \\ 0 & 0 & 0 & \psi_3 \end{pmatrix}.$$

The variance-covariance matrix of the observable variables is then

$$\begin{aligned} cov(\mathbf{Y}_i) &= cov\left(\boldsymbol{\beta}\mathbf{X}_i + \boldsymbol{\epsilon}_i\right) \\ &= \boldsymbol{\beta}\boldsymbol{\Phi}\boldsymbol{\beta}^\top + \boldsymbol{\Psi}. \end{aligned}$$

This is the quantity we'll compute with Sagemath.

```
# Ex 1 - Single measurement but 3 response variables
beta = matrix(SR,4,1) # SR is the Symbolic Ring. Want 4 rows, 1 col.
beta[0,0] = 1 ; beta[1,0] = var('beta1'); beta[2,0] = var('beta2');
beta[3,0] = var('beta3')
beta
```

evaluate

$$\begin{pmatrix} 1 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}$$

```
Phi = matrix(SR,1,1); Phi[0,0] = var('phi')
show(Phi)
Psi = matrix(SR,4,4)
Psi[0,0] = var('omega'); Psi[1,1] = var('psi1')
Psi[2,2] = var('psi2'); Psi[3,3] = var('psi3')
Psi
```

evaluate

$$\begin{pmatrix} \phi \end{pmatrix}$$

$$\begin{pmatrix} \omega & 0 & 0 & 0 \\ 0 & \psi_1 & 0 & 0 \\ 0 & 0 & \psi_2 & 0 \\ 0 & 0 & 0 & \psi_3 \end{pmatrix}$$

```
Sigma = beta*Phi*beta.transpose() + Psi ; Sigma
```

evaluate

$$\begin{pmatrix} \omega+\phi & \beta_1\phi & \beta_2\phi & \beta_3\phi \\ \beta_1\phi & \beta_1^2\phi+\psi_1 & \beta_1\beta_2\phi & \beta_1\beta_3\phi \\ \beta_2\phi & \beta_1\beta_2\phi & \beta_2^2\phi+\psi_2 & \beta_2\beta_3\phi \\ \beta_3\phi & \beta_1\beta_3\phi & \beta_2\beta_3\phi & \beta_3^2\phi+\psi_3 \end{pmatrix}$$

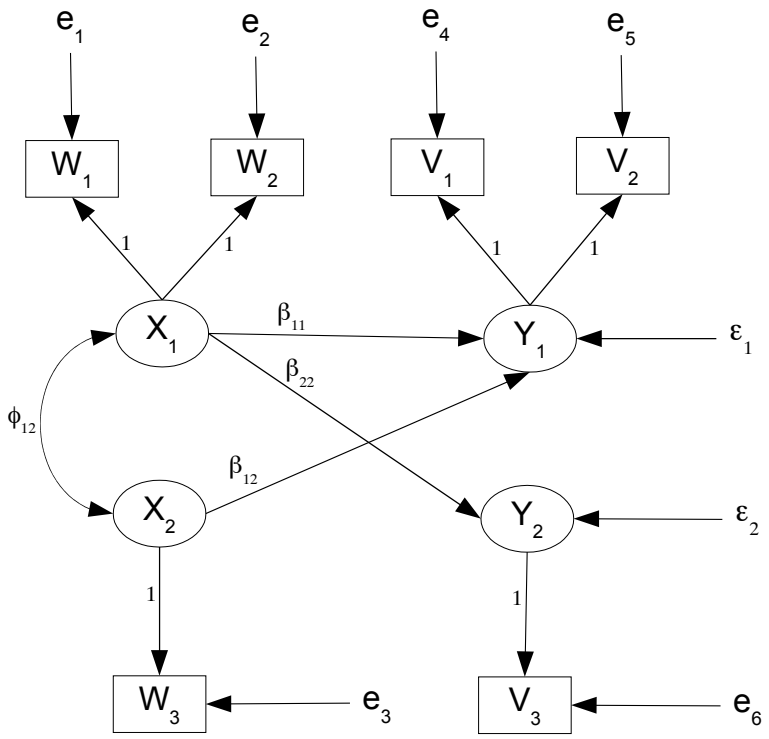
It is clear that all the parameters will be identifiable provided that at least two of the three regression coefficients are non-zero. This condition could be verified in practice by testing whether simple correlations are different from zero.

### Example B.1.2

This example is a latent variable regression that does not fit the standard rules. The latent variable component is over-identified while the measurement component is under-

identified. Parameter identifiability for the combined model is unknown, and it's back to the drawing board. Here is the path diagram.

Figure B.2: Path Diagram for Example B.1.2




The distinctive features of this model are that while $Y_1$ depends on both $X_1$ and $X_2$, $Y_2$ depends only on $X_2$ — and at the same time, there is double measurement of $X_1$ and $Y_1$, but only single measurement of $X_2$ and $Y_2$. There are 14 unknown parameters and $6(6+1)/2 = 21$ covariance structure equations, so the model passes the test of the parameter count rule. Identifiability is possible, but not guaranteed. The first step is to calculate the 21 unique variances and covariances, a substantial amount of work if the calculation is done by hand.

It's a lot easier with `Sagemath`, but still a bit more challenging than Example B.1.1. First, we calculate the covariance matrix for a latent model, stitching together a partitioned matrix consisting of the variance of the exogenous variables, the covariance of the exogenous and endogenous variables, and the variance of the endogenous variables. Then that matrix is used as the covariance matrix of the latent variables ("factors") in a measurement model. The model equations are (independently for $i = 1, \ldots, n$)

$$\begin{pmatrix} Y_{i,1} \\ Y_{i,2} \end{pmatrix} = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} \\ 0 & \beta_{2,2} \end{pmatrix} \begin{pmatrix} X_{i,1} \\ X_{i,2} \end{pmatrix} + \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \end{pmatrix}$$

and

$$\mathbf{D}_i = \begin{pmatrix} W_{i,1} \\ W_{i,2} \\ W_{i,3} \\ V_{i,1} \\ V_{i,2} \\ V_{i,3} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{i,1} \\ X_{i,2} \\ Y_{i,1} \\ Y_{i,2} \end{pmatrix} + \begin{pmatrix} e_{i,1} \\ e_{i,2} \\ e_{i,3} \\ e_{i,4} \\ e_{i,5} \\ e_{i,6} \end{pmatrix},$$

where

- $cov \begin{pmatrix} X_{i,1} \\ X_{i,2} \end{pmatrix} = \mathbf{\Phi}_x = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}$,
- $\mathbf{\Phi}_x$ is positive definite,
- $cov(\epsilon_{i,1}) = \psi_2$, $cov(\epsilon_{i,2}) = \psi_2$,
- $cov(e_{i,j}) = \omega_j$ for $j = 1, \ldots, 6$ and
- All the error terms are independent of one another, and independent of $X_{i,1}$ and $X_{i,2}$.

To calculate the covariance matrix of the observed data $\mathbf{D}_i$, write the model equations as

$$\begin{aligned} \mathbf{Y}_i &= \boldsymbol{\beta}\mathbf{X}_i + \boldsymbol{\epsilon}_i \\ \mathbf{D}_i &= \mathbf{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \end{aligned}$$

where $\mathbf{F}_i = \left(\dfrac{\mathbf{X}_i}{\mathbf{Y}_i}\right)$. That is, the vector of latent variables or "factors" is just $\mathbf{X}_i$ stacked on top of $\mathbf{Y}_i$. Denoting the variance-covariance matrices by $cov(\mathbf{X}_i) = \mathbf{\Phi}_x$, $cov(\boldsymbol{\epsilon}_i) = \mathbf{\Psi}$ and $cov(\mathbf{e}_i) = \mathbf{\Omega}$, we first calculate the variance-covariance matrix of $\mathbf{F}_i$ as the partitioned matrix

$$cov(\mathbf{F}_i) = \mathbf{\Phi} = \left(\begin{array}{c|c} \mathbf{\Phi}_x & \mathbf{\Phi}_x\boldsymbol{\beta}^\top \\ \hline \boldsymbol{\beta}\mathbf{\Phi}_x & \boldsymbol{\beta}\mathbf{\Phi}_x\boldsymbol{\beta}^\top + \mathbf{\Psi} \end{array}\right),$$

and then using that, the variance-covariance matrix of the observed data:

$$cov(\mathbf{D}_i) = \mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Phi}\mathbf{\Lambda}^\top + \mathbf{\Omega}.$$

Here is the calculation in `Sagemath`.

```
# Ex 2 - More challenging

# Y = beta X + epsilon
# F = (X,Y)'
# D = Lambda F + e
# cov(X) = Phi11, cov(epsilon) = Psi, cov(e) = Omega

# Set up matrices
beta = matrix(SR,2,2)
beta[0,0] = var('beta11'); beta[0,1] = var('beta12')
beta[1,0] = var('beta21');  beta[1,1] = var('beta22')
beta[1,0] = 0
show(beta)
```

evaluate

$$\begin{pmatrix} \beta_{11} & \beta_{12} \\ 0 & \beta_{22} \end{pmatrix}$$

```
Phi11 =  matrix(SR,2,2) # cov(X), Symmetric
Phi11[0,0] = var('phi11'); Phi11[0,1] = var('phi12')
Phi11[1,0] = var('phi12'); Phi11[1,1] = var('phi22')
show(Phi11)
```

evaluate

$$\begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{12} & \phi_{22} \end{pmatrix}$$

```
Psi =  matrix(SR,2,2) # cov(epsilon)
Psi[0,0] = var('psi1') ; Psi[1,1] = var('psi2')
show(Psi)
```

evaluate

$$\begin{pmatrix} \psi_{1} & 0 \\ 0 & \psi_{2} \end{pmatrix}$$

```
Omega =  matrix(SR,6,6) # cov(e)
Omega[0,0] = var('omega1') ; Omega[1,1] = var('omega2')
Omega[2,2] = var('omega3') ; Omega[3,3] = var('omega4')
Omega[4,4] = var('omega5'); Omega[5,5] = var('omega6')
show(Omega)
```

evaluate

$$\begin{pmatrix} \omega_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \omega_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_6 \end{pmatrix}$$

```
Lambda = matrix(SR,6,4)
Lambda[0,0]=1; Lambda[1,0]=1; Lambda[2,1]=1
Lambda[3,2]=1; Lambda[4,2]=1; Lambda[5,3]=1
show(Lambda)
```

evaluate

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
# Calculate Phi = cov(F)
EXY = Phi11 * beta.transpose()
VY = beta*Phi11*beta.transpose() + Psi
top = Phi11.augment(EXY) # Phi11 on left, EXY on right
bot = EXY.transpose().augment(VY)
Phi = (top.stack(bot)).factor() # Stack top over bot, then factor
show(Phi)
```

evaluate

$$\begin{pmatrix} \phi_{11} & \phi_{12} & \beta_{11}\phi_{11} + \beta_{12}\phi_{12} & \beta_{22}\phi_{12} \\ \phi_{12} & \phi_{22} & \beta_{11}\phi_{12} + \beta_{12}\phi_{22} & \beta_{22}\phi_{22} \\ \beta_{11}\phi_{11} + \beta_{12}\phi_{12} & \beta_{11}\phi_{12} + \beta_{12}\phi_{22} & \beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \psi_1 & (\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22} \\ \beta_{22}\phi_{12} & \beta_{22}\phi_{22} & (\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22} & \beta_{22}^2\phi_{22} + \psi_2 \end{pmatrix}$$

```
# Calculate Sigma = cov(D)
Sigma = Lambda * Phi * Lambda.transpose() + Omega
show(Sigma)
```

evaluate

$$\begin{pmatrix} \omega_1 + \phi_{11} & \phi_{11} & \phi_{12} & \beta_{11}\phi_{11} + \beta_{12}\phi_{12} & \beta_{11}\phi_{11} + \beta \\ \phi_{11} & \omega_2 + \phi_{11} & \phi_{12} & \beta_{11}\phi_{11} + \beta_{12}\phi_{12} & \beta_{11}\phi_{11} + \beta \\ \phi_{12} & \phi_{12} & \omega_3 + \phi_{22} & \beta_{11}\phi_{12} + \beta_{12}\phi_{22} & \beta_{11}\phi_{12} + \beta \\ \beta_{11}\phi_{11} + \beta_{12}\phi_{12} & \beta_{11}\phi_{11} + \beta_{12}\phi_{12} & \beta_{11}\phi_{12} + \beta_{12}\phi_{22} & \beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \omega_4 + \psi_1 & \beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} \\ \beta_{11}\phi_{11} + \beta_{12}\phi_{12} & \beta_{11}\phi_{11} + \beta_{12}\phi_{12} & \beta_{11}\phi_{12} + \beta_{12}\phi_{22} & \beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \psi_1 & \beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \omega \\ \beta_{22}\phi_{12} & \beta_{22}\phi_{12} & \beta_{22}\phi_{22} & (\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22} & (\beta_{11}\phi_{12} + \beta_{12}\phi \end{pmatrix}$$

Again, this is the covariance matrix of the observable data vector $\mathbf{D}_i = (W_{i,1}, W_{i,2}, W_{i,3}, V_{i,1}, V_{i,2}, V_{i,3})^\top$. The covariance matrix is big and the last two columns got cut off, but in `Sagemath` you can scroll to the right and see something like the following:

$$\left(\begin{array}{ccc}
\cdots & \beta_{11}\phi_{11}+\beta_{12}\phi_{12} & \beta_{11}\phi_{11}+\beta_{12}\phi_{12} \\
& \beta_{11}\phi_{11}+\beta_{12}\phi_{12} & \beta_{11}\phi_{11}+\beta_{12}\phi_{12} \\
& \beta_{11}\phi_{12}+\beta_{12}\phi_{22} & \beta_{11}\phi_{12}+\beta_{12}\phi_{22} \\
& \beta_{11}^2\phi_{11}+2\,\beta_{11}\beta_{12}\phi_{12}+\beta_{12}^2\phi_{22}+\omega_4+\psi_1 & \beta_{11}^2\phi_{11}+2\,\beta_{11}\beta_{12}\phi_{12}+\beta_{12}^2\phi_{22}+\psi_1 \\
& \beta_{11}^2\phi_{11}+2\,\beta_{11}\beta_{12}\phi_{12}+\beta_{12}^2\phi_{22}+\psi_1 & \beta_{11}^2\phi_{11}+2\,\beta_{11}\beta_{12}\phi_{12}+\beta_{12}^2\phi_{22}+\omega_5+\psi_1 \\
& (\beta_{11}\phi_{12}+\beta_{12}\phi_{22})\beta_{22} & (\beta_{11}\phi_{12}+\beta_{12}\phi_{22})\beta_{22}
\end{array}\right.
\left.\begin{array}{c}
\beta_{22}\phi_{12} \\
\beta_{22}\phi_{12} \\
\beta_{22}\phi_{22} \\
(\beta_{11}\phi_{12}+\beta_{12}\phi_{22})\beta_{22} \\
(\beta_{11}\phi_{12}+\beta_{12}\phi_{22})\beta_{22} \\
\beta_{22}^2\phi_{22}+\omega_6+\psi_2
\end{array}\right)$$

Now it appears that at points in the parameter space where $\phi_{12} \neq 0$, the regression parameters $\beta_{11}$, $\beta_{12}$ and $\beta_{22}$ may be identifiable in spite of the single measurement. This is just a tentative conclusion based on inspecting the equations without actually doing all the work. We will continue to work on this example using the tools of the `sem` package.

# B.2 The sem Package

## B.2.1 Introduction and Examples

Example B.1.2 showed how `Sagemath` can be used to carry out useful symbolic calculations that are too tedious to perform by hand. Even with `Sagemath`, parts of the job can be repetitive and this can be a barrier to using the technology. Fortunately, it is easy for users to write special purpose functions. The `sem` package is a collection of functions for structural equation modeling. Currently, it is limited to symbolic calculation. For numerical model fitting, it is necessary to use specialized statistical software[9].

To load the `sem` package,

```
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
# load('~/sem.sage') # To load a local version in your home directory
```

evaluate

After the package is loaded, `Contents()` will display a list of the available functions. For help on a particular function, type the function name followed by a question mark, like `PathCov?`

The `sem` package currently includes the following functions. You can go directly to the documentation for a particular function, or continue reading to see how the functions are used together in context.

[9] `Sagemath` has very strong numerical capabilities, and it would not be very difficult to write a function to do numerical maximum likelihood estimation. What holds me back is the issue of starting values. Programs like `Amos`, `Lisrel` and SAS `proc calis` have extensive bags of tricks for generating automatic starting values, and typically they are very good. It is difficult to appreciate how convenient they are until you have tried to come up with your own starting values for a few models.

1. Matrix Creation

   1a) `DiagonalMatrix(size,symbol='psi',double=False)`

   1b) `GeneralMatrix(nrows,ncols,symbol)`

   1c) `IdentityMatrix(size)`

   1d) `SymmetricMatrix(size,symbol,corr=False)`

   1e) `ZeroMatrix(nrows,ncols)`

2. Covariance Matrix Calculation

   2a) `EqsCov(beta,gamma,Phi,oblist,simple=True)`

   2b) `FactorAnalysisCov(Lambda,Phi,Omega)`

   2c) `NoGammaCov(Beta,Psi)`

   2d) `PathCov(Phi,Beta,Gamma,Psi,simple=True)`

   2e) `RegressionCov(Phi,Gamma,Psi,simple=True)`

3. Manipulation

   3a) `GroebnerBasis(polynomials,variables)`

   3b) `LSTarget(M,x,y)`

   3c) `Parameters(M)`

   3d) `SigmaOfTheta(M,symbol='sigma')`

   3e) `Simplify(x)`

4. Utility

   4a) `BetaCheck(Beta)`

   4b) `Contents()`

   4c) `CovCheck(Psi)`

   4d) `MultCheck(Beta,Psi)`

   4e) `Pad(M)`

Here is Example B.1.2 again from the beginning, using the `sem` package. Repeating the model equations,

$$\begin{array}{ccccc} \mathbf{y}_i & = & \boldsymbol{\beta} & \mathbf{x}_i & + & \boldsymbol{\epsilon}_i \\ \begin{pmatrix} y_{i,1} \\ y_{i,2} \end{pmatrix} & = & \begin{pmatrix} \beta_{1,1} & \beta_{1,2} \\ 0 & \beta_{2,2} \end{pmatrix} & \begin{pmatrix} x_{i,1} \\ x_{i,2} \end{pmatrix} & + & \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{i,2} \end{pmatrix} \end{array}$$

$$\begin{array}{ccccc} \mathbf{d}_i & = & \boldsymbol{\Lambda} & \mathbf{F}_i & + & \mathbf{e}_i \\ \begin{pmatrix} w_{i,1} \\ w_{i,2} \\ w_{i,3} \\ v_{i,1} \\ v_{i,2} \\ v_{i,3} \end{pmatrix} & = & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ y_{i,1} \\ y_{i,2} \end{pmatrix} & + & \begin{pmatrix} e_{i,1} \\ e_{i,2} \\ e_{i,3} \\ e_{i,4} \\ e_{i,5} \\ e_{i,6} \end{pmatrix} \end{array}$$

```
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
# Set up matrices (Remember, indices begin with zero)
beta = GeneralMatrix(2,2,'beta'); beta[1,0]=0
Phi11 = SymmetricMatrix(2,'phi')  # cov(X)
Psi = DiagonalMatrix(2,'psi')     # cov(epsilon)
Omega = DiagonalMatrix(6,'omega') # cov(e)
Lambda = ZeroMatrix(6,4) # Factor loadings
Lambda[0,0]=1; Lambda[1,0]=1; Lambda[2,1]=1
Lambda[3,2]=1; Lambda[4,2]=1; Lambda[5,3]=1
```

evaluate

The `GeneralMatrix` function generates doubly subscripted symbols by default; it is easy to replace the lower left entry with a zero. The other functions are pretty much self-explanatory, but see the links to function documentation above. In general, native `Sagemath` functions are lower case, while functions in the `sem` package are capitalized. This makes them easy to distinguish in the examples. Next we calculate $\boldsymbol{\Sigma}$ the easy way. The output is not shown because it is big and you have seen it before on page 661.

```
# Calculate Phi = cov(F)
Phi = RegressionCov(Phi11,beta,Psi) # The first argument is cov(X)
# Calculate Sigma = cov(D)
Sigma = FactorAnalysisCov(Lambda,Phi,Omega); Sigma
```

evaluate

Based on inspection of $\boldsymbol{\Sigma}$, I tentatively concluded that the parameters were identifiable, at least in most of the parameter space. Now we will nail it down. The `SetupEqns` function assembles a list of covariance structure equations. Each equation is displayed with its index as a tuple – not very pretty, but useful when one needs to refer to equations by number (starting with zero). Note the Python syntax for looping.

```
eqlist = SetupEqns(Sigma); k = len(eqlist)
for index in range(k): index,eqlist[index]
```

evaluate

$(0,\ \omega_1 + \phi_{11} = \sigma_{11})$
$(1,\ \phi_{11} = \sigma_{12})$
$(2,\ \phi_{12} = \sigma_{13})$
$(3,\ \beta_{11}\phi_{11} + \beta_{12}\phi_{12} = \sigma_{14})$
$(4,\ \beta_{11}\phi_{11} + \beta_{12}\phi_{12} = \sigma_{15})$
$(5,\ \beta_{22}\phi_{12} = \sigma_{16})$
$(6,\ \omega_2 + \phi_{11} = \sigma_{22})$
$(7,\ \phi_{12} = \sigma_{23})$
$(8,\ \beta_{11}\phi_{11} + \beta_{12}\phi_{12} = \sigma_{24})$
$(9,\ \beta_{11}\phi_{11} + \beta_{12}\phi_{12} = \sigma_{25})$
$(10,\ \beta_{22}\phi_{12} = \sigma_{26})$
$(11,\ \omega_3 + \phi_{22} = \sigma_{33})$
$(12,\ \beta_{11}\phi_{12} + \beta_{12}\phi_{22} = \sigma_{34})$
$(13,\ \beta_{11}\phi_{12} + \beta_{12}\phi_{22} = \sigma_{35})$
$(14,\ \beta_{22}\phi_{22} = \sigma_{36})$
$(15,\ \beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \omega_4 + \psi_1 = \sigma_{44})$
$(16,\ \beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \psi_1 = \sigma_{45})$
$(17,\ (\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22} = \sigma_{46})$
$(18,\ \beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \omega_5 + \psi_1 = \sigma_{55})$
$(19,\ (\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22} = \sigma_{56})$
$(20,\ \beta_{22}^2\phi_{22} + \omega_6 + \psi_2 = \sigma_{66})$

The next step is to assemble a list of model parameters. The function `Parameters` returns a list of the parameters in a parameter matrix — that is, a list of the unique elements that are not one or zero. Unfortunately, it cannot operate on a computed covariance matrix, just on the parmeter matrices that are used as input. Still, it's better than doing the job by hand.

```
# Assemble a list of model parameters. I count 14 by hand.
param = Parameters(beta) # Start with parameters in beta
param.extend(Parameters(Phi11)) # Add the parameters in Phi11
param.extend(Parameters(Psi)) # Add the parameters in Psi
param.extend(Parameters(Omega)) # Add the parameters in Omega
param.extend(Parameters(Lambda)) # Add the parameters in Lambda
show(param)
len(eqlist), len(param) # This many equations in this many unknowns
```

evaluate

$[\beta_{11}, \beta_{12}, \beta_{22}, \phi_{11}, \phi_{12}, \phi_{22}, \psi_1, \psi_2, \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6]$

(21, 14)

So there are 21 equations in 14 unknowns. `Sagemath`'s very powerful `solve` function requires the same number of equations as unknowns and will not work here. However, we'll try it anyway to see what happens.

```
solve(eqlist,param,solution_dict=True)
```

evaluate

[]

That little rectangle is a left square bracket followed by a right square bracket; it's an empty list (empty set), meaning that the system of equations has no general solution. This happens because, for example, equation number two in the list says $\phi_{12} = \sigma_{13}$, while equation seven says $\phi_{12} = \sigma_{23}$. To `Sagemath`, $\sigma_{13}$ and $\sigma_{23}$ are just numbers, and there is no reason to assume they are equal. Thus there is no *general* solution.

Actually, because we think of the $\sigma_{ij}$ values as arising from a single, fixed point in the parameter space, we recognize $\sigma_{13} = \sigma_{24}$ as a distinctive feature that the model imposes on the covariance matrix $\mathbf{\Sigma}$. But `Sagemath` does not know this, and I don't know how to tell it without specifying exactly what the restrictions are. One solution is to set aside the redundant equations and then give the solve function a system with the same number of equations and unknowns. Unfortunately, this is not automatic because it is not always obvious which equations are redundant. Groebner basis methods (to be discussed later in this appendix) can do the job automatically when they work.

Because there are 21 equations in 14 unknowns, there should be seven equality constraints; seven equations should be redundant. Carefully inspecting the covariance structure equations, I conclude

- $\sigma_{15}, \sigma_{24}$ and $\sigma_{25}$ are redundant with $\sigma_{14}$.
- $\sigma_{26}$ is redundant with $\sigma_{16}$.
- $\sigma_{23}$ is redundant with $\sigma_{13}$.
- $\sigma_{35}$ is redundant with $\sigma_{34}$.
- $\sigma_{56}$ is redundant with $\sigma_{46}$.

```
# Set redundant equations aside.
extra = [4,8,9,7,10,13,19] # Indices of redundant equations
extra.sort() # Sort them (change in place)
# Save and display the redundant equations
aside = [] # Empty list to start
for index in extra:
    extraeq = eqlist[index]
    show(extraeq)
    aside.append(extraeq)
```

evaluate

$\beta_{11}\phi_{11} + \beta_{12}\phi_{12} = \sigma_{15}\phi_{12} = \sigma_{23}$
$\beta_{11}\phi_{11} + \beta_{12}\phi_{12} = \sigma_{24}$
$\beta_{11}\phi_{11} + \beta_{12}\phi_{12} = \sigma_{25}$
$\beta_{22}\phi_{12} = \sigma_{26}$
$\beta_{11}\phi_{12} + \beta_{12}\phi_{22} = \sigma_{35}$
$(\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22} = \sigma_{56}$

```
# Remove extra equations
for item in aside: eqlist.remove(item)
len(eqlist) # Should be 14 now
```

evaluate

14

```
# Solve, returning solutions as a list of dictionaries
solist = solve(eqlist,param,solution_dict=True)
len(solist) # Should have one item (unique solution)
```

evaluate

0

The length of the list is zero; there are no solutions, meaning no general solutions according to Sagemath. This is a sure sign of redundancy in the covariance structure equations we are trying to solve. They still imply one or more constraints on the $\sigma_{ij}$ quantities – constraints that Sagemath does not accept. In other words, we missed something. Looking at the covariance structure equations again,

```
for item in eqlist: item
```

evaluate

$\omega_1 + \phi_{11} = \sigma_{11}$
$\phi_{11} = \sigma_{12}$
$\phi_{12} = \sigma_{13}$
$\beta_{11}\phi_{11} + \beta_{12}\phi_{12} = \sigma_{14}$
$\beta_{22}\phi_{12} = \sigma_{16}$
$\omega_2 + \phi_{11} = \sigma_{22}$
$\omega_3 + \phi_{22} = \sigma_{33}$
$\beta_{11}\phi_{12} + \beta_{12}\phi_{22} = \sigma_{34}$
$\beta_{22}\phi_{22} = \sigma_{36}$
$\beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \omega_4 + \psi_1 = \sigma_{44}$
$\beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \psi_1 = \sigma_{45}$
$(\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22} = \sigma_{46}$
$\beta_{11}^2\phi_{11} + 2\,\beta_{11}\beta_{12}\phi_{12} + \beta_{12}^2\phi_{22} + \omega_5 + \psi_1 = \sigma_{55}$
$\beta_{22}^2\phi_{22} + \omega_6 + \psi_2 = \sigma_{66}$

To be honest, it took me a while to see it. The parameters $\omega_6$ and $\psi_2$ appear only in the last equation, as a sum. This means that infinitely many pairs $(\omega_6, \psi_2)$ will satisfy the system of equations. Those parameters are not identifiable. A glance at the path diagram on 656 shows why. Because $Y_2$ does not influence any other variables in the latent model, measuring it just once means that the variance of $V_2$ is just the variance of $Y_2$ plus $\omega_6$, with no hope of separating $\omega_6$ from $\psi_2$.

The solution is easy; re-parameterize by combining $\omega_6$ and $\psi_2$ into a single variance parameter. This could be accomplished by re-writing the path diagram and running an arrow directly from $X_1$ to $V_3$. When a purely endogenous variable (that is, purely endogenous in the latent model) is measured once, pretending that it is measured without error is a standard, harmless trick. Here, it's unnecessary to make a new path diagram and calculate the covariance structure equations again. Just setting $\omega_6 = 0$ would effectively treat $\omega_6 + \psi_2$ as a single parameter now called $\psi_2$.

But now there are more equations than unknowns, implying another equality constraint I missed. After looking at the equations for a while, I finally saw it. It's the third equation from the bottom. Starting at the third equation from the top, $\phi_{12}$ is identified from $\sigma_{13}$, and using that, $\beta_{22}$ is identified from $\sigma_{16}$. The equation for $\sigma_{46}$ (third from the bottom) is $\beta_{22}$ multiplied by the expression for $\sigma_{34}$. So the third equation from the bottom is redundant, and induces an equality constraint. Starting with zero, that should be equation eleven. Check it.

```
sig46 = eqlist[11]; sig46
```

evaluate

$(\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22} = \sigma_{46}$

Now we'll remove that equation from the list of covariance structure equations and add it to the list of equations we set aside. Once we finally get a list of explicit solutions of the covariance structure equations, we can obtain the equality constraints by substituting

the solutions into the the equations that were set aside.

```
aside.append(sig46)
eqlist.remove(sig46); len(eqlist)
```

evaluate

13

We now have thirteen equations in fourteen unknown parameters. Before re-parameterizing by setting $\omega_6 = 0$, let's see how `Sagemath` deals with infinitely many solutions. One might expect it to hang up, but the task is completed instantly.

```
# Now there are 13 equations in 14 unknown parameters. See what happens
# when we try to solve. Return the solutions as a list of dictionaries.
solist = solve(eqlist,param,solution_dict=True)
len(solist)
```

evaluate

1

One dictionary (essentially a Python dictionary) looks like one solution – unique. This is odd. How many items are in the dictionary?

```
sol = solist[0]
len(sol)
```

evaluate

14

There are fourteen items in the dictionary, suggesting one solution for each of the 14 parameters. This is unexpected, because we know there are infinitely many solutions. Let's take a look. The keys of the dictionary are the parameters, and the corresponding values are the solutions in terms of the $\sigma_{ij}$s. As in Python, `dictionary[key]` yields `value`.

```
# Display the solutions. item==sol[item] just causes that equation
# to be displayed.
for item in param: item==sol[item]
```

evaluate

$$\beta_{11} = \frac{\sigma_{16}\sigma_{34}-\sigma_{14}\sigma_{36}}{\sigma_{13}\sigma_{16}-\sigma_{12}\sigma_{36}}$$
$$\beta_{12} = \frac{\sigma_{13}\sigma_{14}\sigma_{16}-\sigma_{12}\sigma_{16}\sigma_{34}}{\sigma_{13}^2\sigma_{16}-\sigma_{12}\sigma_{13}\sigma_{36}}$$
$$\beta_{22} = \frac{\sigma_{16}}{\sigma_{13}}$$
$$\phi_{11} = \sigma_{12}$$
$$\phi_{12} = \sigma_{13}$$
$$\phi_{22} = \frac{\sigma_{13}\sigma_{36}}{\sigma_{16}}$$
$$\psi_1 = -\frac{2\,\sigma_{13}\sigma_{14}\sigma_{16}\sigma_{34}-\sigma_{12}\sigma_{16}\sigma_{34}^2-\sigma_{13}\sigma_{14}^2\sigma_{36}-\left(\sigma_{13}^2\sigma_{16}-\sigma_{12}\sigma_{13}\sigma_{36}\right)\sigma_{45}}{\sigma_{13}^2\sigma_{16}-\sigma_{12}\sigma_{13}\sigma_{36}}$$
$$\psi_2 = r_1$$
$$\omega_1 = \sigma_{11} - \sigma_{12}$$
$$\omega_2 = -\sigma_{12} + \sigma_{22}$$
$$\omega_3 = \frac{\sigma_{16}\sigma_{33}-\sigma_{13}\sigma_{36}}{\sigma_{16}}$$
$$\omega_4 = \sigma_{44} - \sigma_{45}$$
$$\omega_5 = -\sigma_{45} + \sigma_{55}$$
$$\omega_6 = -\frac{r_1\sigma_{13}+\sigma_{16}\sigma_{36}-\sigma_{13}\sigma_{66}}{\sigma_{13}}$$

Scanning down the list, we see $\psi_2 = r_1$. The quantity $r_1$ (which we have not seen before) is an arbitrary variable that could be anything[10]. I believe the `Sagemath` people call it a *parameter*, which is vastly different from a parameter in statistical estimation. Right at the bottom of the list is the solution $\omega_6 = -\frac{r_1\sigma_{13}+\sigma_{16}\sigma_{36}-\sigma_{13}\sigma_{66}}{\sigma_{13}}$. This neatly expresses the infinitely many solutions to the covariance structure equations. All the other solutions are unique (provided that denominators are non-zero), but the pair $(\omega_6, \psi_2)$ can be recovered from $\mathbf{\Sigma}$ in infinitely many ways, one for each $r_1 > 0$.

This is so nice that we will not bother to re-parameterize and obtain a unique solution. Of course with real data, one would have to re-parameterize $\omega_6$ and $\psi_2$ in order to estimate the other parameters by maximum likelihood, because otherwise the maximum would not be unique and there would be unpleasant numerical consequences.

Our main interest is in $\beta_{11}$, $\beta_{12}$ and $\beta_{22}$. The existence of unique solutions means that these parameters are identifiable (and as a practical matter, estimable) as long as the denominators are non-zero. The natural thing is to substitute for those $\sigma_{ij}$ quantities in the denominators, in terms of model parameters. Perhaps the denominators are never zero, or perhaps the $\beta ij$s can be identified in some other way when they are.

The formula for $\beta_{22}$ is simplest. Scanning the list of solutions, we see $\phi_{12} = \sigma_{13}$. So, the solution for $\beta_{22}$ does not apply when the two latent explanatory variables have zero covariance. Perhaps there is another way.

[10] `Sagemath` does not know that $r_1 = \psi_2$ is positive, or even that it's a real number.

```
# Can beta22 be identified when phi12=0?
factor(Sigma(phi12=0))
```

evaluate

$$\begin{pmatrix} \omega_1+\phi_{11} & \phi_{11} & 0 & \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & 0 \\ \phi_{11} & \omega_2+\phi_{11} & 0 & \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & 0 \\ 0 & 0 & \omega_3+\phi_{22} & \beta_{12}\phi_{22} & \beta_{12}\phi_{22} & \beta_{22}\phi_{22} \\ \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & \beta_{12}\phi_{22} & \beta_{11}^2\phi_{11}+\beta_{12}^2\phi_{22}+\omega_4+\psi_1 & \beta_{11}^2\phi_{11}+\beta_{12}^2\phi_{22}+\psi_1 & \beta_{12}\beta_{22}\phi_{22} \\ \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & \beta_{12}\phi_{22} & \beta_{11}^2\phi_{11}+\beta_{12}^2\phi_{22}+\psi_1 & \beta_{11}^2\phi_{11}+\beta_{12}^2\phi_{22}+\omega_5+\psi_1 & \beta_{12}\beta_{22}\phi_{22} \\ 0 & 0 & \beta_{22}\phi_{22} & \beta_{12}\beta_{22}\phi_{22} & \beta_{12}\beta_{22}\phi_{22} & \beta_{22}^2\phi_{22}+\omega_6+\psi_2 \end{pmatrix}$$

Yes! As long as $\beta_{12} \neq 0$,

$$\frac{\sigma_{46}}{\sigma_{34}} = \frac{\beta_{12}\beta_{22}\phi_{22}}{\beta_{12}\phi_{22}} = \beta_{22}$$

Actually, this way of identifying $\beta_{22}$ works even when $\phi_{12} \neq 0$. We could scroll up and look at the orignal $\boldsymbol{\Sigma}$ in terms of the parameters. Or, the `sem` package's `pad` function can be used to add a row and column of zeros to a matrix, making it more convenient to refer to the elements.

```
# Does sigma46/sigma34 work without phi12=0?
padSigma = Pad(Sigma)
show(padSigma[4,6]); show(padSigma[3,4])
```

evaluate

$$(\beta_{11}\phi_{12}+\beta_{12}\phi_{22})\beta_{22}$$
$$\beta_{11}\phi_{12}+\beta_{12}\phi_{22}$$

The identifying solution $\beta_{22} = \frac{\sigma_{46}}{\sigma_{34}}$ is superior to the solution $\beta_{22} = \frac{\sigma_{16}}{\sigma_{13}}$ on page 669, because it only fails when *both* $\beta_{12} = 0$ and ($\beta_{11} = 0$ or $\phi_{12} = 0$). Some ways of solving the covariance structure equations are better than others, in the sense that they reveal more clearly where in the parameter space the parameters are identifiable. `Sagemath`'s `solve` function will not necessarily locate the most informative solution, and neither will you if you do it by hand.

The solution $\beta_{22} = \frac{\sigma_{46}}{\sigma_{34}}$ does not apply when both $\phi_{12} = 0$, and $\beta_{12} = 0$, but it is a good idea to examine $\boldsymbol{\Sigma}$ under these conditions to see if yet another solution appears.

```
# What if both phi12 and beta12 equal zero?
factor(Sigma(phi12=0,beta12=0))
```

evaluate

$$\begin{pmatrix} \omega_1+\phi_{11} & \phi_{11} & 0 & \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & 0 \\ \phi_{11} & \omega_2+\phi_{11} & 0 & \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & 0 \\ 0 & 0 & \omega_3+\phi_{22} & 0 & 0 & \beta_{22}\phi_{22} \\ \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & 0 & \beta_{11}^2\phi_{11}+\omega_4+\psi_1 & \beta_{11}^2\phi_{11}+\psi_1 & 0 \\ \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & 0 & \beta_{11}^2\phi_{11}+\psi_1 & \beta_{11}^2\phi_{11}+\omega_5+\psi_1 & 0 \\ 0 & 0 & \beta_{22}\phi_{22} & 0 & 0 & \beta_{22}^2\phi_{22}+\omega_6+\psi_2 \end{pmatrix}$$

It seems that $\beta_{22}$ is not identifiable when both $\phi_{12}$, and $\beta_{12}$ equal zero. The only way to get at it is through $\phi_{22}$, which is not accessible at all. The conclusion is that $\beta_{22}$ is identifiable if either $\beta_{12} \neq 0$, or if both $\beta_{11}$ and $\phi_{12}$ are non-zero.

It is worth noting that the sufficient condition $\beta_{12} \neq 0$ was concealed until we actually set $\phi_{12} = 0$ and took another look at the covariance matrix. The general principle is that when the solution for a parameter in terms of $\sigma_{ij}$ quantities is a fraction, the parameter is identifiable at points in the parameter space where the denominator is non-zero. While it is tempting to think that identifiability fails where the denominator *is* zero, this need not be the case. If the model imposes equality constraints on the covariance matrix, there may be other ways to recover the parameter.

In our examination of identifiability for $\beta_{22}$, it was easy (with `Sagemath`) to re-calculate the covariance matrix with $\phi_{12} = 0$ to see if it was possible to solve for $\beta_{22}$ in that part of the parameter space. Doing this by hand would have been possible though tedious. For $\beta_{11}$ and $\beta_{12}$, hand calculation is almost out of the question because the denominators are so complicated; it's quite easy with `Sagemath` and the `sem` package.

```
# Look at beta11 and beta12.
show(beta11 == sol[beta11]);  show(beta12 == sol[beta12])
```

evaluate

$$\beta_{11} = \frac{\sigma_{16}\sigma_{34}-\sigma_{14}\sigma_{36}}{\sigma_{13}\sigma_{16}-\sigma_{12}\sigma_{36}}$$

$$\beta_{12} = \frac{\sigma_{13}\sigma_{14}\sigma_{16}-\sigma_{12}\sigma_{16}\sigma_{34}}{\sigma_{13}^2\sigma_{16}-\sigma_{12}\sigma_{13}\sigma_{36}}$$

To see where in the parameter space the denominators equal zero, we need to take the formulas for the $\sigma_{ij}$s in terms of the parameters, and substitute them into the denominators (just the denominators, of course). The `SigmaOfTheta` function of the `sem` package is designed to make this task easy. Given a covariance matrix that is a function of model parameters, `SigmaOfTheta` makes a dictionary that will allow any function of the $\sigma_{ij}$ variances and covariances to be evaluated at the model parameters. In the following, `SigmaOfTheta` is used to create a dictionary called `theta`, and the denominator of the solution for $\beta_{11}$ is put into `d1`. Then, `d1(theta)` gives `d1` as a function of the model parameters. The notation is simple and natural, partly because `theta` is a very good name for the dictionary. The `Simplify` function first expands an expression (multiplies it out), and then factors the result. I find it more helpful than `Sagemath`'s built-in `simplify` function, which is already applied to everything automatically anyway.

```
# Now examine denominators of the solutions to see exactly where in
# the parameter space they equal zero.
theta = SigmaOfTheta(Sigma)
d1 = denominator(sol[beta11])
Simplify(d1(theta))
```

evaluate

$(\phi_{12}^2 - \phi_{11}\phi_{22})\beta_{22}$

See how nice that was? The denominator is just $-|\mathbf{\Phi}_x|\beta_{22}$. Since $\mathbf{\Phi}_x$ is positive definite, the denominator will be zero if and only if $\beta_{22} = 0$.

```
# What if beta22=0?
Sigma(beta22=0)
```

evaluate

$$\begin{pmatrix}
\omega_1+\phi_{11} & \phi_{11} & \phi_{12} & \beta_{11}\phi_{11}+\beta_{12}\phi_{12} & \\
\phi_{11} & \omega_2+\phi_{11} & \phi_{12} & \beta_{11}\phi_{11}+\beta_{12}\phi_{12} & \\
\phi_{12} & \phi_{12} & \omega_3+\phi_{22} & \beta_{11}\phi_{12}+\beta_{12}\phi_{22} & \\
\beta_{11}\phi_{11}+\beta_{12}\phi_{12} & \beta_{11}\phi_{11}+\beta_{12}\phi_{12} & \beta_{11}\phi_{12}+\beta_{12}\phi_{22} & \beta_{11}^2\phi_{11}+2\,\beta_{11}\beta_{12}\phi_{12}+\beta_{12}^2\phi_{22}+\omega_4+\psi_1 & \beta_{11}^2\phi_{11}+2\,\beta_{11} \\
\beta_{11}\phi_{11}+\beta_{12}\phi_{12} & \beta_{11}\phi_{11}+\beta_{12}\phi_{12} & \beta_{11}\phi_{12}+\beta_{12}\phi_{22} & \beta_{11}^2\phi_{11}+2\,\beta_{11}\beta_{12}\phi_{12}+\beta_{12}^2\phi_{22}+\psi_1 & \beta_{11}^2\phi_{11}+2\,\beta_{11}\beta_{12}\phi_1 \\
0 & 0 & 0 & 0 &
\end{pmatrix}$$

The answer got cut off and there is no scrollbar in this document, but you can see that the only useable equations involving $\beta_{11}$ are variations of

$$\begin{aligned} \sigma_{42} &= \beta_{11}\phi_{11} + \beta_{12}\phi_{12} \\ \sigma_{43} &= \beta_{11}\phi_{12} + \beta_{12}\phi_{22} \end{aligned} \tag{B.7}$$

The parameters $\phi_{11}$ and $\phi_{12}$ are immediately identifiable, but $\phi_{22}$ is inaccessible when $\beta_{22} = 0$. This means that solving two linear equations in two unknowns won't work. The parameter of interest, $\beta_{11}$, can only be recovered if $\phi_{12} = 0$ as well as $\beta_{22} = 0$.

The conclusion is that $\beta_{11}$ is identifiable provided that $\beta_{22} \neq 0$ or $\beta_{22} = \phi_{12} = 0$.

```
# Study the identifiability of beta12
d2 = denominator(sol[beta12]); Simplify(d2(theta))
```

evaluate

$(\phi_{12}^2 - \phi_{11}\phi_{22})\beta_{22}\phi_{12}$

It looks like we need both $\beta_{22}$ and $\phi_{12}$ non-zero. Earlier, we calculated the covariance matrix $\mathbf{\Sigma}$ with $\phi_{12} = 0$ but not $\beta_{22}$. In that case,

$$\beta_{12} = \frac{\sigma_{46}}{\sigma_{36}} = \frac{\beta_{12}\beta_{22}\phi_{22}}{\beta_{22}\phi_{22}}.$$

If $\beta_{22}$ but not $\phi_{12} = 0$, we are back to the two linear equations (B.7). We can't solve the two equations for $\beta_{11}$ and $\beta_{12}$ because $\phi_{22}$ isn't identifiable. However, we can recover $\beta_{12}$ if $\beta_{11} = 0$. Okay, so far we have established that $\beta_{12}$ is identifiable if

- $\beta_{22} \neq 0$, or
- $\beta_{22} = 0$ and $\phi_{12} \neq 0$ and $\beta_{11} = 0$.

Now let's see what happens if both $\beta_{22}$ and $\phi_{12}$ equal zero.

```
# If both beta22 and phi12 equal zero,
Sigma(beta22=0,phi12=0)
```

evaluate

$$\begin{pmatrix} \omega_1+\phi_{11} & \phi_{11} & 0 & \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & 0 \\ \phi_{11} & \omega_2+\phi_{11} & 0 & \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & 0 \\ 0 & 0 & \omega_3+\phi_{22} & \beta_{12}\phi_{22} & \beta_{12}\phi_{22} & 0 \\ \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & \beta_{12}\phi_{22} & \beta_{11}^2\phi_{11}+\beta_{12}^2\phi_{22}+\omega_4+\psi_1 & \beta_{11}^2\phi_{11}+\beta_{12}^2\phi_{22}+\psi_1 & 0 \\ \beta_{11}\phi_{11} & \beta_{11}\phi_{11} & \beta_{12}\phi_{22} & \beta_{11}^2\phi_{11}+\beta_{12}^2\phi_{22}+\psi_1 & \beta_{11}^2\phi_{11}+\beta_{12}^2\phi_{22}+\omega_5+\psi_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_6+\psi_2 \end{pmatrix}$$

The sign of $\beta_{12}$ can be identified but not the value, because $\phi_{22}$ can't be recovered.

We now have a detailed picture of the identifiability of the key parameters $\beta_{11}$, $\beta_{12}$ and $\beta_{22}$, a picture that would be just too much work to obtain without a symbolic math program like `Sagemath`. If at this point you are wishing that you didn't know so much about the identifiability of the $\beta_{ij}$, think again. For example, it would be natural to try testing $H_0 : \beta_{11} = \beta_{12} = \beta_{22} = 0$ with a likeihood ratio test, but this would be a disaster because the parameters are not identifiable under the null hypothesis.

Next, we will obtain explicit formulas for the model-induced equality constraints on the variances and covariances of the observable data, by substituting solutions for the parameters into the equations that were set aside. Results without an = sign are polynomials implicitly set to zero.

```
for item in aside: factor(item(sol))
```

evaluate

$$\sigma_{14} - \sigma_{15}$$

$$\sigma_{13} - \sigma_{23}$$

$$\sigma_{14} - \sigma_{24}$$

$$\sigma_{14} - \sigma_{25}$$

$$\sigma_{16} - \sigma_{26}$$

$$\sigma_{34} - \sigma_{35}$$

$$\frac{\sigma_{16}\sigma_{34}}{\sigma_{13}} = \sigma_{56}$$

$$\frac{\sigma_{16}\sigma_{34}}{\sigma_{13}} = \sigma_{46}$$

If the last two polynomials are multiplied through by $\sigma_{13}$, we get

$$\sigma_{16}\sigma_{34} = \sigma_{13}\sigma_{56} = \sigma_{13}\sigma_{56},$$

which is a nice way to express the constraints because the statement remains true when the denominator $\sigma_{13} = \phi_{12}$ equals zero. This claim is verified by evaluating the $\sigma_{ij}$ quantities at the model parameters, as follows.

```
# Are constraints still true when sigma13=0?
equal3 = [sigma16*sigma34, sigma13*sigma56, sigma13*sigma46]
for item in equal3: show(item(theta))
```

evaluate

$$(\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22}\phi_{12}$$
$$(\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22}\phi_{12}$$
$$(\beta_{11}\phi_{12} + \beta_{12}\phi_{22})\beta_{22}\phi_{12}$$

The equality constraints we have worked so hard to obtain can be quite valuable in data analysis. If the model is re-parameterized by making $\psi_2 + \omega_6$ a single parameter, we have 21 covariance structure equations in 13 unknown parameters. The likelihood ratio chi-squared test for goodness of fit will have $21 - 13 = 8$ degrees of freedom, and the null hypothesis is that exactly those eight equality constraints hold. If the model does not fit, the constraints can be tested individually to track down why the model does not fit, and suggest how it might be fixed up.

**Groebner Basis** While there is no doubt that `Sagemath` can make life easier by reducing the computational burden of studying a model, it's still too bad that so much thinking is required. In particular, in order to prove identifiability by obtaining explicit solutions for the parameters, you need to figure out which equations are redundant so you can give `solve` a system that has a general solution. To do this, you almost have to solve the

equations by hand, or at least look at them carefully and decide how you would proceed if you were going to do it by hand.

An alternative that sometimes works (but not always) is to apply Groebner basis methods. If you subtract the $\sigma_{ij}$ from both sides of the covariance structure equations, you get a set of multivariate polynomials, and the roots of those polynomials are the solutions of the equations. A Groebner basis is a set of polynomials having the same roots as the input set, but they are typically much easier to solve. See the documentation for the `GroebnerBasis` function on page 686 for more details.

Input to the `GroebnerBasis` function is a list of polynomials and a list of variables. The polynomials correspond to the covariance structure equations, and are produced as an option by the `SetupEqns` function. The "variables" are the model parameters *and* the $\sigma_{ij}$ quantities. Ordering of the list of variables is very important. The $\sigma_{ij}$ come last. The model parameters go before the $\sigma_{ij}$ quantities, usually in reverse order of how interesting or important they are.

If the $\sigma_{ij}$ quantities are last in the input list of variables and there are equality constraints among them, the first set of polynomials in the Groebner basis will involve only $\sigma_{ij}$s. Setting these to zero gives you the equality constraints. Then come the model parameters. If the first parameter (the last you mentioned in the list of variables) is identifiable it will appear by itself, accompanied only by covariances. If fortune smiles, the next polynomial will involve two model parameters, and so on.

The Groebner basis algorithm simplifies the input by multiplying polynomials together and then adding multiples of polynomials to other polynomials. Depending on the size and structure of the problem, the number of polynomials can become very large before finally reducing to a small set with a nice simple form. As a mathematical certainty, the target (a Groebner basis) exits and algorithm terminates at the right answer, but in practice this may not happen during your lifetime. As I said, Groebner basis does not always work, but when it works it is beautiful. In the following, we will just hand the whole system of Example B.1.2 to the `GroebnerBasis` function, warts and all.

Notice how the list of parameters is reversed, so that the $\beta_{ij}$ come last and therefore the solutions for those parameters will emerge first. The $\sigma_{ij}$ quantities are reversed as well. This makes the output of the `GroebnerBasis` function easier to compare with earlier work. I may as well explain why, because it sheds light on how Groebner basis works in practice, as well as features of some other functions in the `sem` package.

The `GroebnerBasis` function requires $\sigma_{ij}$ quantities as input, and I do not want to type in the names of the 21 unique elements. `Parameters(SymmetricMatrix(6,'sigma'))` does the trick. When I examine a covariance matrix, my preference is to look at the upper triangle, scanning from left to right. For this reason, the `SymmetricMatrix` function, which produces a matrix containing only unique elements, puts copies of the upper triangle into the lower triangle. So, for example, row 4 column 2 contains $\sigma_{24}$. In this example, the `Parameters` function detects that the matrix is symmetric, and returns the main diagonal and the upper triangle, from left to right and top to bottom.

When I was deciding which equations to set aside, I followed my usual practice of looking at the upper triangle left to right an top to bottom. If I discovered an equation that was redundant with the earlier ones, I selected it for deletion. When I did this I was

just trying to be systematic and not thinking about Groebner basis, but it was fortunate. Groebner basis works from the end of the list of input variables (parameters and $\sigma_{ij}$ quantities). When a variable appears in the list of output polynomials for the first time, it will tend to appear with variables closer to the end of the list.

With the $\sigma_{ij}$ reversed as well as at the end, the list of variables looks like $\ldots \sigma_{13}, \sigma_{12}, \sigma_{11}]$. This means, for example, that if $\sigma_{14} = \sigma_{15}$, other polynomials in the output (and the corresponding solutions for the model parameters) will be in terms of $\sigma_{14}$ rather than $\sigma_{15}$. That's exactly the way I did it. It is only because of this happy coincidence that we have a prayer of checking that the Groebner basis results are consistent with what we did before without doing a lot of work.

```
param2 = copy(param) # Work with a copy to avoid changing the original.
param2.reverse() # Reversed order of interest
sigmaij = Parameters(SymmetricMatrix(6,'sigma'))
sigmaij.reverse() # Reverse the sigma_ij too
param2.extend(sigmaij) # Put sigma_ij values at the end
polynoms = SetupEqns(Sigma,poly=True) # Covariance structure polynomials
# Throw the whole thing at GroebnerBasis.
basis1 = GroebnerBasis(polynoms,param2)
```

evaluate

```
Defining tT1, tT2, tT3, tT4, tT5, tT6, tT7, tT8, tT9, tT10, tT11, tT12,
tT13, tT14, tT15, tT16, tT17, tT18, tT19, tT20, tT21, tT22, tT23, tT24,
tT25, tT26, tT27, tT28, tT29, tT30, tT31, tT32, tT33, tT34, tT35
```

To my surprise, it finished almost immediately. Take a look.

```
for item in basis1: show(item)
```

evaluate

$$-\sigma_{14} + \sigma_{15}$$
$$-\sigma_{13} + \sigma_{23}$$
$$-\sigma_{14} + \sigma_{24}$$
$$-\sigma_{14} + \sigma_{25}$$
$$-\sigma_{16} + \sigma_{26}$$
$$-\sigma_{34} + \sigma_{35}$$
$$-\sigma_{16}\sigma_{34} + \sigma_{13}\sigma_{46}$$
$$-\sigma_{46} + \sigma_{56}$$
$$-\beta_{11}\sigma_{13}\sigma_{16} + \beta_{11}\sigma_{12}\sigma_{36} + \sigma_{16}\sigma_{34} - \sigma_{14}\sigma_{36}$$
$$\beta_{11}\sigma_{12} + \beta_{12}\sigma_{13} - \sigma_{14}$$
$$\beta_{12}\sigma_{16}\sigma_{34} + \beta_{11}\sigma_{12}\sigma_{46} - \sigma_{14}\sigma_{46}$$
$$\beta_{11}\sigma_{16} + \beta_{12}\sigma_{36} - \sigma_{46}$$
$$\beta_{22}\sigma_{13} - \sigma_{16}$$
$$\beta_{22}\sigma_{34} - \sigma_{46}$$
$$\beta_{11}\beta_{22}\sigma_{12} - \beta_{22}\sigma_{14} + \beta_{12}\sigma_{16}$$
$$\phi_{11} - \sigma_{12}$$
$$\phi_{12} - \sigma_{13}$$
$$\phi_{22}\sigma_{16} - \sigma_{13}\sigma_{36}$$
$$-\sigma_{34}\sigma_{36} + \phi_{22}\sigma_{46}$$
$$\beta_{11}\phi_{22}\sigma_{12} - \beta_{11}\sigma_{13}^2 - \phi_{22}\sigma_{14} + \sigma_{13}\sigma_{34}$$
$$\beta_{12}\phi_{22} + \beta_{11}\sigma_{13} - \sigma_{34}$$
$$\beta_{22}\phi_{22} - \sigma_{36}$$
$$\beta_{11}\sigma_{14} + \beta_{12}\sigma_{34} + \psi_1 - \sigma_{45}$$
$$\omega_1 - \sigma_{11} + \sigma_{12}$$
$$\omega_2 + \sigma_{12} - \sigma_{22}$$
$$\omega_3 + \phi_{22} - \sigma_{33}$$
$$\omega_4 - \sigma_{44} + \sigma_{45}$$
$$\omega_5 + \sigma_{45} - \sigma_{55}$$
$$\beta_{22}\sigma_{36} + \omega_6 + \psi_2 - \sigma_{66}$$

These polynomials have the same roots as the input set. There are more polynomials than in the input set, but not hundreds — something that can easily happen. The first eight polynomials in the Groebner basis involve only $\sigma_{ij}$ quantities. Comparing them to the constraints we obtained earlier (see page 674), we see that they are exactly the same, except just a little better. The first six constraints are even in the same order. For the last two, the Groebner basis is better because $\frac{\sigma_{16}\sigma_{34}}{\sigma_{13}} = \sigma_{56}$ and $\frac{\sigma_{16}\sigma_{34}}{\sigma_{13}} = \sigma_{46}$ do imply $\sigma_{46} = \sigma_{56}$. A simple equality between covariances is preferable to a product set equal to another product.

The next polynomial involves $\beta_{11}$, which appears first because it is the last parameter on the input list. Setting it equal to zero and solving yields the solution on page 669. Next comes not one but three polynomials involving $\beta_{11}$ and $\beta_{12}$. If the solution for $\beta_{11}$ in terms of $\sigma_{ij}$ is substituted into the first polynomial yields the solution for $\beta_{12}$ on page 669. The other two yield alternative solutions for $\beta_{12}$; these solutions are also correct. Setting any two of them equal yields a complicated equality constraint on the $\sigma_{ij}$ — a constraint

that causes `solve` to think the whole system has no general solution. There is nothing new though, because these constraints are implied by the constraints located earlier.

The next two polynomials involve $\beta_{22}$ and $\sigma_{ij}$ quantities. Setting the first one equal to zero yields the solution for $\beta_{22}$ on page 669. Setting the second one equal to zero yields the "superior" solution on page 670.

This is the way it goes. There may be multiple ways of solving for a particular parameter in terms of $\sigma_{ij}$ quantities and parameters that have come before. Because of the way the variables are ordered in this example, the first polynomial involving a particular parameter always corresponds to one of the solutions given on page 669. When more than one way of solving for a parameter is indicated by the Gorebner basis, sometimes one of them is preferable because it's simpler or applies in more of the parameter space; sometimes not. Almost always, the polynomials are simple enough that one can verify the existence of a solution by inspection without actually calculating it.

The last polynomial in the set is $\beta_{22}\sigma_{36} + \omega_6 + \psi_2 - \sigma_{66}$. This is the first time either $\psi_2$ or $\omega_6$ appears, and the fact that they appear together tells you they are not identifiable. They come last not because they are non-identifiable, but because one of them, $\omega_6$, is first in the list of variables. The way they appear together as a sum reflects the way they are non-identifiable.

When Groebner basis works, it is hard to exaggerate how excellent it is. Equality constraints involving the $\sigma_{ij}$ quantities appear immediately without all the hard work, and identifiability or lack of identifiability can usually be verified by inspection. It is really wonderful that the equality constraints implied by models whose parameters are non-identifiable can be so easy to obtain, because it makes these models testable (falsifiable) without finding a way to re-parameterize them in a way that preserves the equality constraints.

But as I have mentioned several times, the Groebner basis approach does not always work. When it fails, it usually fails by not finishing. I have had most trouble with unrestricted factor analysis, and multi-stage models of the $a$ influences $b$ influences $c$ variety — the kind for which identifiablity would be established by the Acyclic Rule. It seems likely that this case could be resolved by ordering the variables better.

### B.2.2 Function Documentation

To use the `sem` functions, you must load them once per session.

```
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
# load('~/sem.sage') # To load a local version in your home directory
```

evaluate

After the package is loaded, `Contents()` will display a list of the available functions. For help on a particular function, type the function name followed by a question mark, like "`PathCov?`"

1. Matrix Creation

   1a) `DiagonalMatrix(size,symbol='psi')`

   1b) `GeneralMatrix(nrows,ncols,symbol)`

   1c) `IdentityMatrix(size)`

   1d) `SymmetricMatrix(size,symbol,corr=False)`

   1e) `ZeroMatrix(nrows,ncols)`

2. Covariance Matrix Calculation

   2a) `EqsCov(beta,gamma,Phi,oblist,simple=True)`

   2b) `FactorAnalysisCov(Lambda,Phi,Omega)`

   2c) `NoGammaCov(Beta,Psi)`

   2d) `PathCov(Phi,Beta,Gamma,Psi,simple=True)`

   2e) `RegressionCov(Phi,Gamma,Psi,simple=True)`

3. Manipulation

   3a) `GroebnerBasis(polynomials,variables)`

   3b) `LSTarget(M,x,y)`

   3c) `Parameters(M)`

   3d) `SigmaOfTheta(M,symbol='sigma')`

   3e) `Simplify(x)`

4. Utility

   4a) `BetaCheck(Beta)`

   4b) `Contents()`

   4c) `CovCheck(Psi)`

   4d) `MultCheck(Beta,Psi)`

   4e) `Pad(M)`

For each function, explanation is followed by the function definition (without the documentation string).

1. **Matrix Creation**

   (a) `DiagonalMatrix(size,symbol='psi',double=False)`

   This function creates a diagonal symbolic matrix (size by size) with Greek-letter symbols (default $\psi$), and single subscripts. Double subscripts are optional. The arguments of the function are

- `size`: Number of rows, equal to number of columns
- `symbol`: A string containing the root. It is usually a Greek letter, but does not have to be. Notice the single quotes in the examples below.
- `double`: Should diagonal elements be doubly subscripted? Default is no, use single subscripts.

Examples:

```
DiagonalMatrix(4) # Will have psi1 to psi4 on main diagonal
DiagonalMatrix(4,double=True) # Will have psi11 to psi44 on main diagon
DiagonalMatrix(2,'phi')
DiagonalMatrix(2,'phi',True)
DiagonalMatrix(size=2,symbol='phi')
```

```
DiagonalMatrix(3,'omega')
```

evaluate

$$\begin{pmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{pmatrix}$$

Here is the function definition without the documentation string.

```
def DiagonalMatrix(size,symbol='psi',double=False):
    M = identity_matrix(SR,size) # SR stands for Symbolic Ring
    for i in interval(1,size):
        subscr = str(i)
        if double: subscr = subscr+str(i)
        M[i-1,i-1] = var(symbol+subscr)
    return M
```

(b) `GeneralMatrix(nrows,ncols,symbol)`

This function returns a general symbolic matrix containing symbols with specified root, usually a Greek letter. In each cell of the matrix are the root symbol and subscript(s). The argumments are

- `nrows`: Number of rows
- `ncols`: Number of columns
- `symbol`: A string containing the root. It is usually a Greek letter, but does not have to be. Notice the single quotes in the examples below.

Because it is difficult (impossible?) to get good doubly subscripted variables with the two subscripts separated by a comma, there is potential ambiguity when either `nrows` or `ncols` gets into double figures. What is $\gamma_{111}$? Is it $\gamma_{1,11}$ or $\gamma_{11,1}$? For this reason, if either the number of rows or the number of columns exceeds 9, the contents of the matrix returned by this function are singly subscripted.

Examples:

```
GeneralMatrix(6,2,'lambda')
GeneralMatrix(11,3,'L')
GeneralMatrix(3,4,'gamma')
GeneralMatrix(nrows=3,ncols=4,symbol='gamma')
```

```
Gamma = GeneralMatrix(nrows=3,ncols=5,symbol='gamma')
Gamma
```

evaluate

$$\begin{pmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & \gamma_{14} & \gamma_{15} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & \gamma_{24} & \gamma_{25} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & \gamma_{34} & \gamma_{35} \end{pmatrix}$$

Here is the function definition without the documentation string.

```
def GeneralMatrix(nrows,ncols,symbol):
    M = matrix(SR,nrows,ncols) # SR is the Symbolic Ring
    if nrows < 10 and ncols < 10:
        for i in interval(1,nrows):
            for j in interval(1,ncols):
                M[i-1,j-1] = var(symbol+str(i)+str(j))
    else:
        index=1
        for i in interval(1,nrows):
            for j in interval(1,ncols):
                M[i-1,j-1] = var(symbol+str(index))
                index = index+1
    return M
```

(c) `IdentityMatrix(size)`

This function returns a symbolic identity matrix of specified size. It's the same as `identity_matrix(SR,size)`.

Example: `IdentityMatrix(3)`

Here is the function definition without the documentation string.

```
def IdentityMatrix(size):
    M = identity_matrix(SR,size) # SR is the Symbolic Ring
    return M
```

(d) `SymmetricMatrix(size,symbol,corr=False)`

This function returns a square symmetric matrix of the symbolic type, containing symbols with a specified root, usually a Greek letter. In each cell of the

matrix is the root symbol with subscript(s). The matrix contains only unique elements; the lower triangle contains symbols from the upper triangle, so that the element in row 5 and column 2 is something like $\sigma_{25}$.

The arguments of the function are

- `size`: Number of rows, equal to number of columns
- `symbol`: A string containing the root. It is usually a Greek letter, but does not have to be. Notice the single quotes in the examples below.
- `corr`: A logical variable (True or False) specifying whether it's a correlation matrix. If True, there are ones on the main diagonal. This argument is optional, with a default of False.

Examples:

```
SymmetricMatrix(6,'phi')
SymmetricMatrix(11,'psi')
SymmetricMatrix(4,'rho',True)
SymmetricMatrix(size=4,symbol='rho',corr=True)
```

Because it is difficult or maybe even impossible with `Sagemath` to get good doubly subscripted variables with the two subscripts separated by a comma, there is potential ambiguity when either `nrows` or `ncols` gets into double figures. What is $\sigma_{111}$? Is it $\sigma_{1,11}$ or $\sigma_{11,1}$? For this reason, if either the number of rows or the number of columns exceeds 9, the contents of the matrix returned by this function are singly subscripted. In this case the diagonal elements are numbered last, which is usually what you want once you get used to it.

```
Phi = SymmetricMatrix(5,'phi'); Phi
```

evaluate

$$\begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ \phi_{12} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \\ \phi_{13} & \phi_{23} & \phi_{33} & \phi_{34} & \phi_{35} \\ \phi_{14} & \phi_{24} & \phi_{34} & \phi_{44} & \phi_{45} \\ \phi_{15} & \phi_{25} & \phi_{35} & \phi_{45} & \phi_{55} \end{pmatrix}$$

Here is the function definition without the documentation string.

```
def SymmetricMatrix(size,symbol,corr=False):
    M = identity_matrix(SR,size) # SR is the Symbolic Ring
    if size < 10:
        for i in interval(1,size):
            for j in interval(i+1,size):
                M[i-1,j-1] = var(symbol+str(i)+str(j))
                M[j-1,i-1] = M[i-1,j-1]
        if not corr:
```

```
            for i in interval(1,size):
                M[i-1,i-1] = var(symbol+str(i)+str(i))
    else:
        index=1
        for i in interval(1,size):
            for j in interval(i+1,size):
                M[i-1,j-1] = var(symbol+str(index))
                M[j-1,i-1] = M[i-1,j-1]
                index = index+1
        if not corr:
            for i in interval(1,size):
                M[i-1,i-1] = var(symbol+str(index))
                index = index+1
    return M
```

(e) `ZeroMatrix(nrows,ncols)`

This function returns a symbolic matrix with specified number of rows and nummber of columns, full of zeros. It's the same as the sage function `matrix(SR,size)`. The `ZeroMatrix` function is particularly useful for setting up parameter matrices that consist mostly of zeros.

Example: `ZeroMatrix(4,4)`

Here is the function definition without the documentation string.

```
def ZeroMatrix(nrowz,ncolz):
    M = matrix(SR,nrowz,ncolz) # SR is the Symbolic Ring
    return M
```

2. **Covariance Matrix Calculation**

(a) `EqsCov(beta,gamma,Phi,oblist,simple=True)`

The `EqsCov` function is alphabetically first in the category of covariance matrix calculation, but it is among the less frequently used. It calculates the covariance matrix of an obvservable data vector for the EQS model of Bentler and Weeks (1980). The EQS model makes no distinction between error terms and other exogenous variables, and there is no notationalq difference between latent and observable variables. Instead, the covariance matrix of all variables in the model is calculated, and then the rows end columns corresponding to the observable variables are selected to form $\mathbf{\Sigma}$ , the common covariance matrix of the $n$ observable data vectors.

The model equations are

$$\boldsymbol{\eta}_i = \boldsymbol{\beta}\boldsymbol{\eta}_i + \boldsymbol{\gamma}\boldsymbol{\xi}_i,$$

with $cov(\boldsymbol{\xi}_i) = \mathbf{\Phi}$.

The exogenous variables (including error terms) are in the vector $\boldsymbol{\xi}_i$, which is spelled "xi" and pronounced more or less like the letter "c." The endogenous

variables are in $\boldsymbol{\eta}_i$, which is spelled "eta" and pronounced like "I can't believea I atea the whole thing." Because $\boldsymbol{\xi}_i$ includes error terms as well as ordinary exogenous variables, the `EqsCov` function is useful for calculating the covariance matrix for pathological but disturbingly realistic models in which exogenous variables are correlated with error terms, or measurement errors are correlated with errors in the latent variable model. Other functions in the `sem` package are based on standard models which do not admit this possibility.

In the `EqsCov` function, $\mathbf{V} = cov\begin{pmatrix} \boldsymbol{\eta}_i \\ \boldsymbol{\xi}_i \end{pmatrix}$ is first calculated, and then the covariance matrix $\boldsymbol{\Sigma}$ is formed by selecting rows and columns corresponding to the observable variables.

The indices of the observable variables are given in the function argument `oblist`. The indices start with one, not zero. Following EQS conventions, *the endogenous variables come first in the list of variables* $(\boldsymbol{\eta}_i, \boldsymbol{\xi}_i)$.

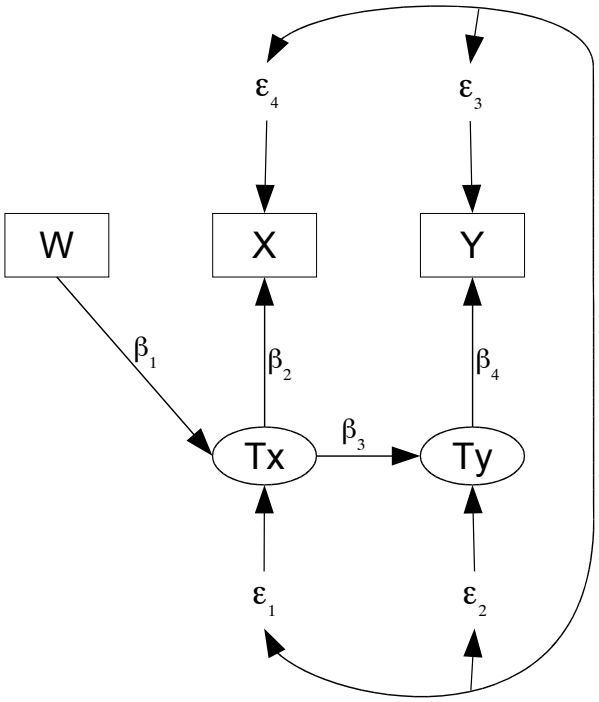
The arguments of the function are

- `beta`: A square matrix containing the coefficients from each element of eta to each other element. Number of rows equals number of columns equals number of exogenous variables, including error terms. Diagonal elements of Beta should be zeros.
- `gamma`: A matrix of regression coefficients linking each exogenous ($\xi$) variable to each endogenous ($\eta$) variable. There is one row for each eta variable and one column for each $\xi$ variable. Thus, the number of rows in `gamma` must equal the number of rows (and columns) in `beta`, and the number of columns in `gamma` must equal the number of rows (and columns) in `Phi`.
- `Phi`: The variance-covariance matrix of the exogenous variables $\boldsymbol{\eta}_i$.
- `oblist`: List of indices of observable variables. First index is one, not zero. May be in any order. Following EQS conventions, the endogenous variables come first in the list of variables $(\boldsymbol{\eta}_i, \boldsymbol{\xi}_i)$. So the variable with index one is the first *endogenous* variable.
- `simple`: Should the covariance matrix be simplified? Simplification consists of expanding and then factoring all the elements of $\boldsymbol{\Sigma}$. This is time consuming, but usually worth it. The default for this optional argument is True.

Example: `EqsCov(beeta,gammma,fee,pickout)`

The following more detailed example is an extension of Example 0.5.1 on page 36, which was about the relationship between income and credit card debt among real estate agents. In the path diagram of Figure B.3, $X$ is reported income ($Tx$ measured with error), $Y$ is reported credit card debt ($Ty$ measured with error), and $W$ is local average selling price of a resale home (real estate agents typically get a percentage of the the selling price). Because of numerous omitted variables, the error terms are all correlated with one another.

Figure B.3: Massively correlated error terms




In the notation of the EQS model,

$$\boldsymbol{\xi}_i = \begin{pmatrix} \epsilon_{i,1} \\ \epsilon_{2,1} \\ \epsilon_{i,3} \\ \epsilon_{i,4} \\ W_i \end{pmatrix} \quad \text{and} \quad \boldsymbol{\eta}_i = \begin{pmatrix} Tx_i \\ Ty_i \\ X_i \\ Y_i \end{pmatrix}$$

```
# In EsqCov, eta = beta eta + gamma xi,   with cov(xi) = Phi
# eta' = (Tx,Ty,X,Y) and xi' = (epsilon1,epsilon2,epsilon3,epsilon4,W)
B = ZeroMatrix(4,4) # beta
B[1,0] = var('beta3') ; B[2,0] = var('beta2') ; B[3,1] = var('beta4')
G[0,0] = 1; G[0,4] = var('beta1'); G[1,1] = 1; G[2,3] = 1; G[3,2] = 1
P = SymmetricMatrix(5,'psi'); P[4,4]=var('phi') # This is the Phi matrix
# No correlations between W and the errors
for j in interval(0,3):
    P[j,4] = 0
    P[4,j] = 0
P
```

evaluate

$$\begin{pmatrix} \psi_{11} & \psi_{12} & \psi_{13} & \psi_{14} & 0 \\ \psi_{12} & \psi_{22} & \psi_{23} & \psi_{24} & 0 \\ \psi_{13} & \psi_{23} & \psi_{33} & \psi_{34} & 0 \\ \psi_{14} & \psi_{24} & \psi_{34} & \psi_{44} & 0 \\ 0 & 0 & 0 & 0 & \phi \end{pmatrix}$$

```
pickout = 9,3,4 # Indices of observable variables, order eta, xi
Sigma = EqsCov(B,G,P,pickout); Sigma
```

evaluate

$$\begin{pmatrix} \phi & & \beta_1\beta_2\phi & \\ \beta_1\beta_2\phi & & \beta_1^2\beta_2^2\phi + \beta_2^2\psi_{11} + 2\,\beta_2\psi_{14} + \psi_{44} & \beta_1^2\beta_2\beta_3\beta_4\phi + \\ \beta_1\beta_3\beta_4\phi & \beta_1^2\beta_2\beta_3\beta_4\phi + \beta_2\beta_3\beta_4\psi_{11} + \beta_2\beta_4\psi_{12} + \beta_3\beta_4\psi_{14} + \beta_2\psi_{13} + \beta_4\psi_{24} + \psi_{34} & \beta_1^2\beta_3^2\beta_4^2\phi + \beta_3^2 \end{pmatrix}$$

(b) `FactorAnalysisCov(Lambda,Phi,Omega)`

(c) `NoGammaCov(Beta,Psi)`

(d) `PathCov(Phi,Beta,Gamma,Psi,simple=True)`

(e) `RegressionCov(Phi,Gamma,Psi,simple=True)`

3. **Manipulation**

(a) `GroebnerBasis(polynomials,variables)`

(b) `LSTarget(M,x,y)`

(c) `Parameters(M)`

(d) `SigmaOfTheta(M,symbol='sigma')`

(e) `Simplify(x)`

4. **Utility**

(a) `BetaCheck(Beta)`

(b) `Contents()`

(c) `CovCheck(Psi)`

(d) `MultCheck(Beta,Psi)`

(e) `Pad(M)`

Indices of arrays and vectors in `Sagemath` start with zero, which can be a minor irritant to those of us who are used to counting on our fingers. This function returns a "padded" version of a matrix by inserting a row zero and a column zero consisting entirely of zeros. This makes it more convenient to refer to elements of the matrix.

Here is the function definition without the documentation string.

```
def Pad(M):

    """
    Pad by making first row and first col all zeros, so it is

    Argument: A matrix that needs padding
    Result: A padded matrix, with one additional row and one additional
            column.

    Example

    SIGMA = Pad(Sigma)



    """
    nrowz = M.nrows(); nrowz = nrowz+1 # Strange work-around
    ncolz = M.ncols(); ncolz = ncolz+1
    padM = matrix(SR,nrowz,ncolz)
    for i in interval(1,M.nrows()):
        for j in interval(1,M.ncols()):
            padM[i,j] = M[i-1,j-1]
    return padM
```

```
Phi = SymmetricMatrix(5,'phi')
PadPhi = Pad(Phi); PadPhi
```

evaluate

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ 0 & \phi_{12} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \\ 0 & \phi_{13} & \phi_{23} & \phi_{33} & \phi_{34} & \phi_{35} \\ 0 & \phi_{14} & \phi_{24} & \phi_{34} & \phi_{44} & \phi_{45} \\ 0 & \phi_{15} & \phi_{25} & \phi_{35} & \phi_{45} & \phi_{55} \end{pmatrix}$$

Here is the function definition without the documentation string.

## B.3 Using Sagemath on your Computer

Sagemath has a browser interface, which means you interact with it through an ordinary Web browser[11]. This means that the actual Sagemath software can reside either on your computer or a remote server. In practice, there are three possibilities:

1. You may use Sagemath free of charge on computers maintained by the Sagemath development group. To do it this way, go to http://sagenb.com, set up a free account, and start using Sagemath. This is the easiest way to get started, but be aware that many people may be trying to use the service at the same time. My experience is that performance is sometimes quick and pleasant (for example, during the summer), and sometimes very slow. So this is an excellent way to give Sagemath a try and it's very handy for occasional use, but depending on it to do homework assignments is a bit risky.

2. You can connect to Sagemath on a server at your university or organization, provided that someone has gone to the trouble to set it up. If you can use Sagemath this way, you are fortunate, and you only have some minor font issues to take care of. These are discussed below.

3. You can download and install Sagemath on your own computer. You still use a Web browser, but the Web server is your own machine, and it serves only you. It's pretty straightforward, but the details depend on your operating system. Some of these details may change, because the Sagemath developers are constantly working (without payment) to improve the package. They also are responding to the actions of various companies like Apple, Google and Microsoft.

**Mac OS and Linux** There are two steps. First, go to http://www.sagemath.org, download the software, and install it as usual. As of March 2013, there was almost[12] nothing out of the ordinary for Mac OS, and this appeared to be the case for linux as well.

The second step is probably needed if you do *not* already have LaTeX installed, which will be the case for many students. Even if you do have LaTeX installed, the following is very helpful if you plan to use Sagemath on the servers at http://sagenb.com, even occasionally. Go to

http://www.math.union.edu/ dpvc/jsMath/download/jsMath-fonts.html,

download the jsMath fonts, and install them. You should only download one set of fonts. To install, Mac users can open the System folder, open the library sub-folder, and then drag the fonts to the Fonts sub-sub folder. You may need to click "Authenticate" and type your password. A re-start will be required before the new fonts are available.

[11] The Sagemath website says Mozilla Firefox and Google Chrome are recommended, and if you are a Windows user, you should believe it. In a Mac environment, I have had no trouble with Safari.

[12] Under Mac OS, the "App" version of the software is recommended. It works like any other Mac application. The first time you start it, you might have to wait

**Microsoft Windows** As mentioned earlier, Sagemath incorporates a number of other open source math programs, and makes them work together using a common interface. This marvelous feat, which is accomplished mostly with Python scripts, depends heavily on features that are part of the linux and unix operating systems, but are missing from Microsoft Windows. This makes it difficult or perhaps actually impossible to construct a native version of Sagemath for Windows. The current (and possibly final) solution is to run Sagemath in a *virtual machine* – a set of software instructions that act like a separate computer within Windows. The virtual machine uses the linux operating system, and has Sagemath preinstalled. The http://www.sagemath.org website calls it the "Sage appliance."

The software that allows the virtual machine to function under Windows is Oracle Corporation's free open-source VirtualBox, and you need to install that first. Start at http://wiki.sagemath.org/SageApplianceInstallation, and follow the directons. You will see that the first step is to download VirtualBox.

Then, go to http://wiki.sagemath.org/SageAppliance, and follow the directions there. It is *highly* recommended that you set up a folder for sharing files between Windows and the Sage appliance, because a good way of printing your Sagemath output depends on it. Follow *all* the directions, including the part about resetting the virtual machine.

Now you are ready to use Sagemath and see your output on screen. Printing under Windows is a separate issue, but it's easy once you know how.

**Printing Under Windows** The virtual machine provided by VirtualBox is incomplete by design; it lacks USB support[13]. So, most printers don't work easily. I know of four ways to print, and I have gotten the first three to work. The fourth way is speculation only and I don't intend to try it. The methods are ordered in terms of my personal preference.

1. In the Sagemath appliance, click on the printer icon or press the right mouse button and choose Print from the resulting menu. The default will be to Save as PDF. To choose the location to save the file, click on File System, then media, then the name of the shared folder[14]. Click Save. In Windows, go to the shared folder and print the pdf file[15]. An advantage of this method is that you don't need to install any fonts, because the jsMath fonts are already installed in the linux system of the Sage Applicance.

2. For this method, you do need to install the jsMath fonts under Windows. Go to

[13]Presumably this is a strategic decision by Oracle Corporation. As of this writing, USB support is available from Oracle as a separate free add-on. It's free to individual users for their personal use, meaning nobody can legally re-sell a virtual machine that includes it without paying Oracle a royalty. Sagemath would give it away and not sell it, but the developers strongly prefer to keep Sagemath fully free under the GNU public license.

[14]You set up the shared folder when you installed the Sage applicance.

[15]When working with Sagemath in a Windows environment, it may be helpful to keep the shared folder open in Windows Explorer. As soon as you save the file you want to print, you will see it appear in Windows Explorer.

http://www.math.union.edu/ dpvc/jsMath/download/jsMath-fonts.html,

download the jsMath fonts, and install them; A darkness level of 25 is good. To install under Windows 7, I needed to double-click on each font individually and click install. More experienced Windows users may be able to install the fonts some other way, or perhaps it's easier with later versions of Windows. A re-start is required.

Now once the jsMath fonts are installed, note that you can reach the Sagemath runnning in your virtual machine from Windows. Minimize the browser in the virtual machine, and open Firefox or Chrome under Windows. Go to https://localhost:8000. Now you can do whatever calculations you wish and print as usual. When you are done, you need to close the browser in the Sagemath appliance as well as Windows, and sent the shutdown signal before closing Virtualbox.

3. When you choose Print from within the Sagemath appliance, the default is Save as PDF. But because the Web browser in the Sage appliance is Google Chrome, Google Cloud Print is also an option. You can connect your printer to Google Cloud Print provided that Google Chrome is installed under Windows, and you have a Google (gmail) account. Using Chrome, go to http://www.google.com/cloudprint/learn and locate the instructions to set up your printer. If the printer is physically connected to the computer (not wireless), it's called a "classic" printer. Once your printer is connected, you can print to it from the Sage appliance through Google's servers, provided you are connected to the Internet and signed in to your Google account under Windows at the time. There is no need to install any fonts; they are already installed on the virtual linux machine.

4. Finally, in principle one should be able to install the appropriate printer driver (if one exists) in the virtual linux machine and print directly from the Sage appliance. Under Windows, you can access the linux command line using the free open source PuTTy SSH client, which can be obtained from www.putty.org. Once the Sagemath appliance is running, connect using Host Name localhost through port 2222. The user name is sage and the password is also sage. There may be better ways to reach the linux shell, but this works. You can ignore all the warnings.

   A package containing USB support for VirtualBox is available at https://www.virtualbox.org. Once it's installed, you can start looking for a linux driver for your printer. This printing method is appropriate only for those with linux experience who feel like playing around.

# Appendix C

# Data Sets

This appendix gives links to the data sets used in the text and homework problems, along with a listing of the first few lines. A zip archive of the data sets is also included with the full distribution of this text. All data sets are provided under the GNU Free Documentation license.

- **Baby Double**: Simulated $W_1, W_2, Y$ for an easy double measurement regression example. See Section 0.10.2 starting on Page 66.
  http://www.utstat.toronto.edu/~brunner/data/legal/Babydouble.data.txt

```
     W1    W2     Y
1  9.94 12.24 15.23
2 12.42 11.32 14.55
3 10.43 10.40 12.40
4  9.07  9.85 17.09
5 11.04 11.98 16.83
6 10.40 10.85 15.04
```

- **BMI Health**: Age, BMI, percent body fat, cholesterol level, and diastolic blood pressure were measured twice. See Page 89 for details.
  http://www.utstat.toronto.edu/~brunner/data/legal/bmi.data.txt

```
  age1 bmi1 fat1 cholest1 diastol1 age2 bmi2 fat2 cholest2 diastol2
1   63 24.5 16.5    195.4       38   60 23.9 20.1    203.5       66
2   42 13.0  1.9    184.3       86   44 14.8  2.6    197.3       78
3   32 22.5 14.6    354.1      104   33 21.7 20.4    374.3       73
4   59 25.5 19.0    214.6       93   58 28.5 20.0    203.7      106
5   45 26.5 17.8    324.8       97   43 25.0 12.3    329.7       92
6   31 19.4 17.1    280.7       92   42 19.9 19.9    276.7       87
```

- :
  http://www.utstat.toronto.edu/~brunner/data/legal/

- :
  http://www.utstat.toronto.edu/~brunner/data/legal/
- :
  http://www.utstat.toronto.edu/~brunner/data/legal/
- :
- :
- :
- :

# Appendix D

# Rules for Parameter Identifiability

**Note:** The rules listed here assume that errors are independent of exogenous variables that are not errors, and that all variables have been centered to have expected value zero. The following definition (Definition 3.1) is used frequently.

*An **indicator** for a latent variable is an observable variable that is a function only of that latent variable and an error term. The factor loading is non-zero.*

1. *Parameter Count Rule* (p. 61): If a model has more parameters than covariance structure equations, the parameter vector can be identifiable on at most a set of volume zero in the parameter space. This applies to all models.

2. *Measurement model* (Factor analysis) In these rules, latent variables that are not error terms are described as "factors."

   (a) *Double Measurement Rule* (p. 1.6): Parameters of the double measurement model are identifiable. All factor loadings equal one. Correlated measurement errors are allowed within sets of measurements, but not between sets.

   (b) *Three-variable Rule* (p. 295) The parameters of a factor analysis model are identifiable provided

   - There are at least three indicators for each factor.
   - For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading for at least one indicator is equal to one.
   - Errors are independent of one another.

   (c) *Reference Variable Rule* (p. 304) The parameters of a factor analysis model are identifiable except possibly on a set of volume zero in the parameter space, provided

   - The number of observable variables (including indicators) is at least three times the number of factors.
   - There is at least one indicator for each factor.

- For each factor, either the variance equals one and the sign of the indicator's factor loading is known, or the factor loading of the indicator is equal to one.
- Divide the observable variables into sets. The first set contains one indicator for each factor. The number of variables in the second set and the number in the third set is also equal to the number of factors. The fourth set may contain any number of additional variables, including zero. The error terms for the variables in the first three sets may have non-zero covariance within sets, but not between sets. The error terms for the variables in the fourth set may have non-zero covariance within the set, and with the error terms of sets two and three, but they must have zero covariance with the error terms of the indicators.

(d) *Two-variable Rule* (p. 313) The parameters of a factor analysis model are identifiable provided

- There are at least two factors.
- There are at least two indicators for each factor.
- For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading of at least one indicator is equal to one.
- Each factor has a non-zero covariance with at least one other factor.
- Errors are independent of one another.

(e) *Two-variable Addition Rule* (p. 313) A factor with just two indicators may be added to a measurement model whose parameters are identifiable, and the parameters of the combined model will be identifiable provided

- The errors for the two additional indicators are independent of one another and of the error terms already in the model.
- For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading of at least one indicator is equal to one.
- In the existing model with identifiable parameters,
  - There is at least one indicator for each factor, and
  - At least one factor has a non-zero covariance with the new factor.

(f) *Combination Rule* (p. 315) Suppose that two factor analysis models are based on non-overlapping sets of observable variables from the same data set, and that the parameters of both models are identifiable. The two models may be combined into a single model provided that the error terms of the first model are independent of the error terms in the second model. The additional parameters of the combined model are the covariances between the two sets of factors. These are all identifiable, except possibly on a set of volume zero in the parameter space.

(g) *Extra Variables Rule* (p. 317) If the parameters of a factor analysis model are identifiable, then a set of additional observable variables (without any new factors) may be added to the model. In the path diagram, straight arrows with factor loadings on them may point from each existing factor to each new variable. Error terms for the new variables may have non-zero covariances with each other. If the error terms of the new set have zero covariance with the error terms of the initial set and with the factors, then the parameters of the combined model are identifiable, except possibly on a set of volume zero in the parameter space.

(h) *Error-Free Rule* (p. 318) A set of observable variables may be added to the factors of a measurement model whose parameters are identifiable, provided that the new observed variables are independent of the error terms that are already in the model. The parameters of the resulting model are identifiable, except possibly on a set of volume zero in the parameter space.

(i) *Equivalence Rule* (p. 299) For a centered factor analysis model with at least one indicator for each factor, suppose that surrogate models are obtained by either standardizing the factors, or by setting the factor loading of an indicator to one for each factor. Then the parameters of one surrogate model are identifiable if and only if the parameters of the other surrogate model are identifiable.

3. *Latent variable model*: $\mathbf{y}_i = \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i$ Here, identifiability means that the parameters involved are functions of $cov(\mathbf{F}_i) = \boldsymbol{\Phi}$.

   (a) *Regression Rule:* (p. 376) If no endogenous variables are influenced by other endogenous variables in the latent variable model, the result is a regression model. The parameters of a regression model are identifiable.

   (b) *Acyclic Rule*: (p. 377) The parameters of the latent variable model are identifiable if the model is acyclic (no feedback loops through straight arrows) and the following conditions hold.

   - Organize the variables that are not error terms into sets. Set 0 consists of all the exogenous variables. They may have non-zero covariances.
   - For $j = 1, \ldots, m$, each endogenous variable in set $j$ may be influenced by all the variables in sets $\ell < j$.
   - Error terms for the endogenous variables in a set may have non-zero covariances. All other covariances between error terms are zero.

4. *Two-Step Rule*: This applies to models with both a measurement component and a latent variable component, including the full two-stage structural equation model.

   1: Consider the latent variable model as a model for observed variables. Check identifiability (usually using the Regression Rule and the Acyclic Rule).

   2: Consider the measurement model as a factor analysis model, ignoring the structure of $cov(\mathbf{F}_i)$. Check identifiability.

If both identification checks are successful, the parameters of the combined model are identifiable.

# Appendix E

# GNU Free Documentation License

Version 1.3, 3 November 2008


http://fsf.org



### Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers

to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the

beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque

copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.