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

$$\begin{aligned} \mathbf{y}_i &= \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \qquad (1.1) \\ \mathbf{F}_i &= \left(\frac{\mathbf{x}_i}{\mathbf{y}_i} \right) \\ \mathbf{d}_i &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \end{aligned}$$

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.

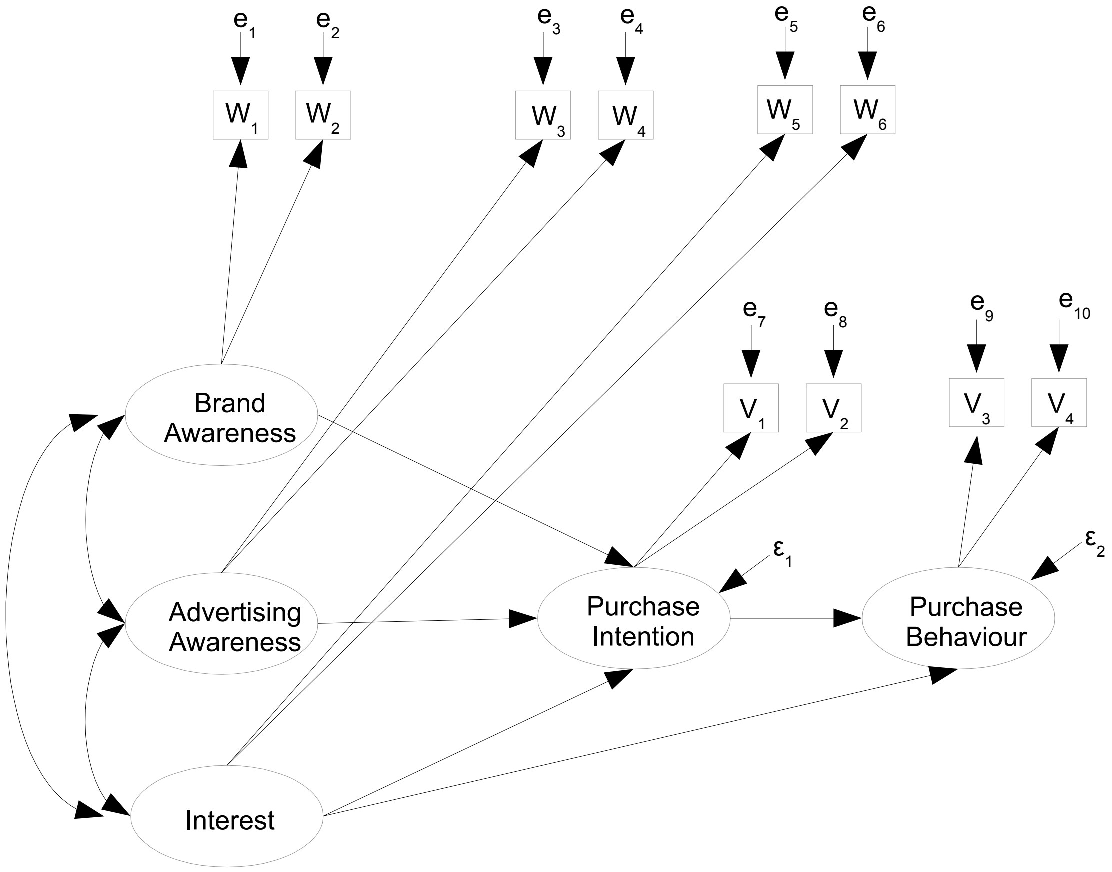
Figure 1.1 shows a path diagram for these data. It is a picture of how some variables are thought to influence other variables. The notation is standard. Straight arrows go from exogenous variables to endogenous variables, and possibly from endogenous variables to other endogenous variables. Correlations 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

$$\begin{array}{ccccccccccc}
\mathbf{y}_i & = & \boldsymbol{\alpha} & + & \boldsymbol{\beta} & \mathbf{y}_i & + & \boldsymbol{\Gamma} & \mathbf{x}_i & + & \boldsymbol{\epsilon}_i \\
\left(\begin{array}{c} Y_{i,1} \\ Y_{i,2} \end{array} \right) & = & \left(\begin{array}{c} \alpha_1 \\ \alpha_2 \end{array} \right) & + & \left(\begin{array}{cc} 0 & 0 \\ \beta & 0 \end{array} \right) & \left(\begin{array}{c} Y_{i,1} \\ Y_{i,2} \end{array} \right) & + & \left(\begin{array}{ccc} \gamma_1 & \gamma_2 & \gamma_3 \\ 0 & 0 & \gamma_4 \end{array} \right) & \left(\begin{array}{c} X_{i,1} \\ X_{i,2} \\ X_{i,3} \end{array} \right) & + & \left(\begin{array}{c} \epsilon_{i,1} \\ \epsilon_{i,2} \end{array} \right)
\end{array}$$

with

$$\boldsymbol{\Phi}_x = cov(\mathbf{x}_i) = \left(\begin{array}{ccc} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{12} & \phi_{22} & \phi_{23} \\ \phi_{13} & \phi_{23} & \phi_{33} \end{array} \right) \text{ and } \boldsymbol{\Psi} = cov(\boldsymbol{\epsilon}_i) = \left(\begin{array}{cc} \psi_1 & 0 \\ 0 & \psi_2 \end{array} \right).$$

Collecting $\mathbf{x}_i$ and $\mathbf{y}_i$ into a single vector of "factors,"

$$\mathbf{F}_i = \left(\begin{array}{c} \mathbf{x}_i \\ \hline \mathbf{y}_i \end{array} \right) = \left(\begin{array}{c} X_{i,1} \\ X_{i,2} \\ X_{i,3} \\ Y_{i,1} \\ Y_{i,2} \end{array} \right).$$

Finally, the equations of the measurement model are

$$\begin{array}{ccccccccc}
\mathbf{d}_i & = & \boldsymbol{\nu} & + & \boldsymbol{\Lambda} & & \mathbf{F}_i & + & \mathbf{e}_i \\
\left(\begin{array}{c} W_{i,1} \\ W_{i,2} \\ W_{i,3} \\ W_{i,4} \\ W_{i,5} \\ W_{i,6} \\ V_{i,1} \\ V_{i,2} \\ V_{i,3} \\ V_{i,4} \end{array} \right) & = &
\left(\begin{array}{c} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \\ \nu_5 \\ \nu_6 \\ \nu_7 \\ \nu_8 \\ \nu_9 \\ \nu_{10} \end{array} \right) & + &
\left(\begin{array}{ccccc} \lambda_1 & 0 & 0 & 0 & 0 \\ \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & \lambda_3 & 0 & 0 & 0 \\ 0 & \lambda_4 & 0 & 0 & 0 \\ 0 & 0 & \lambda_5 & 0 & 0 \\ 0 & 0 & \lambda_6 & 0 & 0 \\ 0 & 0 & 0 & \lambda_7 & 0 \\ 0 & 0 & 0 & \lambda_8 & 0 \\ 0 & 0 & 0 & 0 & \lambda_9 \\ 0 & 0 & 0 & 0 & \lambda_{10} \end{array} \right) & &
\left(\begin{array}{c} X_{i,1} \\ X_{i,2} \\ X_{i,3} \\ Y_{i,1} \\ Y_{i,2} \end{array} \right) & + &
\left(\begin{array}{c} e_{i,1} \\ e_{i,2} \\ e_{i,3} \\ e_{i,4} \\ e_{i,5} \\ e_{i,6} \\ e_{i,7} \\ e_{i,8} \\ e_{i,9} \\ e_{i,10} \end{array} \right)
\end{array}$$

with

$$\boldsymbol{\Omega} = cov(\mathbf{e}_i) = \left(\begin{array}{cccccccccc}
\omega_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \omega_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \omega_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \omega_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \omega_5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \omega_6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \omega_7 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_8 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_9 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_{10}
\end{array} \right)$$

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 Θ 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 Θ 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 Θ 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

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

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.

$$\begin{aligned}
& \mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\
\Leftrightarrow \;& \mathbf{y}_i - \boldsymbol{\beta}\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\
\Leftrightarrow \;& \mathbf{I}\mathbf{y}_i - \boldsymbol{\beta}\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\
\Leftrightarrow \;& (\mathbf{I} - \boldsymbol{\beta})\mathbf{y}_i = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\
\Leftrightarrow \;& (\mathbf{I} - \boldsymbol{\beta})^{-1}(\mathbf{I} - \boldsymbol{\beta})\mathbf{y}_i = (\mathbf{I} - \boldsymbol{\beta})^{-1}(\boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}) \\
\Leftrightarrow \;& \mathbf{y}_i = (\mathbf{I} - \boldsymbol{\beta})^{-1}(\boldsymbol{\alpha} + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i) \qquad (1.3)
\end{aligned}$$

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,

$$
\begin{aligned}
\boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F &= E(\mathbf{F}_i) = \left(\begin{array}{c} E(\mathbf{x}_i) \\ \hline E(\mathbf{y}_i) \end{array} \right) = \left(\begin{array}{c} \boldsymbol{\mu}_x \\ \hline (\mathbf{I}-\boldsymbol{\beta})^{-1}(\boldsymbol{\alpha} + \boldsymbol{\Gamma}\boldsymbol{\mu}_x) \end{array} \right) \qquad (1.4) \\
\boldsymbol{\mu} &= E(\mathbf{d}_i) = \boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F \\
\boldsymbol{\Phi} &= cov(\mathbf{F}_i) = \left(\begin{array}{c|c} cov(\mathbf{x}_i) & cov(\mathbf{x}_i, \mathbf{y}_i) \\ \hline & cov(\mathbf{y}_i) \end{array} \right) = \left(\begin{array}{c|c} \boldsymbol{\Phi}_x & \boldsymbol{\Phi}_x \boldsymbol{\Gamma}^\top (\mathbf{I}-\boldsymbol{\beta})^{-1T} \\ \hline & (\mathbf{I}-\boldsymbol{\beta})^{-1} \left(\boldsymbol{\Gamma}\boldsymbol{\Phi}_x\boldsymbol{\Gamma}^\top + \boldsymbol{\Psi} \right) (\mathbf{I}-\boldsymbol{\beta})^{-1T} \end{array} \right) \\
\boldsymbol{\Sigma} &= cov(\mathbf{d}_i) = \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^\top + \boldsymbol{\Omega}
\end{aligned}
$$

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

$$\overset{c}{\mathbf{F}}_i = \mathbf{F}_i - \boldsymbol{\mu}_F = \left(\begin{array}{c} \overset{c}{\mathbf{x}}_i \\ \hline \overset{c}{\mathbf{y}}_i \end{array} \right).$$

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

$$\begin{array}{rcl} \mathbf{y}_i & = & \boldsymbol{\beta}\mathbf{y}_i + \boldsymbol{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i \\ \mathbf{F}_i & = & \left(\begin{array}{c} \mathbf{x}_i \\ \hline \mathbf{y}_i \end{array} \right) \\ \mathbf{d}_i & = & \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \end{array} \tag{1.5}$$

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

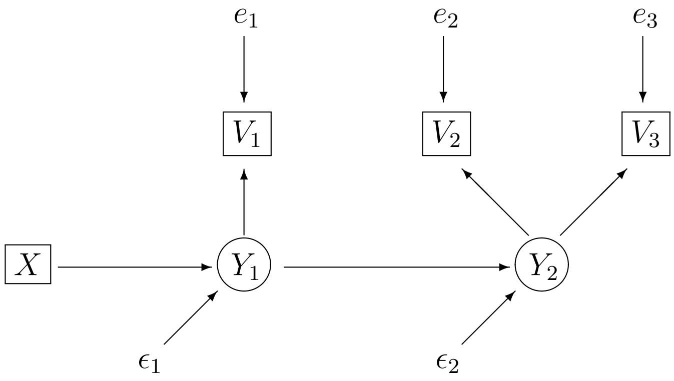
In general, the parameters of the centered surrogate model are still not identifiable. In most cases, even after restricting the parameters based on modeling considerations, further technical restrictions are necessary to obtain a model whose parameters are identifiable. Like centering, these restrictions 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. Independently 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 λ_1 and λ_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

$$\left(\begin{array}{cccc}
\phi & \gamma\lambda_1\phi & \beta\gamma\lambda_2\phi & \beta\gamma\lambda_2\phi \\
 & (\gamma^2\phi + \psi_1)\lambda_1^2 + \omega_1 & (\gamma^2\phi + \psi_1)\beta\lambda_1\lambda_2 & (\gamma^2\phi + \psi_1)\beta\lambda_1\lambda_2 \\
 & & (\beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2)\lambda_2^2 + \omega_2 & (\beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2)\lambda_2^2 \\
 & & & (\beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2)\lambda_2^2 + \omega_2
\end{array} \right). \tag{1.7}$$

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 σ_{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 σ_{13} and σ_{14} both represent the same real number. At this point, parameter symbols like β and γ 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 β and γ are treated as variables, while the σ_{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 γ, β, λ_1 and λ_2 all non-zero) that yield the same covariance matrix.

	γ	β	λ_1	λ_2	ψ_1	ψ_2	ϕ	ω_1	ω_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

$$\boldsymbol{\Sigma} = \left(\begin{array}{rrrr} 1 & 2 & 8 & 8 \\ 2 & 9 & 32 & 32 \\ 8 & 32 & 145 & 144 \\ 8 & 32 & 144 & 145 \end{array} \right).$$

By Definition 0.5, the parameter vector is not identifiable.

The next step is to re-examine the model equations in (surrogate) centered form,

$$\begin{aligned} Y_{i,1} &= \gamma X_i + \epsilon_{i,1} \\ Y_{i,2} &= \beta Y_{i,1} + \epsilon_{i,2} \\ V_{i,1} &= \lambda_1 Y_{i,1} + e_{i,1} \\ V_{i,2} &= \lambda_2 Y_{i,2} + e_{i,2} \\ V_{i,3} &= \lambda_2 Y_{i,2} + e_{i,3} \end{aligned} \qquad (1.8)$$

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 ω_1 and ω_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 (λ_1 and λ_2, in this case) that have been made invisible by the re-parameterization. In terms of the new parameters, the covariance matrix $\boldsymbol{\Sigma}$ is

$$\left(\begin{array}{rrrr} \phi & \gamma\phi & \beta\gamma\phi & \beta\gamma\phi \\ \gamma\phi & \gamma^2\phi + \omega_1 + \psi_1 & (\gamma^2\phi + \psi_1)\beta & (\gamma^2\phi + \psi_1)\beta \\ \beta\gamma\phi & (\gamma^2\phi + \psi_1)\beta & \beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 & \beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2 \\ \beta\gamma\phi & (\gamma^2\phi + \psi_1)\beta & \beta^2\gamma^2\phi + \beta^2\psi_1 + \psi_2 & \beta^2\gamma^2\phi + \beta^2\psi_1 + \omega_2 + \psi_2 \end{array} \right). \tag{1.11}$$

It is easy to solve for the new parameters in terms of the variances and covariances σ_{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 γ and β 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 λ_1 and λ_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 ϕ_{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, γ and β might appear unaltered as parameters in the new model. But the numerical example shows that γ and β 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

$$\begin{array}{ccccc} \mathbf{d}_i & = & \boldsymbol{\Lambda} & \mathbf{F}_i & + & \mathbf{e}_i \\ \begin{pmatrix} X_i \\ V_{i,1} \\ V_{i,2} \\ V_{i,3} \end{pmatrix} & = & \begin{pmatrix} 1 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \\ 0 & 0 & \lambda_2 \end{pmatrix} & \begin{pmatrix} X_i \\ Y_{i,1} \\ Y_{i,2} \\ X_{i,3} \end{pmatrix} & + & \begin{pmatrix} e_{i,1} \\ e_{i,2} \\ e_{i,3} \\ e_{i,4} \end{pmatrix}. \end{array}$$

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 1×1.

```
# 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 2×1.

```
GAMMA = ZeroMatrix(2,1); GAMMA[0,0] = var('gamma'); show(GAMMA)
```

evaluate

$$\begin{pmatrix} \gamma \\ 0 \end{pmatrix}$$

The matrix $\boldsymbol{\beta}$ is 2×2.

```
BETA = ZeroMatrix(2,2); BETA[1,0] = var('beta'); show(BETA)
```

evaluate

$$\begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix}$$

The 2×2 matrix $\mathbf{\Psi}$ can be created directly with the `DiagonalMatrix` function; the default symbol is a ψ.

```
PSI = DiagonalMatrix(2); show(PSI)
```

evaluate

$$\begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix}$$

The matrix $\mathbf{\Lambda}$ is 4×3.

```
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 $\boldsymbol{\Sigma}$ 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 $\boldsymbol{\Sigma}$ 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 $\boldsymbol{\Phi}$ or $\boldsymbol{\Sigma}$. For example, the 4×3 matrix $\boldsymbol{\Lambda}$ contains just two parameters, λ_1 and λ_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 `extend`ed by adding the contents of $\boldsymbol{\Psi}$ and $\boldsymbol{\Omega}$. 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, σ_{13} and σ_{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 σ_{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 $\boldsymbol{\Sigma}$. 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 σ_{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 ψ_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

$$\left(\phi, \sigma_{11}\right)$$
$$\left(\beta, \frac{\sigma_{13}}{\sigma_{12}}\right)$$
$$\left(\gamma, \frac{\sigma_{12}}{\sigma_{11}}\right)$$
$$\left(\psi_{1}, \frac{\sigma_{11}\sigma_{12}\sigma_{23} - \sigma_{12}^{2}\sigma_{13}}{\sigma_{11}\sigma_{13}}\right)$$
$$\left(\psi_{2}, \frac{\sigma_{12}\sigma_{34} - \sigma_{13}\sigma_{23}}{\sigma_{12}}\right)$$
$$\left(\omega_{1}, -\frac{\sigma_{12}\sigma_{23} - \sigma_{13}\sigma_{22}}{\sigma_{13}}\right)$$
$$\left(\omega_{2}, \sigma_{33} - \sigma_{34}\right)$$

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 σ_{ij} values. It can also be useful to have a dictionary that goes in the other direction, where the input is in terms σ_{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 $\boldsymbol{\Sigma}$, 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 ψ_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, β, γ and λ_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.

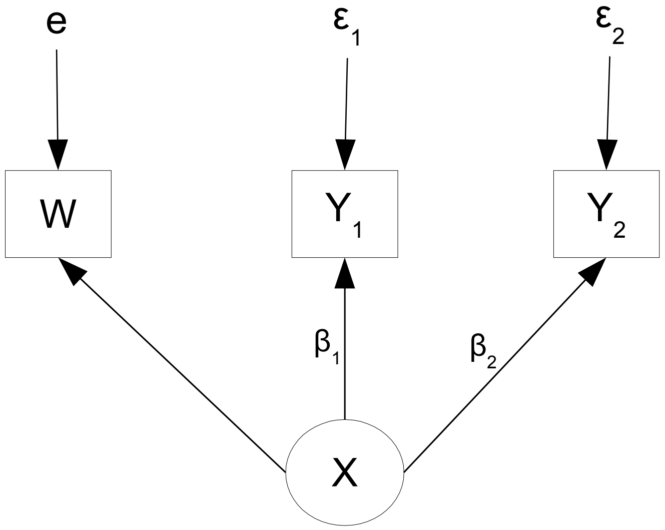
$$
\begin{aligned}
W_i &= X_i + e_i \\
Y_{i,1} &= \beta_1 X_i + \epsilon_{i,1} \\
Y_{i,2} &= \beta_2 X_i + \epsilon_{i,2}
\end{aligned}
\tag{1.12}
$$

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 β_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.

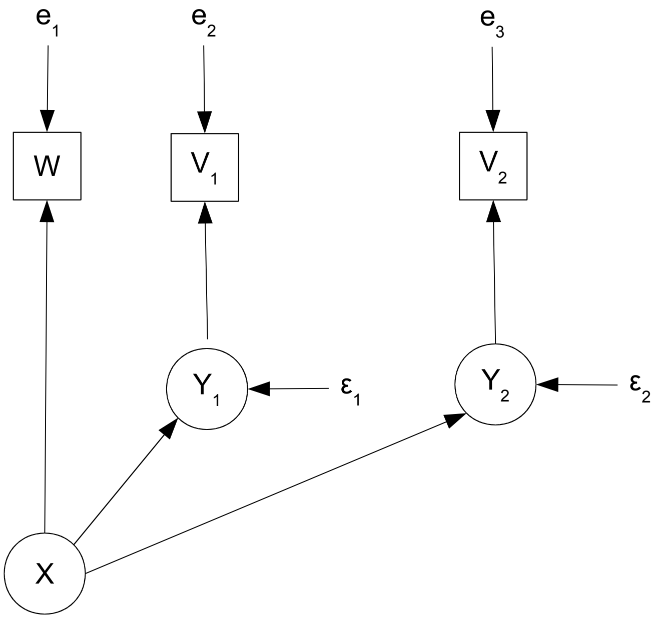
Figure 1.5 depicts a somewhat more reasonable model for the smoking example, and it is proposed as the original model. In this model, $Y_{i,1}$ is true body mass index, while $V_{i,1}$ is the measured version. $Y_{i,2}$ is true average resting heart rate, while $V_{i,2}$ is the snapshot measured with 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 λ_1, λ_2 and λ_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 β_1 and β_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,

$$\begin{array}{rcl} Y_{i,j} & = & \mathbf{r}_j^\top \mathbf{x}_i + \epsilon_{i,j} \\ V_{i,j} & = & \lambda_j Y_{i,j} + e_{i,j}, \end{array}$$

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 ψ_j and ω_j, all in one step.

$$\begin{array}{rcl} V_{i,j} & = & \lambda_j Y_{i,j} + e_{i,j} \\ & = & \lambda_j(\mathbf{r}_j^\top \mathbf{x}_i + \epsilon_{i,j}) + e_{i,j} \\ & = & (\lambda_j \mathbf{r}_j)^\top \mathbf{x}_i + (\lambda_j \epsilon_{i,j} + e_{i,j}) \\ & = & \mathbf{r}_j'^\top \mathbf{x}_i + \epsilon_{i,j}', \end{array}$$

with $Var(\epsilon_{i,j}') = \psi_j' = \lambda_j^2 \psi_j + \omega_j$. The β and γ 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.

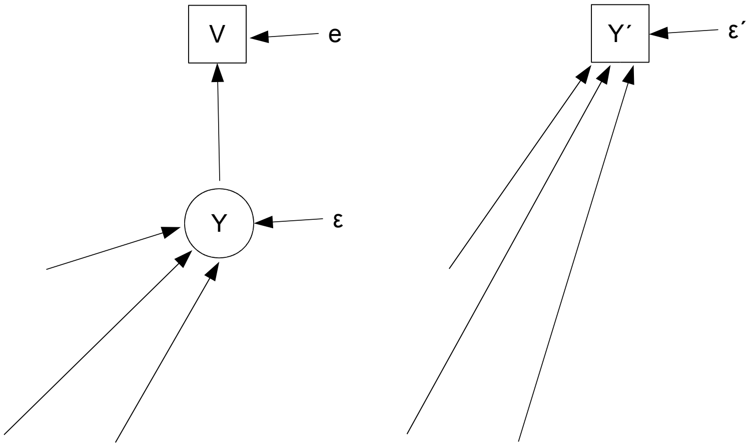
- The meaning of the new parameters is clear. The β and γ parameters in $\mathbf{r}_j$ are positive multiples of what they were before, while any *separate* meaning that ψ_j and ω_j may have had is lost. They were probably not knowable anyway.

- 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

$$\boldsymbol{\mu}_F = \left(\begin{array}{c} \boldsymbol{\mu}_x \\ \hline (\mathbf{I} - \boldsymbol{\beta})^{-1} (\boldsymbol{\alpha} + \boldsymbol{\Gamma}\boldsymbol{\mu}_x) \end{array} \right).$$

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 $n \rightarrow \infty$. 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 θ_j,

$$z = \frac{\widehat{\theta}_j - \theta_j}{s_{\theta_j}} \qquad (1.15)$$

has an approximate standard normal distribution for large samples, where s_{θ_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

$$\begin{aligned} G^2 &= -2\log\left(\frac{L(\widehat{\boldsymbol{\theta}}_0)}{L(\widehat{\boldsymbol{\theta}})}\right) \qquad (1.17)\\ &= 2\left(\ell(\widehat{\boldsymbol{\theta}}) - \ell(\widehat{\boldsymbol{\theta}}_0)\right), \end{aligned}$$

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 σ_{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

$$\begin{aligned}
G^2 &= -2\log\frac{L\Big(\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})\Big)}{L(\widehat{\boldsymbol{\Sigma}})} \\
&= -2\log\frac{|\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})|^{-n/2}(2\pi)^{-nk/2}\exp -\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1})\right\}}{|\widehat{\boldsymbol{\Sigma}}|^{-n/2}(2\pi)^{-nk/2}\exp -\frac{n}{2}\left\{tr(\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Sigma}}^{-1})\right\}} \\
&= n\left(\log|\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})| + tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}| - k\right) \\
&= n\left(tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}| - k\right) \qquad (1.18)
\end{aligned}$$

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"

$$b(\boldsymbol{\theta}) = tr(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}) - k - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}| \qquad (1.19)$$

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 σ_{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 ω 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 4×4 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.

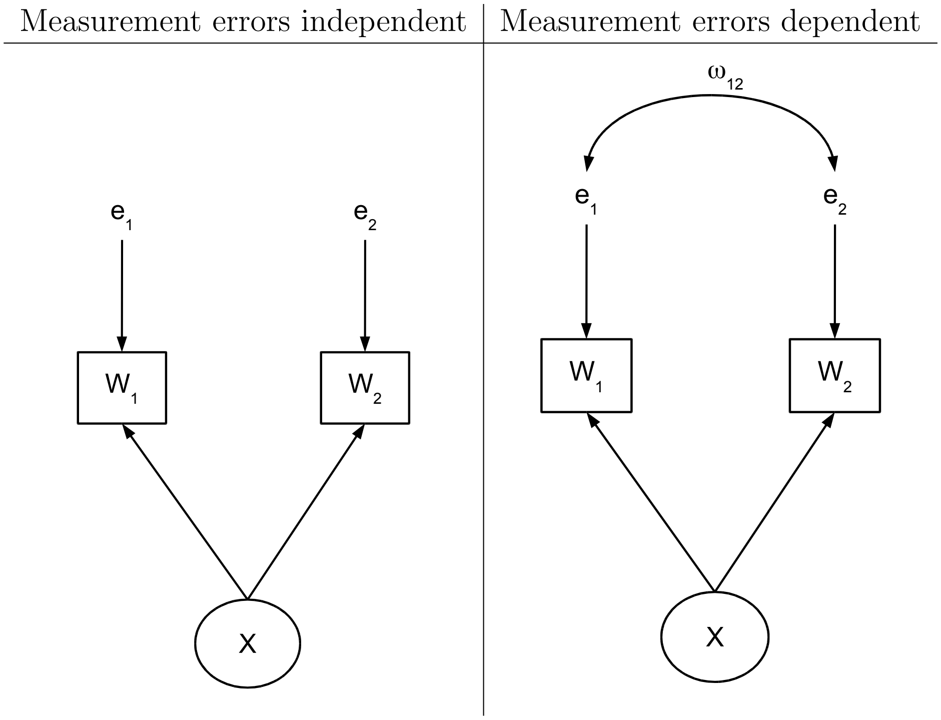
Example 1.5.1 *A negative variance estimate*

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 observable variables $(W_{i,1}, W_{i,2})^\top$ is

Figure 1.7: Two measurements of a latent variable



Measurement errors independent | Measurement errors dependent

$$\left(\begin{array}{cc} \omega_1 + \phi & \phi \\ \phi & \omega_2 + \phi \end{array} \right) = \left(\begin{array}{cc} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{array} \right).$$

The model is saturated, with three linear covariance structure equations in three unknown parameters. The solutions are

$$\begin{aligned} \phi &= \sigma_{12} \\ \omega_1 &= \sigma_{11} - \sigma_{12} \\ \omega_2 &= \sigma_{22} - \sigma_{12}, \end{aligned} \qquad (1.20)$$

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 ω_{12}. The resulting path diagram is shown in the right panel of Figure 1.7. The covariance matrix of the observable variables is now

$$\left(\begin{array}{cc} \omega_1 + \phi & \phi + \omega_{12} \\ \phi + \omega_{12} & \omega_2 + \phi \end{array}\right) = \left(\begin{array}{cc} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{array}\right).$$

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 ω_1 is $\widehat{\omega}_1 = \widehat{\sigma}_{11} - \widehat{\sigma}_{12}$. But under the correct model,

$$\begin{array}{rcl} \widehat{\omega}_1 & = & \widehat{\sigma}_{11} - \widehat{\sigma}_{12} \\ & \overset{a.s.}{\rightarrow} & \sigma_{11} - \sigma_{12} \\ & = & (\omega_1 + \phi) - (\phi + \omega_{12}) \\ & = & \omega_1 - \omega_{12}. \end{array}$$

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.

$$\left(\begin{array}{cc} \omega_1 & \omega_{12} \\ \omega_{12} & \omega_2 \end{array}\right) = \left(\begin{array}{cc} 1 & 2 \\ 2 & 5 \end{array}\right).$$

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 variables 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.

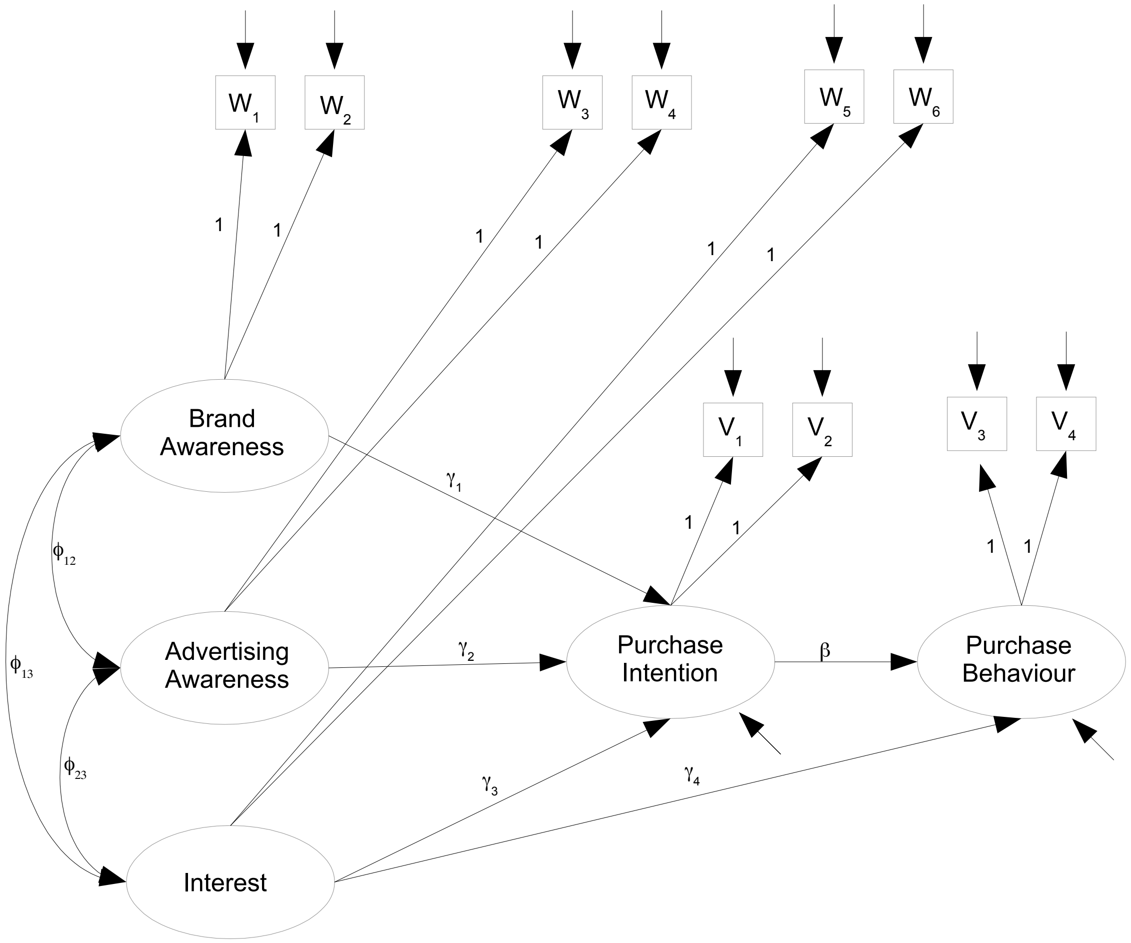
Figure 1.8 shows the path diagram for a surrogate model. It is more detailed than Figure 1.1 on page 140, in that symbols are indicted on the arrows. You can tell it's a surrogate model because of the symbol "1" on the arrows linking latent to observed variables. The model asserts that all measurement 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

$$\begin{aligned}
Y_{i,1} &= \gamma_1 X_{i,1} + \gamma_2 X_{i,2} + \gamma_3 X_{i,3} + \epsilon_{i,1} \qquad (1.21)\\
Y_{i,2} &= \beta Y_{i,1} + \gamma_4 X_{i,3} + \epsilon_{i,2}\\
W_{i,1} &= X_{i,1} + e_{i,1}\\
W_{i,2} &= X_{i,1} + e_{i,2}\\
W_{i,3} &= X_{i,2} + e_{i,3}\\
W_{i,4} &= X_{i,2} + e_{i,4}\\
W_{i,5} &= X_{i,3} + e_{i,5}\\
W_{i,6} &= X_{i,3} + e_{i,6}\\
V_{i,1} &= Y_{i,1} + e_{i,7}\\
V_{i,2} &= Y_{i,1} + e_{i,8}\\
V_{i,3} &= Y_{i,2} + e_{i,9}\\
V_{i,4} &= Y_{i,2} + e_{i,10},
\end{aligned}$$

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 $\mathbf{d}_i = \mathbf{\Lambda}\mathbf{F}_i + \mathbf{e}_i$, with $cov(\mathbf{F}_i) = \mathbf{\Phi}$ and $cov(\mathbf{e}_i) = \mathbf{\Omega}$.
- The latent variable model is $\mathbf{y}_i = \boldsymbol{\beta}\mathbf{y}_i + \mathbf{\Gamma}\mathbf{x}_i + \boldsymbol{\epsilon}_i$, with $cov(\mathbf{x}_i) = \mathbf{\Phi}_x$ and $cov(\boldsymbol{\epsilon}_i) = \mathbf{\Psi}$.
- The models are linked by $\mathbf{F}_i = \left(\frac{\mathbf{x}_i}{\mathbf{y}_i} \right)$.

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

$$\mathbf{d}_i = \left(\frac{\mathbf{d}_{i,1}}{\mathbf{d}_{i,2}} \right), \text{ where } \mathbf{d}_{i,1} = \begin{pmatrix} W_{i,1} \\ W_{i,3} \\ W_{i,5} \\ V_{i,1} \\ V_{i,3} \end{pmatrix} \text{ and } \mathbf{d}_{i,2} = \begin{pmatrix} W_{i,2} \\ W_{i,4} \\ W_{i,6} \\ V_{i,2} \\ V_{i,4} \end{pmatrix}.$$

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 ψ_1. The second line of (1.21) is also just a regression, and the parameters γ_4, β and ψ_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
```

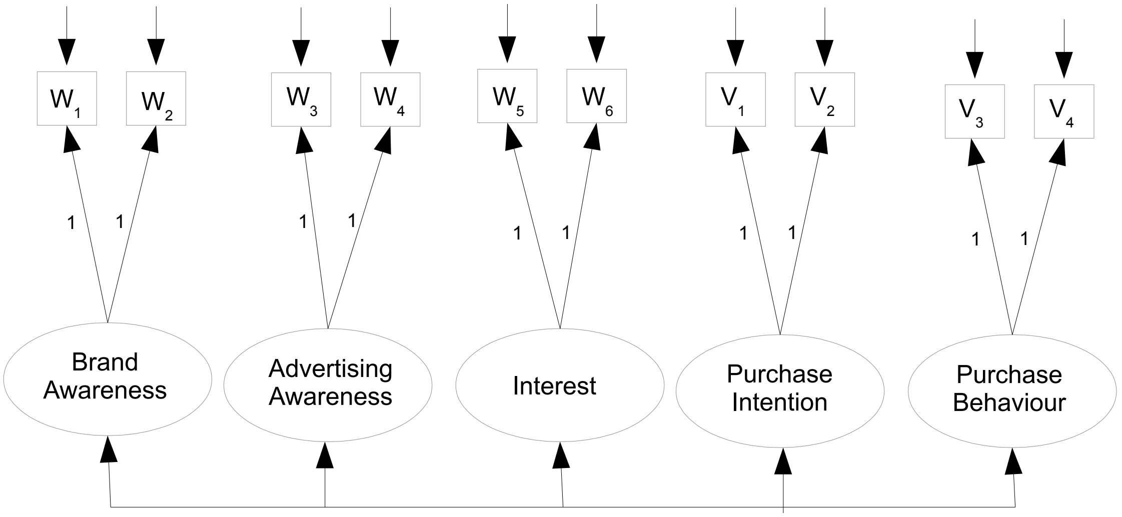
By the likelihood ratio test, the model does not fit[19]. A close look at the output of **summary** and **partable** reveals nothing out of the ordinary. We need determine why the model did not fit, and fix it if possible. To do this, a divide and conquer strategy can be helpful. We'll split the problem 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
```

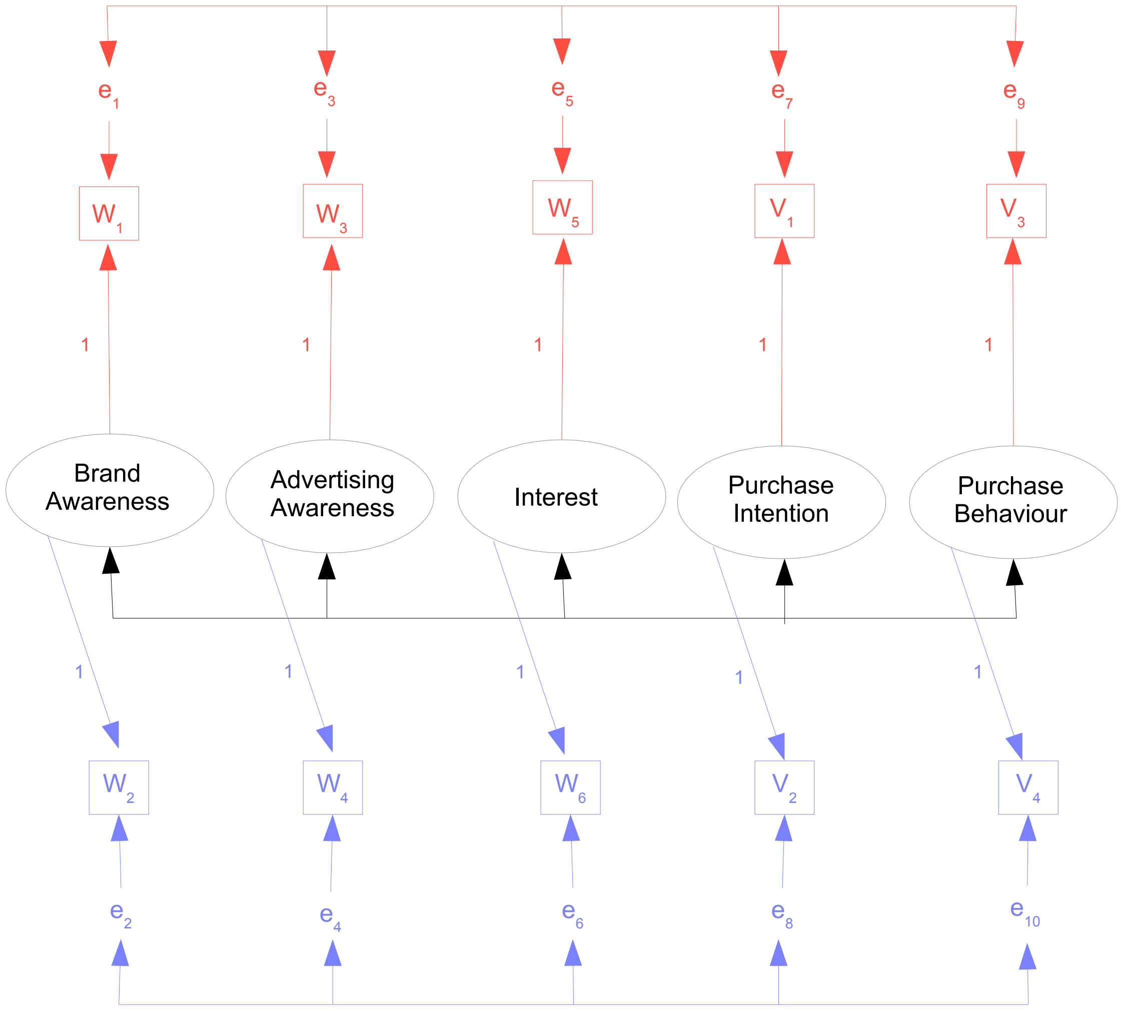
The measurement model does not fit[21], and we need to fix it. Now, the model asserts a kind of double measurement, but it's a restricted kind in which all the measurement errors are all independent. Maybe independence does not hold, and that's causing the lack of fit.

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 $\widehat{\boldsymbol{\Phi}}$ is okay. Computing and testing the estimated covariance matrices of the error terms,

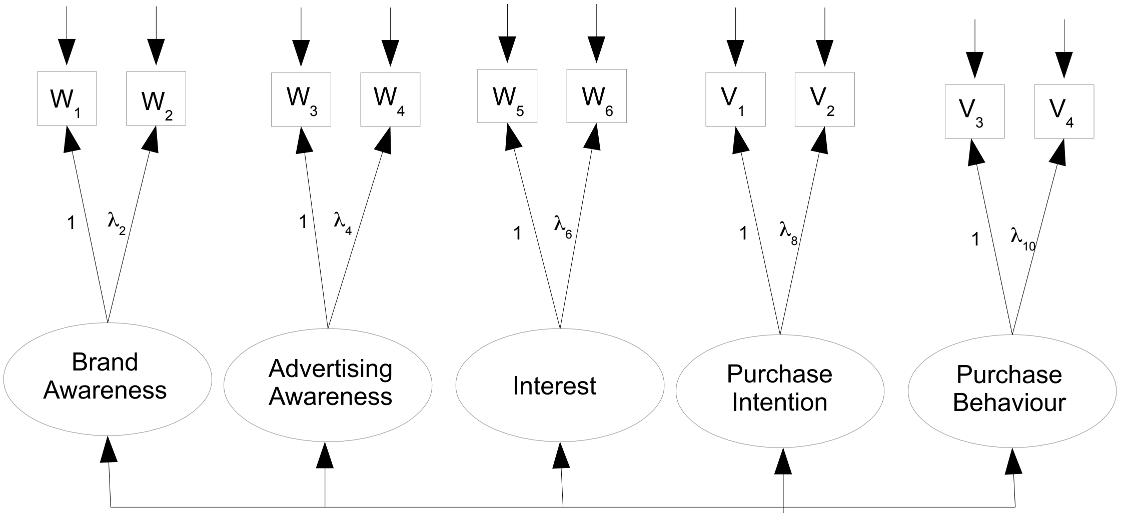
```
> Omega1_hat = cov(d1) - Phi_hat
> Omega2_hat = cov(d2) - Phi_hat
> eigen(Omega1_hat)$values # Is Omega1_hat positive definite?
[1] 26.402687  9.301147  8.288868  5.106178  2.868356
> eigen(Omega2_hat)$values # Is Omega2_hat positive definite?
[1] 12.867799 11.828405  9.847771  4.712254 -3.393667
```

The method-of-moments estimate $\widehat{\boldsymbol{\Omega}}_2$ 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,

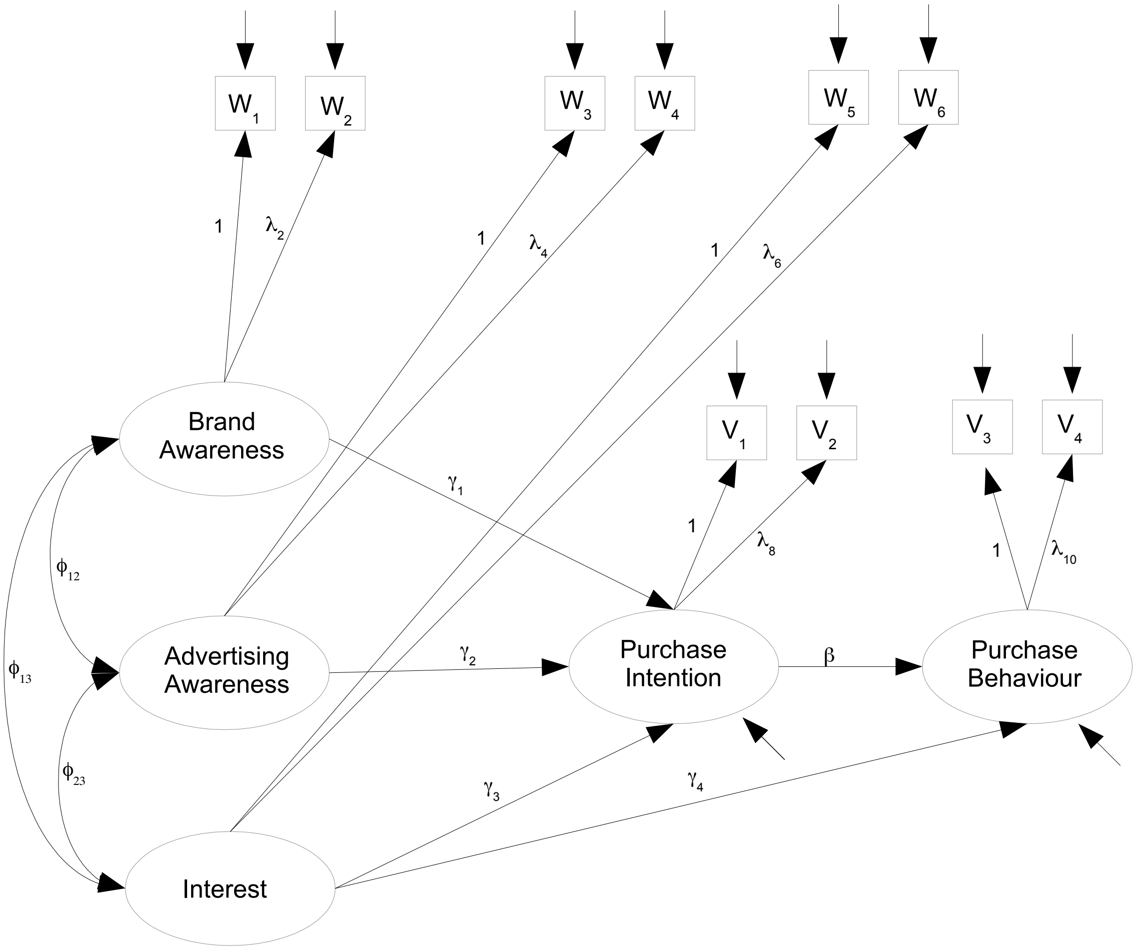
Figure 1.11: Brand Awareness Model Four



```
> 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 λ_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 γ_1, γ_2 and γ_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 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.

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
```

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(0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
+            c(0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
+            c(0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
+            c(0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
+            c(0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0))
> 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 ($\mathbf{\Phi}$ and $\mathbf{\Lambda}$) are first recovered from $\mathbf{\Sigma}$, the variance-covariance matrix of a vector of an observable data vector. Then, the parameters of the latent variable model ($\mathbf{\Phi}, \mathbf{\Gamma}, \boldsymbol{\beta}$ and $\mathbf{\Psi}$) are recovered from $\mathbf{\Phi}$. Since $\mathbf{\Phi}$ has already been shown to be a function of $\mathbf{\Sigma}$, this shows that all the parameters are a function of $\mathbf{\Sigma}$, 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
-