

Chapter 3

Confirmatory Factor Analysis

In confirmatory factor analysis, as in exploratory factor analysis, a set of unobservable latent variables called “factors” give rise to a set of observable variables. The principal difference between exploratory and confirmatory factor analysis is in the treatment of parameter identifiability. Exploratory factor analysis models include a link between every factor and every observable variable, and attempt to deal with the resulting lack of identifiability by rotating the factor solutions. Confirmatory factor analysis behaves much more like a traditional statistical method. Based on substantive considerations and re-parameterizations, the dimension of the parameter space is reduced so as to make the parameters identifiable. Then, estimation and inference proceed as usual. Confirmatory factor analysis models are directly imported as the measurement model in the general two-stage model of Chapter 1.

Given a set of data (or proposed set of data), it is generally quite easy to come up with a confirmatory factor analysis model. Such a model may be blessed with identifiability, or it may not. If not, it’s back to the drawing board. The primary objective of this chapter is to develop a set of rules that will allow the reader to determine the identifiability status of a model without elaborate calculation – usually by just examining the path diagram. As in Chapters 0 and 1, identifiable almost always means identifiable from the covariance matrix. The rules for parameter identifiability from throughout the book, including this chapter, are collected in Appendix D.

Using the conceptual framework of Chapter 1, underlying everything is a regression-like *original model*. The parameters of the original model will not be identifiable, so it is simplified and re-parameterized to obtain a *surrogate model* whose parameters may be identifiable. The parameters of the surrogate model bear a systematic relationship to the parameters of the original model, and by keeping track of what that relationship is, it will be possible to draw conclusions about the parameters of the original model. For example, suppose a parameter θ_j of the surrogate model is a positive multiple of a parameter in the original model. Then if a test determines that $\theta_j > 0$, it can also be concluded that the parameter of the original model is positive.

Here is the original model for confirmatory factor analysis. It is a part of the general two-stage model (1.1). Independently for $i = 1, \dots, n$, let

$$\mathbf{d}_i = \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \quad (3.1)$$

where

- \mathbf{d}_i is a $k \times 1$ observable random vector. The expected value of \mathbf{d}_i will be denoted by $\boldsymbol{\mu}$, and the covariance matrix of \mathbf{d}_i will be denoted by $\boldsymbol{\Sigma}$.
- $\boldsymbol{\nu}$ is a $k \times 1$ vector of constants.
- $\boldsymbol{\Lambda}$ is a $k \times (p + q)$ matrix of constants.
- \mathbf{F}_i (F for Factor) is a $p \times 1$ latent random vector whose expected value is denoted by $\boldsymbol{\mu}_F$, and whose variance-covariance matrix is denoted by $\boldsymbol{\Phi}$.
- \mathbf{e}_i is a $k \times 1$ vector of error terms that is independent of \mathbf{F}_i . It has expected value zero and covariance matrix $\boldsymbol{\Omega}$, which need not be positive definite.

This looks a lot like a multivariate regression model, and it is more or less acceptable for all the reasons that regression is acceptable. It may not be exactly correct, but there is hope that it's a reasonable approximation of the truth, at least within the range of the data.

As discussed in Section A.6.1 of Chapter 1, the parameter vectors $\boldsymbol{\nu}$ and $\boldsymbol{\mu}_F$ will almost never be identifiable separately based on $\boldsymbol{\mu}$, even if it were possible to identify $\boldsymbol{\Lambda}$, $\boldsymbol{\Phi}$ and $\boldsymbol{\Omega}$ from $\boldsymbol{\Sigma}$. Accordingly, we re-parameterize, obtaining a *surrogate centered model*. As a warm-up for what is to come, it is helpful to express the re-parameterization as a change of variables.

$$\begin{aligned} \mathbf{d}_i &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + \mathbf{e}_i \\ \iff \mathbf{d}_i &= \boldsymbol{\nu} + \boldsymbol{\Lambda}\mathbf{F}_i + (\boldsymbol{\Lambda}\boldsymbol{\mu}_F - \boldsymbol{\Lambda}\boldsymbol{\mu}_F) + \mathbf{e}_i \\ \iff \mathbf{d}_i - (\boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F) &= \boldsymbol{\Lambda}(\mathbf{F}_i - \boldsymbol{\mu}_F) + \mathbf{e}_i \\ \iff (\mathbf{d}_i - \boldsymbol{\mu}) &= \boldsymbol{\Lambda}(\mathbf{F}_i - \boldsymbol{\mu}_F) + \mathbf{e}_i \\ \iff \overset{c}{\mathbf{d}}_i &= \boldsymbol{\Lambda} \overset{c}{\mathbf{F}}_i + \mathbf{e}_i, \end{aligned} \quad (3.2)$$

where the superscript c indicates *centered* versions of the random vectors, in which \mathbf{d}_i and \mathbf{F}_i are expressed as deviations from their expected values. The centering notation is dropped, and the result is a model from which $\boldsymbol{\nu}$ and $\boldsymbol{\mu}_F$ have been eliminated. We are glad to see them go. They are not identifiable separately anyway, and the function of $\boldsymbol{\nu}$ and $\boldsymbol{\mu}_F$ that is identifiable, $\boldsymbol{\nu} + \boldsymbol{\Lambda}\boldsymbol{\mu}_F$, is of very little interest. The parameters we really care about are the factor loadings in $\boldsymbol{\Lambda}$ and the correlations between factors in $\boldsymbol{\Phi}$. These quantities are unaffected by centering.

Here is a full statement of the *centered surrogate model*. Independently for $i = 1, \dots, n$,

$$\mathbf{d}_i = \mathbf{\Lambda}\mathbf{F}_i + \mathbf{e}_i, \quad (3.3)$$

where

- All expected values are zero.
- \mathbf{d}_i is a $k \times 1$ observable random vector, with $cov(\mathbf{d}_i) = \mathbf{\Sigma}$.
- $\mathbf{\Lambda}$ is a $k \times (p + q)$ matrix of constants (factor loadings).
- \mathbf{F}_i (F for Factor) is a $p \times 1$ latent random vector with $cov(\mathbf{F}_i) = \mathbf{\Phi}$.
- \mathbf{e}_i is a $k \times 1$ random vector of error terms that is independent of \mathbf{F}_i . Its covariance matrix is $\mathbf{\Omega}$.

In practice, special cases of this model will be fit to data sets where the expected values of the variables are definitely not zero. There are two ways to justify this, equivalent in practice. The first solution is to leave \mathbf{d}_i uncentered in the model, and estimate the nuisance parameters in $\boldsymbol{\mu} = \boldsymbol{\nu} + \mathbf{\Lambda}\boldsymbol{\mu}_F$ with the vector of sample means $\bar{\mathbf{d}}$. The other solution is to center \mathbf{d}_i in the data set, by subtracting off $\bar{\mathbf{d}}$. In either case, inference about $\mathbf{\Lambda}$ and $\mathbf{\Phi}$ will be based on the sample covariance matrix $\hat{\mathbf{\Sigma}}$.

Readers of Chapter 2 will recognize Model (3.3) as almost identical to the “general factor analysis model” (2.6) on page 213. The only difference is that here, $cov(\mathbf{e}_i) = \mathbf{\Omega}$ need not be diagonal, though it is diagonal in many of the simpler models. One could say that, recognizing exploratory factor analysis as a failure, we are starting over.

It was shown in Chapter 2 (especially Sections 2.2 and 2.3) that the parameters of the centered surrogate are not identifiable without some further restrictions on the parameter space. These restrictions are of two kinds. The first kind of restriction is substantive, based on the nature of the data. Setting parameters equal to one another (for example, equal factor loadings) or equal to zero are invariably substantive restrictions, and must be justified in terms of the data set.

The other kind of restriction involves setting certain parameters to the value one. Thinking of the original Model (3.1) as the “true model,” this might seem like an arbitrary restriction of the parameter space. However, it will turn out that the resulting model is a surrogate model, in which the centered model (3.3) has been re-parameterized by a change of variables. The parameters of the surrogate model are identifiable *functions* of the original model parameters. By making the process of re-parameterization explicit, we will be able to tell what the surrogate model parameters mean.

Again, the primary objective of this chapter is to build up a set of simple rules for deciding whether the parameters of a proposed model are identifiable. Two important rules have already been established. They are the *Parameter Count Rule* (Rule 1, first stated on page 61) and the *Double Measurement Rule* (Rule 2a, page 178). The parameter count

rule gives a simple necessary condition for identifiability¹, while the double measurement rule, like most of the other standard rules in this book, describes a sufficient condition. The double measurement rule fits neatly into the next section.

3.1 Setting Some Factor Loadings to One

In both the original Model (3.1) and the centered surrogate model (3.3), the factor loadings in the matrix $\mathbf{\Lambda}$ are unrestricted. In this section, parameter identifiability will be obtained by setting some factor loadings to one. We will start by just accepting these models as given, focusing on the technical details of identifiability. Then later, it will be shown how these seemingly arbitrary restrictions of the parameter space are actually reparameterizations that result in surrogate model, one whose parameters have a systematic relationship to the parameters of the original model.

Double Measurement Recall the double measurement model (1.22) on page 177, which arose in the course of checking identifiability for the brand awareness data. Figure 3.1 shows a simple scalar example.

Each factor is measured by two observable variables; the factor loadings are all equal to one. There are two sets of measurements, with potentially non-zero covariances within sets, but not between sets. As in the brand awareness Example 1.2 on page 139, common extraneous influences on the measurements within each set are to be expected, but pains have been taken to make the two sets of measurements independent. In general there can be any number of factors, but it becomes challenging to draw the path diagram. Figure 1.10 on page 184 is a try with five factors.

To re-state the double measurement model in matrix form, let

$$\begin{aligned} \mathbf{d}_{i,1} &= \mathbf{F}_i + \mathbf{e}_{i,1} \\ \mathbf{d}_{i,2} &= \mathbf{F}_i + \mathbf{e}_{i,2}, \end{aligned}$$

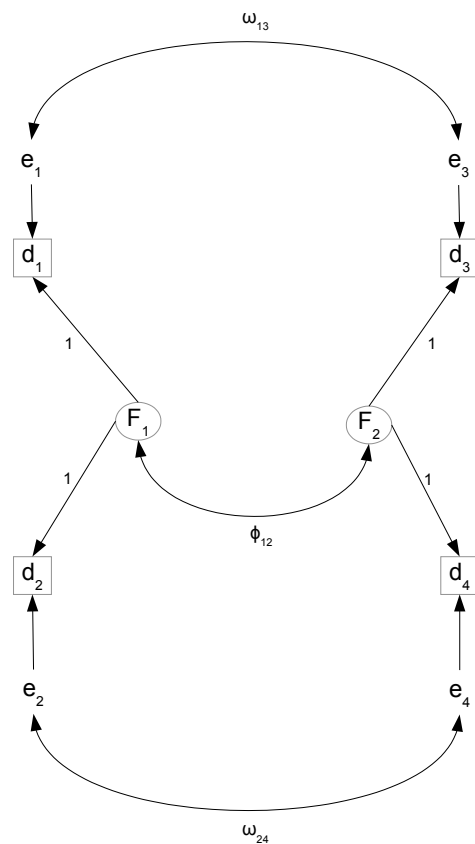
where $E(\mathbf{F}_i) = \mathbf{0}$, $cov(\mathbf{F}_i) = \mathbf{\Phi}$, \mathbf{F}_i has zero covariance with $\mathbf{e}_{i,1}$ and $\mathbf{e}_{i,2}$, $cov(\mathbf{e}_{i,1}) = \mathbf{\Omega}_1$, $cov(\mathbf{e}_{i,2}) = \mathbf{\Omega}_2$ and $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) = \mathbf{0}$.

The parameters in this model (which will be most useful as part of a larger model) are the unique elements of the matrices $\mathbf{\Phi}$, $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$. The double measurement rule (Rule 2a) says that these parameters are identifiable.

Three observed variables We now develop an identifiability rule in which for each factor, there are three observable variables of a certain kind. Figure 3.2 shows the path diagram when there is one factor. Here is a statement of the model. Independently for

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

Figure 3.1: Scalar Double Measurement



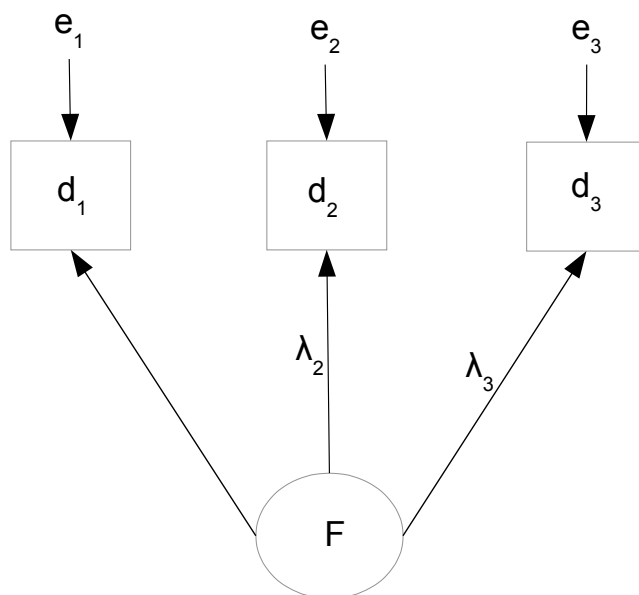
$i = 1, \dots, n$, let

$$\begin{aligned}
 d_{i,1} &= F_i + e_{i,1} \\
 d_{i,2} &= \lambda_2 F_i + e_{i,2} \\
 d_{i,3} &= \lambda_3 F_i + e_{i,3},
 \end{aligned} \tag{3.4}$$

with all expected values zero, $\text{Var}(F_i) = \phi > 0$, $\text{Var}(e_{i,j}) = \omega_j > 0$, and F_i and $e_{i,j}$ all independent. Note that this is a centered model, and that in the first equation, a factor loading that would be denoted λ_1 has been set to one. Centered variables and parameters equal to one are signs that it's a surrogate model.

The parameter vector is $\boldsymbol{\theta} = (\phi, \lambda_2, \lambda_3, \omega_1, \omega_2, \omega_3)$. There are six unknown parameters, and the covariance matrix of $(d_{i,1}, d_{i,2}, d_{i,3})^\top$ has six unique elements. This means that there are six covariance structure equations in six unknown parameters. If the parameters

Figure 3.2: One Unstandardized Factor, Three Observed Variables



are identifiable, they are just identifiable. Calculating the covariance matrix,

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{pmatrix} = \begin{array}{c|ccc} & d_1 & d_2 & d_3 \\ \hline d_1 & \phi + \omega_1 & \lambda_2\phi & \lambda_3\phi \\ d_2 & & \lambda_2^2\phi + \omega_2 & \lambda_2\lambda_3\phi \\ d_3 & & & \lambda_3^2\phi + \omega_3 \end{array}.$$

The covariance structure equations are

$$\begin{aligned} \sigma_{11} &= \phi + \omega_1 \\ \sigma_{12} &= \lambda_2\phi \\ \sigma_{13} &= \lambda_3\phi \\ \sigma_{22} &= \lambda_2^2\phi + \omega_2 \\ \sigma_{23} &= \lambda_2\lambda_3\phi \\ \sigma_{33} &= \lambda_3^2\phi + \omega_3. \end{aligned}$$

If $\lambda_2 = \lambda_3 = 0$, that fact can be determined from $\sigma_{12} = \sigma_{13} = 0$, so that λ_2 and λ_3 are identifiable. The parameters $\omega_2 = \sigma_{22}$ and $\omega_3 = \sigma_{33}$ are also identifiable. However, only the equation $\sigma_{11} = \phi + \omega_1$ remains, and there are infinitely many solutions. This means that at points in the parameter space where $\lambda_2 = \lambda_3 = 0$, only four of the six parameters are identifiable.

Suppose just one of λ_2 and λ_3 equals zero, say, λ_2 . In that case, λ_2 and ω_2 are identifiable, but the equation $\sigma_{23} = 0$ is essentially lost. By the [parameter count rule](#),

the remaining three equations in four unknowns do not have a unique solution, except possibly on a set of volume zero in that four-dimensional section of the parameter space. The conclusion is that the parameter vector is not identifiable at points where $\lambda_2 = 0$, $\lambda_3 = 0$, or both.

So assume that $\lambda_2 \neq 0$ and $\lambda_3 \neq 0$. This “assumption” means that we are considering points in the parameter space where both λ_2 and λ_3 are non-zero. In practical situations, it means that the variables d_2 and d_3 (and d_1 too, of course) need to be chosen so that they unquestionably reflect the underlying factor F . In this case, the covariance structure equations have the unique solution

$$\begin{aligned}\phi &= \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \\ \lambda_2 &= \sigma_{23}/\sigma_{13} \\ \lambda_3 &= \sigma_{23}/\sigma_{12} \\ \omega_1 &= \sigma_{11} - \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \\ \omega_2 &= \sigma_{22} - \frac{\sigma_{12}\sigma_{23}}{\sigma_{13}} \\ \omega_3 &= \sigma_{33} - \frac{\sigma_{13}\sigma_{23}}{\sigma_{12}}.\end{aligned}\tag{3.5}$$

Suppose we add another observed variable to the model: $d_{i,4} = \lambda_4 F_i + e_{i,4}$. The covariance matrix is now

$$\Sigma = \begin{pmatrix} \phi + \omega_1 & \lambda_2\phi & \lambda_3\phi & \lambda_4\phi \\ & \lambda_2^2\phi + \omega_2 & \lambda_2\lambda_3\phi & \lambda_2\lambda_4\phi \\ & & \lambda_3^2\phi + \omega_3 & \lambda_3\lambda_4\phi \\ & & & \lambda_4^2\phi + \omega_4 \end{pmatrix}.$$

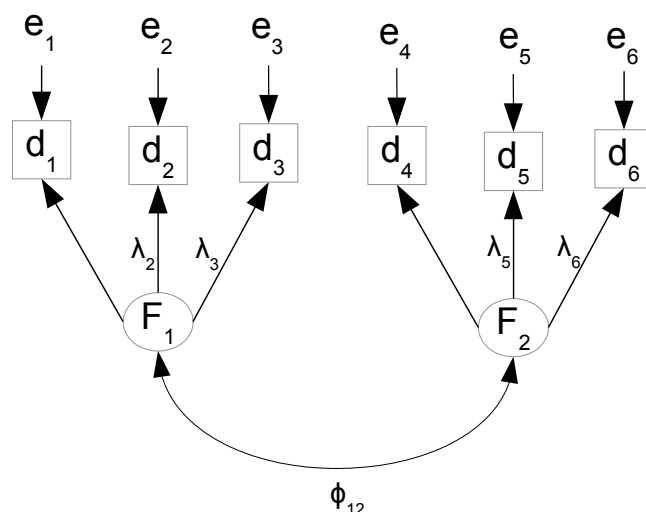
Whether or not $\lambda_4 = 0$, all the parameters are easily identifiable. For five observed variables, two loadings can be zero, and so on.

With more than three observed variables, the parameters are over-identified. In this case, testing model fit is a possibility. For example, if there are four observed variables, then there are eight parameters and ten covariance structure equations, giving rise to $10 - 8 = 2$ equality constraints on the covariance matrix.

Now add another factor to Model (3.4), as in Figure 3.3. A single factor loading has been set to one for each factor, $cov(\mathbf{F}_i) = \Phi = [\phi_{\ell j}]$, and $Var(e_j) = \omega_j$ for $j = 1, \dots, 6$. The model equations are

$$\begin{aligned}d_1 &= F_1 + e_1 \\ d_2 &= \lambda_2 F_1 + e_2 \\ d_3 &= \lambda_3 F_1 + e_3 \\ d_4 &= F_2 + e_4 \\ d_5 &= \lambda_5 F_2 + e_5 \\ d_6 &= \lambda_6 F_2 + e_6,\end{aligned}$$

Figure 3.3: Two Unstandardized Factors



and the covariance matrix of the observable variables is

$$\Sigma = \begin{pmatrix} \omega_1 + \phi_{11} & \lambda_2\phi_{11} & \lambda_3\phi_{11} & \phi_{12} & \lambda_5\phi_{12} & \lambda_6\phi_{12} \\ & \lambda_2^2\phi_{11} + \omega_2 & \lambda_2\lambda_3\phi_{11} & \lambda_2\phi_{12} & \lambda_2\lambda_5\phi_{12} & \lambda_2\lambda_6\phi_{12} \\ & & \lambda_3^2\phi_{11} + \omega_3 & \lambda_3\phi_{12} & \lambda_3\lambda_5\phi_{12} & \lambda_3\lambda_6\phi_{12} \\ & & & \omega_4 + \phi_{22} & \lambda_5\phi_{22} & \lambda_6\phi_{22} \\ & & & & \lambda_5^2\phi_{22} + \omega_5 & \lambda_5\lambda_6\phi_{22} \\ & & & & & \lambda_6^2\phi_{22} + \omega_6 \end{pmatrix}$$

Typesetting that covariance matrix would have been a chore. SageMath kindly agreed to do it for me; then I manually removed the lower triangle to make the matrix easier to look at. Here is the code.

```
sem = 'http://www.utstat.toronto.edu/~brunner/openSEM/sage/sem.sage'
load(sem)
# Two unstandardized factors
L = ZeroMatrix(6,2)
L[0,0]= 1; L[1,0]= var('lambda2'); L[2,0]= var('lambda3')
L[3,1]= 1; L[4,1]= var('lambda5'); L[5,1]= var('lambda6'); L
P = SymmetricMatrix(2,'phi'); P
O = DiagonalMatrix(6,symbol='omega'); O
Sig = FactorAnalysisCov(L,P,O); Sig
print(latex(Sig))
```

Assuming λ_2 , λ_3 , λ_5 and λ_6 to be non-zero, these factor loadings along with ϕ_{11} , ϕ_{22} and $\omega_1, \dots, \omega_6$ may be recovered as for the one-factor model. The remaining parameter, ϕ_{12} , is identified from $\phi_{12} = \sigma_{14}$. Thus, all the parameters are identifiable. Identifiability

is preserved when more factors are added under these same conditions. Adding more variables in any set also does no harm.

Reference variables We are at the point of stating an important general rule, but first, please notice a special feature of the observed variables in the models we have been considering. Each observed variables is influenced by only one factor and an error term. This is almost never seen in exploratory factor analysis, except that it might be considered an extreme case of Thurstone’s “simple structure.” In confirmatory factor analysis models, such variables are quite common, and it helps to have a name for them. The term is taken from Jöreskog’s (1969) classic article in *Psychometrika* [36].

Definition 3.1 *A reference variable for a latent variable is an observable variable that is a function only of that latent variable and an error term. The factor loading is non-zero.*

Obviously, not all observable variables are reference variables by this definition. For example, in the two-factor model of Figure 2.1 on page 214, there are no reference variables at all. In the latent variable regression model of Figure 13 on page 49, W_1 and W_2 are reference variables, but Y is not. Reference variables are very useful for establishing identifiability, and many of the standard sufficient conditions for parameter identifiability involve reference variables for the latent variables.

Before the introduction of reference variables, the following rule was established. If the conditions seem overly restrictive, I agree. We can and will do better.

Three-variable Rule for Unstandardized Factors The parameters of a factor analysis model are identifiable provided

- There are at least three reference variables for each factor.
- For each factor, the factor loading of at least one reference variable is equal to one.
- Errors are independent of one another and of the factors.

Only one reference variable per factor is really needed. The three-variable rule is widely used in practice, but it is more restrictive than it needs to be. It is a lot to ask that each factor have *three* observed variables that are influenced by that factor and none of the others. It’s tough enough to come up with one such pure measurement for each factor. Fortunately, it turns out that only one of the three observed variables for each factor needs to be a reference variable. The other two can be influenced by all the factors.

The reference variable rule is a matrix version of the three-variable rule. For it to apply, there must be at least three observable variables for every factor, including one reference variable per factor. The observable variables are collected into three or possibly four vectors. For case i (there are n cases), $\mathbf{d}_{i,1}$ contains reference variables for the factors, with the factor loadings for the reference variables set to one. The number of variables in $\mathbf{d}_{i,2}$ and $\mathbf{d}_{i,3}$ is also equal to the number of factors. If there are any more observable variables, they are placed in $\mathbf{d}_{i,4}$.

Here is the model. Independently for $i = 1, \dots, n$,

$$\begin{aligned} \mathbf{d}_{i,1} &= \mathbf{F}_i + \mathbf{e}_{i,1} \\ \mathbf{d}_{i,2} &= \mathbf{\Lambda}_2 \mathbf{F}_i + \mathbf{e}_{i,2} \\ \mathbf{d}_{i,3} &= \mathbf{\Lambda}_3 \mathbf{F}_i + \mathbf{e}_{i,3} \\ \mathbf{d}_{i,4} &= \mathbf{\Lambda}_4 \mathbf{F}_i + \mathbf{e}_{i,4}, \end{aligned} \tag{3.6}$$

where

- $\mathbf{d}_{i,1}$, $\mathbf{d}_{i,2}$ and $\mathbf{d}_{i,3}$ are $p \times 1$ observable random vectors. If $\mathbf{d}_{i,4}$ is present, it is an $m \times 1$ observable random vector.
- \mathbf{F}_i (F for Factor) is a $p \times 1$ latent random vector with expected value zero $cov(\mathbf{F}_i) = \mathbf{\Phi}$.
- $\mathbf{\Lambda}_2$ and $\mathbf{\Lambda}_3$ are $p \times p$ non-singular matrices of constants.
- $\mathbf{\Lambda}_4$, if it is present, is an m by p matrix of constants.
- $\mathbf{e}_{i,1}, \dots, \mathbf{e}_{i,4}$ are vectors of error terms, with expected value zero, covariance matrix $cov(\mathbf{e}_{i,j}) = \mathbf{\Omega}_{j,j}$ for $j = 1, \dots, 4$, and
 - $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) = cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,3}) = cov(\mathbf{e}_{i,2}, \mathbf{e}_{i,3}) = \mathbf{O}$, all $p \times p$ matrices.
 - $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,4}) = \mathbf{O}$, a $p \times m$ matrix.
 - $cov(\mathbf{e}_{i,2}, \mathbf{e}_{i,4}) = \mathbf{\Omega}_{2,4}$ and $cov(\mathbf{e}_{i,3}, \mathbf{e}_{i,4}) = \mathbf{\Omega}_{3,4}$.

The parameters of this model are the unique elements of the matrices $\mathbf{\Phi}$, $\mathbf{\Lambda}_2$, $\mathbf{\Lambda}_3$, $\mathbf{\Omega}_{1,1}$, $\mathbf{\Omega}_{2,2}$ and $\mathbf{\Omega}_{3,3}$. If there are more than $3p$ observable variables and $\mathbf{d}_{i,4}$ is necessary, the list of parameter matrices also includes $\mathbf{\Lambda}_4$, $\mathbf{\Omega}_{2,4}$, $\mathbf{\Omega}_{3,4}$ and $\mathbf{\Omega}_{4,4}$.

Detailed discussion of this model is deferred until Section 3.4. For now, just note that while the [three-variable rule](#) allows observable variables to be influenced by only a single factor, Model (3.6) says that at least two-thirds of the variables can be influenced by all the factors, through the matrices $\mathbf{\Lambda}_2$ and $\mathbf{\Lambda}_3$ (and possibly $\mathbf{\Lambda}_4$). Also, the three-variable rule requires all error terms to have zero covariance. In Model (3.6), however, the first $3p$ variables are divided into sets; error terms are allowed to have non-zero covariance within sets, but not between sets. If there is a fourth set of variables, its set of error terms may be correlated with the error terms of sets two and three, as well as with each other — but not with the error terms of set one.

Identifiability The covariance matrix of the observable variables may be written as a partitioned matrix.

$$\begin{aligned} \Sigma = \text{cov} \begin{pmatrix} \mathbf{d}_{i,1} \\ \mathbf{d}_{i,2} \\ \mathbf{d}_{i,3} \\ \mathbf{d}_{i,4} \end{pmatrix} &= \begin{pmatrix} \Sigma_{1,1} & \Sigma_{1,2} & \Sigma_{1,3} & \Sigma_{1,4} \\ & \Sigma_{2,2} & \Sigma_{2,3} & \Sigma_{2,4} \\ & & \Sigma_{3,3} & \Sigma_{3,4} \\ & & & \Sigma_{4,4} \end{pmatrix} \\ &= \begin{pmatrix} \Phi + \Omega_{1,1} & \Phi \Lambda_2^\top & \Phi \Lambda_3^\top & \Phi \Lambda_4^\top \\ & \Lambda_2 \Phi \Lambda_2^\top + \Omega_{2,2} & \Lambda_2 \Phi \Lambda_3^\top & \Lambda_2 \Phi \Lambda_4^\top + \Omega_{2,4} \\ & & \Lambda_3 \Phi \Lambda_3^\top + \Omega_{3,3} & \Lambda_3 \Phi \Lambda_4^\top + \Omega_{3,4} \\ & & & \Lambda_4 \Phi \Lambda_4^\top + \Omega_{4,4} \end{pmatrix}. \end{aligned} \quad (3.7)$$

Viewing (3.7) as a compact way to express the covariance structure equations, one obtains solutions that are directly analogous to (3.5). To avoid transpose signs, the solutions use $\Sigma_{i,j}^\top = \Sigma_{j,i}$.

$$\begin{aligned} \Phi &= \Sigma_{1,3} \Sigma_{2,3}^{-1} \Sigma_{2,1} \\ \Lambda_2 &= \Sigma_{2,3} \Sigma_{1,3}^{-1} \\ \Lambda_3 &= \Sigma_{3,2} \Sigma_{1,2}^{-1} \\ \Omega_{1,1} &= \Sigma_{1,1} - \Sigma_{1,3} \Sigma_{2,3}^{-1} \Sigma_{2,1} \\ \Omega_{2,2} &= \Sigma_{2,2} - \Sigma_{2,1} \Sigma_{3,1}^{-1} \Sigma_{3,2} \\ \Omega_{3,3} &= \Sigma_{3,3} - \Sigma_{3,2} \Sigma_{1,2}^{-1} \Sigma_{1,3}. \end{aligned} \quad (3.8)$$

In case there are more than $3p$ observed variables and $\mathbf{d}_{i,4}$ is needed, solutions for the additional parameter matrices are

$$\begin{aligned} \Lambda_4 &= \Sigma_{4,1} \Sigma_{2,1}^{-1} \Sigma_{2,3} \Sigma_{1,3}^{-1} \\ \Omega_{2,4} &= \Sigma_{2,4} - \Sigma_{2,3} \Sigma_{1,3}^{-1} \Sigma_{1,4} \\ \Omega_{3,4} &= \Sigma_{3,4} - \Sigma_{3,2} \Sigma_{1,2}^{-1} \Sigma_{1,4} \\ \Omega_{4,4} &= \Sigma_{4,4} - \Sigma_{4,1} \Sigma_{3,1}^{-1} \Sigma_{3,2} \Sigma_{1,2}^{-1} \Sigma_{1,4}. \end{aligned} \quad (3.9)$$

This establishes identifiability of all the parameters, except on that set of volume zero in the parameter space where Λ_2 and Λ_3 do not have inverses. This is like the requirement that λ_2 and λ_3 be non-zero in the three-variable model (1usfactor), so that one may “divide” by them. It’s a set of volume zero because if Λ_2 or Λ_3 were singular, then the columns would be linearly dependent, and at least one column would be a perfect linear combination of the others.

We have the following important rule. Full discussion will be deferred until Section 3.4, where an even stronger version will be given.

The Reference Variable Rule for Unstandardized Factors The parameters of a factor analysis model are identifiable except possibly on a set of volume zero in the parameter space, provided

- The number of observable variables (including reference variables) is at least three times the number of factors.
- For each factor, there is at least one reference variable, with a factor loading of one.
- Divide the observable variables into sets. The first set contains one reference variable for each factor; the factor loadings all equal one. The number of variables in the second set and the number in the third set is also equal to the number of factors. The fourth set may contain any number of additional variables, including zero. The error terms for the variables in the first three sets may have non-zero covariance within sets, but not between sets. The error terms for the variables in the fourth set may have non-zero covariance within the set, and with the error terms of sets two and three, but they must have zero covariance with the error terms of the reference variables.

Two reference variables per factor In some models, a factor may influence fewer than three observable variables, a condition that would force either $\mathbf{\Lambda}_2$ or $\mathbf{\Lambda}_3$ to be singular in the preceding discussion. If the model has at least two such factors and non-zero covariance between the factors, we can get away with two reference variables for each factor. Understanding that this may be only part of a larger model, the model equations would be

$$\begin{aligned}d_1 &= F_1 + e_1 \\d_2 &= \lambda_2 F_1 + e_2 \\d_3 &= F_2 + e_3 \\d_4 &= \lambda_4 F_2 + e_4,\end{aligned}$$

with all expected values zero, $\text{cov} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \mathbf{\Phi} = [\phi_{ij}]$, $\text{Var}(e_j) = \omega_j$, and the error terms independent of the factors and each other. An additional critical stipulation is that $\text{Cov}(F_1, F_2) = \phi_{12} \neq 0$.

The covariance matrix of the observable variables is

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ & & \sigma_{33} & \sigma_{34} \\ & & & \sigma_{44} \end{pmatrix} = \begin{pmatrix} \phi_{11} + \omega_1 & \lambda_2 \phi_{11} & \phi_{12} & \lambda_4 \phi_{12} \\ & \lambda_2^2 \phi_{11} + \omega_2 & \lambda_2 \phi_{12} & \lambda_2 \lambda_4 \phi_{12} \\ & & \phi_{22} + \omega_3 & \lambda_4 \phi_{22} \\ & & & \lambda_4^2 \phi_{22} + \omega_4 \end{pmatrix} \quad (3.10)$$

Provided $\phi_{12} \neq 0$, all the parameters are easily identifiable. If $\phi_{12} = 0$, then that parameter is identifiable, but then identifying the other parameters would require a unique solution to six equations in eight unknowns. By the [parameter count rule](#), this is impossible in most of the parameter space. Thus, identifiability requires $\phi_{12} \neq 0$.

With more factors and two observed variables per factor, identifiability is maintained provided that each factor has a non-zero covariance with at least one other factor. Naturally, three or more variables for some of the factors is okay. We have the following rule.

Two-variable Rule for Unstandardized Factors The parameters of a factor analysis model are identifiable provided

- There are at least two factors.
- There are at least two reference variables per factor.
- For each factor, the factor loading of at least one reference variable is equal to one.
- Each factor has a non-zero covariance with at least one other factor.
- Errors are independent of one another and of the factors.

Re-parameterization and surrogate models Setting some of the factor loadings to one is a useful technical device for obtaining identifiability, but does the resulting model make sense? When a factor loading is set to one, it means that the observed variable is just the factor plus a piece of random noise. Models like this were common in Chapter 0, but the regression-like “original” model with a slope possibly not equal to unity is much easier to believe. While it may be true that “all models are wrong²,” it is still not a good idea adopt models that are obviously unrealistic, unless there is a good reason.

A common justification for setting factor loadings to one to describe the process as “setting the scale,” as in Bollen [10] (p. 198). Suppose the latent variable is length, and it is measured twice. One measurement is in inches, and the other is in centimeters. What’s the scale of measurement of the latent variable? This is un-knowable³ and not very interesting anyway, so the scale of the latent variable is arbitrarily made to agree with one of the observed variables, by setting its factor loading to one.

As I see it, this “setting the scale” interpretation does not really hold up. Suppose the latent variable is amount of debt, measured in dollars. One of the observed variables is reported debt, also in dollars. Clearly, the latent and observed variables are on the same scale. I think the factor loading could easily be a constant strictly less than one, so that, for example, for every one dollar increase in true debt, the average person might report 75 cents. Setting the factor loading to one when it is really 0.75 would be to model an interesting phenomenon out of existence. There must be some other explanation for setting a factor loading to one.

To see what is really going on, consider the following one-factor example. None of the factor loadings is necessarily equal to one.

Example 3.1.1 *A centered model with one factor.*

²The famous quote is from Box and Draper (1987, p. 424) who said, “Essentially all models are wrong, but some are useful.” [11]

³A symptom of non-identifiability.

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

$$\begin{aligned}d_{i,1} &= \lambda_1 F_i + e_{i,1} \\d_{i,2} &= \lambda_2 F_i + e_{i,2} \\d_{i,3} &= \lambda_3 F_i + e_{i,3} \\d_{i,4} &= \lambda_4 F_i + e_{i,4},\end{aligned}$$

with all expected values zero, $\text{Var}(F_i) = \phi$, $\text{Var}(e_{i,j}) = \omega_j$, and F_i and $e_{i,j}$ all independent.

As usual, identifiability is to be established by solving the covariance structure equations for the unknown parameters. There are nine unknown parameters, and the covariance matrix of the observable variables has ten unique variances and covariances. The [parameter count rule](#) says that identifiability is possible, but not guaranteed. The covariance matrix is

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ & & \sigma_{33} & \sigma_{34} \\ & & & \sigma_{44} \end{pmatrix} = \begin{pmatrix} \lambda_1^2 \phi + \omega_1 & \lambda_1 \lambda_2 \phi & \lambda_1 \lambda_3 \phi & \lambda_1 \lambda_4 \phi \\ & \lambda_2^2 \phi + \omega_2 & \lambda_2 \lambda_3 \phi & \lambda_2 \lambda_4 \phi \\ & & \lambda_3^2 \phi + \omega_3 & \lambda_3 \lambda_4 \phi \\ & & & \lambda_4^2 \phi + \omega_4 \end{pmatrix} \quad (3.11)$$

If two distinct parameter sets yield the same covariance matrix, the parameter vector is not identifiable. Table 3.1 shows two such parameter sets — actually, infinitely many.

Table 3.1: Non-identifiability

θ_1	ϕ	λ_1	λ_2	λ_3	λ_4	ω_1	ω_2	ω_3	ω_4
θ_2	ϕ/c^2	$c\lambda_1$	$c\lambda_2$	$c\lambda_3$	$c\lambda_4$	ω_1	ω_2	ω_3	ω_4

For any $c \neq 0$, both θ_1 and θ_2 yield the covariance matrix in (3.11).

As usual when parameters are not identifiable, this is a serious problem. Regardless of what the true parameter values are, there are infinitely many sets of *untrue* parameter values that yield exactly the same Σ . Since inference is based on the covariance matrix of the observable data, there is no way to even approach the full truth based on the data, no matter how large the sample size. However, there is a way to get at the partial truth, because certain *functions* of the parameter vector are identifiable. For example, at points in the parameter space where $\lambda_1, \lambda_2, \lambda_3 \neq 0$,

- $\sigma_{11} - \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} = \omega_1$, and the other error variances are identifiable by a similar calculation.
- $\frac{\sigma_{13}}{\sigma_{23}} = \frac{\lambda_1 \lambda_3 \phi}{\lambda_2 \lambda_3 \phi} = \frac{\lambda_1}{\lambda_2}$, so *ratios* of factor loadings are identifiable.

- If the sign of one factor loading is known (say by naming the factor⁴), then the signs of the others can be identified from the covariances in (3.11).
- $\frac{\sigma_{12}\sigma_{13}}{\sigma_{23}\sigma_{11}} = \frac{\lambda_1^2\phi}{\lambda_1^2\phi + \omega_1}$, the reliability of d_1 as a measure of F . Reliabilities are identifiable for this model.

The point here is that while the entire parameter vector may not be identifiable, the covariance matrix still contains useful information about the parameters. It's difficult to get at, though. If we try to fit a model like the one in Example (3.1) by maximum likelihood, lack of identifiability will cause the likelihood function to have a maximum that is not unique, and unpleasant numerical things will happen.

The solution is re-parameterization. It cannot be a one-to-one re-parametrization, because that would leave the identifiability of the model parameters unchanged. Instead, it's a sort of collapsing re-parameterization, one that results in a parameter space of lower dimension. It is accomplished by a change of variables, and the resulting model is a surrogate model. We have already seen how a change of variables is used to transform the original model (3.1) into the centered surrogate model (3.3).

In the model of Example 3.1, assume $\lambda_1 \neq 0$. It can be made positive by naming the factor appropriately, so let $\lambda_1 > 0$. Setting $F' = \lambda_1 F$, we have $d_1 = F' + e_1$, and it *looks* as if λ_1 has been set to one (if you ignore the prime). The consequences for the other factor loadings are, for example,

$$\begin{aligned} d_2 &= \lambda_2 F + e_2 \\ &= \left(\frac{\lambda_2}{\lambda_1}\right) (\lambda_1 F) + e_2 \\ &= \lambda'_2 F' + e_2, \end{aligned}$$

and we have

$$\begin{aligned} d_1 &= F' + e_1 \\ d_2 &= \lambda'_2 F' + e_2 \\ d_3 &= \lambda'_3 F' + e_3, \text{ etc.} \end{aligned}$$

Losing the primes, the result looks exactly like Model (3.4). It is a surrogate for the model of Example 3.1, except that there are three factors instead of four. In terms of the original model parameters, the parameter λ_2 is really λ_2/λ_1 . The variance parameter ϕ is really $\lambda_1^2\phi$. As shown in Table 3.2, these are identifiable functions of the parameters of the original model.

⁴Suppose that the factor is left-right political orientation. Do extremely high scores reflect right-wing ideology, or left-wing ideology? Nobody knows. However, you have an observed variable, score on a questionnaire asking about politics. It is *scored* so that agreement with certain statements gets you a higher Left score or a higher Right score. Which one? It's up to the investigator. So just make a choice, and assume that the factor loading is positive. This way, you have decided whether to call the factor "Left-wing orientation" or "Right-wing orientation." This always works, but you only want to do it when the connection between the factor and the observable variable is completely clear and non-controversial.

Table 3.2: Identifiable Functions in Model (3.4)

Function of Σ	Value under model	
	Surrogate	Original
σ_{23}/σ_{13}	λ_2	λ_2/λ_1
σ_{23}/σ_{12}	λ_3	λ_3/λ_1
$\sigma_{12}\sigma_{13}/\sigma_{23}$	ϕ	$\lambda_1^2\phi$

Suppose there is more than one factor, with a factor loading set to one for each factor. Then

$$\begin{aligned}
 \phi'_{12} &= \text{Cov}(F'_1, F'_2) \\
 &= \text{Cov}(\lambda_1 F_1, \lambda_4 F_2) \\
 &= \lambda_1 \lambda_4 \text{Cov}(F_1, F_2) \\
 &= \lambda_1 \lambda_4 \phi_{12}.
 \end{aligned}$$

To summarize, setting a factor loading to one for each factor is not an arbitrary restriction of the parameter space. It is a very useful re-parameterization, resulting in a surrogate model. The parameters of the surrogate model are identifiable functions of the original model parameters. Their meanings are limited but clear. Everything is relative to the values of the parameters that have apparently been suppressed. The error variances ω_j are unaffected, but all the other parameters are positive multiples of the corresponding parameters of the original model. Any estimated factor loading or covariance is really an estimate of that quantity times an unknown positive constant. If the latent variable model has a causal structure (rather than just covariances between factors), the re-parameterization has cascading effects that run down the chain of causality.

Unless one is actually interested in ratios of factor loadings, point and interval estimates of the surrogate model parameters are not very meaningful. However, a test of whether a surrogate model parameter is positive, negative or zero is also a valid test about the original model parameter. This is good enough for many applications. In the social and biological sciences, the primary research question is often whether a relationship between variables exists, and if so, whether the relationship is positive or negative. In such cases, setting factor loadings to one can be an excellent way to achieve parameter identifiability and get on with the data analysis.

3.2 Standardized Factors

Setting a factor loading to one for each factor is path to identifiability. The other common trick is to set the variances of the factors to one. Please consider again the one-factor model of Example 3.1 on page 288. Recall that this model is centered, but otherwise it has not been re-parameterized, and its parameters are not identifiable. To obtain $\text{Var}(F') = 1$

in a re-parameterized model, let $F' = \frac{1}{\phi^{1/2}}F$. Then $Var(F') = \frac{1}{\phi}Var(F) = 1$ as desired, and

$$\begin{aligned} d_j &= \lambda_j F + e_j \\ &= \lambda_j \phi^{1/2} \frac{1}{\phi^{1/2}} F + e_j \\ &= \lambda'_j F' + e_j. \end{aligned}$$

Under the new, re-parameterized model, the factor loading is expressed as a multiple of the unknown standard deviation of the factor; $\lambda'_j = \lambda_j \phi^{1/2}$ is the expected increase in d_j when F is increased by one standard deviation unit. Since the standard deviation is unknown (and un-knowable) except for being positive, this means that an estimate of λ'_j could be informative about the sign of the original factor loading, but that's all.

Discarding the primes, we have a surrogate model. Consider the following three-variable version. Independently for $i = 1, \dots, n$, let

$$\begin{aligned} d_{i,1} &= \lambda_1 F_i + e_{i,1} \\ d_{i,2} &= \lambda_2 F_i + e_{i,2} \\ d_{i,3} &= \lambda_3 F_i + e_{i,3}, \end{aligned}$$

with all expected values zero, $Var(F_i) = 1$, $Var(e_{i,j}) = \omega_j$ and F_i and $e_{i,j}$ all independent.

The covariance matrix of an observable data vector is,

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ & & \sigma_{33} \end{pmatrix} = \begin{pmatrix} \lambda_1^2 + \omega_1 & \lambda_1 \lambda_2 & \lambda_1 \lambda_3 \\ & \lambda_2^2 + \omega_2 & \lambda_2 \lambda_3 \\ & & \lambda_3^2 + \omega_3 \end{pmatrix}. \quad (3.12)$$

There are six covariance structure equations in six unknown parameters. If two or three of the factor loadings are equal to zero, all three covariances equal zero, it's impossible to tell whether two loadings or three equal zero, and none of the parameters is identifiable. So, consider what happens when just one factor loading equals zero – say, λ_1 . Since $\sigma_{12} = \sigma_{13} = 0$ but $\sigma_{23} \neq 0$, it is clear that $\lambda_1 = 0$. That is, its value is identifiable. Also, $\sigma_{11} = \omega_1$, and ω_1 is identifiable. However, the covariance structure equation $\sigma_{23} = \lambda_2 \lambda_3$ has infinitely many solutions; identification of ω_2 and ω_3 is also impossible.

Accordingly, assume that λ_1 , λ_2 and λ_3 are all non-zero. As in Section 1.3 of Chapter 1, this is an acknowledgement that parameter identifiability need not be the same in different regions of the parameter space. Viewing (3.12) as a compact statement of the covariance

structure equations and trying to solve, we have

$$\begin{aligned}
 \lambda_1^2 &= \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} = \frac{\lambda_1\lambda_2\lambda_1\lambda_3}{\lambda_2\lambda_3} \\
 \lambda_2^2 &= \frac{\sigma_{12}\sigma_{23}}{\sigma_{13}} \\
 \lambda_3^2 &= \frac{\sigma_{13}\sigma_{23}}{\sigma_{12}} \\
 \omega_1 &= \sigma_{11} - \frac{\sigma_{12}\sigma_{13}}{\sigma_{23}} \\
 \omega_2 &= \sigma_{22} - \frac{\sigma_{12}\sigma_{23}}{\sigma_{13}} \\
 \omega_3 &= \sigma_{33} - \frac{\sigma_{13}\sigma_{23}}{\sigma_{12}}
 \end{aligned} \tag{3.13}$$

The error variances are identifiable, but only the squares of the factor loadings can be uniquely identified. To see this clearly, note that if all three λ_j are replaced with $-\lambda_j$, we get same Σ . The likelihood function will have two maxima, of the same height. Which one is located will depend on where the numerical search starts.

The solution is to decide on the sign of one factor loading. It really is a decision that is up to the user, and it's based on the *meaning* of the hypothesized factor. If the three variables are scores on three math tests, is F math ability, or math inability? You decide. Once the sign of one loading is fixed, the signs of the other two may be determined from the signs of the σ_{ij} . Identifiability is purchased by cutting the parameter space in half, but it really doesn't cost anything.

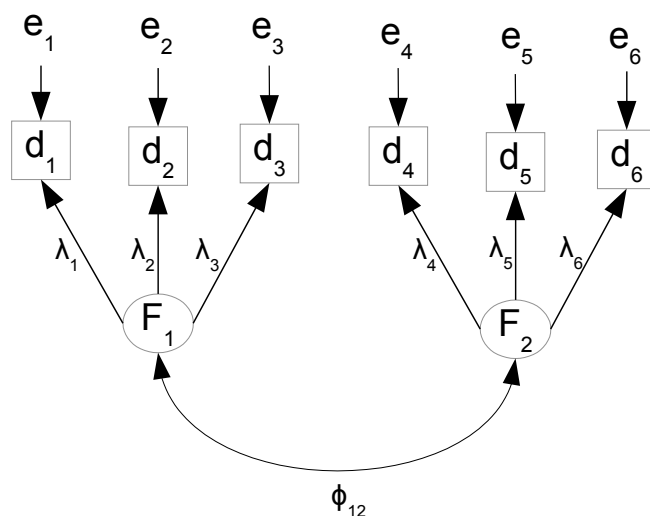
Now suppose we add another observed variable to the model: $d_{i,4} = \lambda_4 F_i + e_{i,4}$. The covariance matrix is

$$\Sigma = \begin{pmatrix} \lambda_1^2 + \omega_1 & \lambda_1\lambda_2 & \lambda_1\lambda_3 & \lambda_1\lambda_4 \\ & \lambda_2^2 + \omega_2 & \lambda_2\lambda_3 & \lambda_2\lambda_4 \\ & & \lambda_3^2 + \omega_3 & \lambda_3\lambda_4 \\ & & & \lambda_4^2 + \omega_4 \end{pmatrix}. \tag{3.14}$$

The parameters will all be identifiable as long as three out of four loadings are non-zero, and one sign is known. For example, if only $\lambda_1 = 0$ then the top row = 0, and it is possible to solve for $\lambda_2, \lambda_3, \lambda_4$ as before. For five observed variables, two loadings can be zero, and so on. With more than three observed variables, the parameters are over-identified. For example, with four observed variables, there are eight parameters and ten covariance structure equations, giving rise to $10 - 8 = 2$ equality constraints on the covariance matrix.

Returning to three observed variables per factor, add another factor as in Figure 3.4. The variances of both factors equal one, and $Var(e_j) = \omega_j$ for $j = 1, \dots, 6$. The model

Figure 3.4: Two factors



equations are

$$\begin{aligned}
 d_1 &= \lambda_1 F_1 + e_1 \\
 d_2 &= \lambda_2 F_1 + e_2 \\
 d_3 &= \lambda_3 F_1 + e_3 \\
 d_4 &= \lambda_4 F_2 + e_4 \\
 d_5 &= \lambda_5 F_2 + e_5 \\
 d_6 &= \lambda_6 F_2 + e_6,
 \end{aligned}$$

and the covariance matrix of the observable variables is

$$\Sigma = \begin{pmatrix}
 \lambda_1^2 + \omega_1 & \lambda_1 \lambda_2 & \lambda_1 \lambda_3 & \lambda_1 \lambda_4 \phi_{12} & \lambda_1 \lambda_5 \phi_{12} & \lambda_1 \lambda_6 \phi_{12} \\
 & \lambda_2^2 + \omega_2 & \lambda_2 \lambda_3 & \lambda_2 \lambda_4 \phi_{12} & \lambda_2 \lambda_5 \phi_{12} & \lambda_2 \lambda_6 \phi_{12} \\
 & & \lambda_3^2 + \omega_3 & \lambda_3 \lambda_4 \phi_{12} & \lambda_3 \lambda_5 \phi_{12} & \lambda_3 \lambda_6 \phi_{12} \\
 & & & \lambda_4^2 + \omega_4 & \lambda_4 \lambda_5 & \lambda_4 \lambda_6 \\
 & & & & \lambda_5^2 + \omega_5 & \lambda_5 \lambda_6 \\
 & & & & & \lambda_6^2 + \omega_6
 \end{pmatrix}$$

Assuming that all the factor loadings are non-zero and the sign of one factor loading is known in each set (one set per factor), $\lambda_1, \lambda_2, \lambda_3$ may be identified from set One and $\lambda_4, \lambda_5, \lambda_6$ may be identified from set Two. Then ϕ_{12} may be identified from any unused covariance, and the ω_j are identifiable from the variances. Thus, all the parameters are identifiable.

Adding more standardized factors, the parameters remain identifiable provided there are at least three variables for each factor with non-zero factor loadings, and the sign of

one factor loading is known in each set. Adding more variables in any set also does no harm.

This establishes the three-variable rule for standardized factors. The parameters will be identifiable provided that there are at least three reference variables per factor, and the errors are independent of one another and of the factors. Comparing these conditions to the three-variable rule for unstandardized factors on page 284, we see the only difference is that the variance of the factor equal to one (and one sign known) is substituted for the factor loading of one (in which case its sign is positive). The result is the following widely-used rule.

Rule 2b: Three-variable Rule The parameters of a factor analysis model are identifiable provided

- There are at least three reference variables for each factor.
- For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading for at least one reference variable is equal to one.
- Errors are independent of one another and of the factors.

3.3 Equivalence of the Surrogate Models

The three-variable rules for standardized and unstandardized factors were very similar, and it was quite easy to combine them into a single rule. The extreme similarity suggests that the two common surrogate models — the one with a factor loading set to one for each factor, and the one with the variances of the factors set to one — might be the same thing in disguise. In fact, this is correct. The surrogate models in question are one-to-one.

What this means is that if the parameters of the two surrogate models are expressed in terms of the parameters of the original model, then there is a one-to-one (injective) function connecting their parameter vectors. There are two important consequences. First, if the parameters of one surrogate model are shown to be a function of Σ and hence identifiable, then the parameters of the other surrogate model are immediately identified as well. This means that it is permissible to check identifiability for one surrogate model even when you intend to fit the other one to your data. Usually, this means doing calculations for a model with factor loadings set to one.

The other consequence is that since the parameter vectors of the two surrogate models are one to one, they capture the same information about the parameters of the original model — and again, the original model is what we really care about. In this sense, the two surrogate models are equally good. However, the *form* of the information may be more convenient for one of the models, depending on the interests and research objectives of the investigator. Section 3.8 includes examples of extracting the same information the easy way and the hard way.

3.3.1 Demonstration of Equivalence

The following is fully rigorous, but much too long and chatty to be called a proof. It is based on three surrogate models. All three models have p factors and k observable variables, and error terms independent of the factors. The first model will be called the *centered original model*. In this model, the intercepts and non-zero expected values in the original model have been swallowed by a re-parameterization, as in the “centered surrogate model” (3.3) on page 278. Note that while the centered original model is a surrogate model, the factor loadings, the covariance matrix of the factors, and the covariance matrix of the error terms are all identical to their counterparts in the original model.

In the centered original model, the observed variables are sorted into two vectors; this is the only difference between the present centered original model and the earlier [centered surrogate model](#). For case i , (there are n cases), $\mathbf{d}_{i,1}$ consists of p reference variables for the factors. These are the best available representatives of the factors. If the factors are named appropriately, the factor loadings linking each factor to its reference variable may be assumed strictly positive. The remaining $k - p$ observed variables are collected into $\mathbf{d}_{i,2}$. In the equations for the centered original model, the subscript i is suppressed⁵ to reduce notational clutter. Implicitly, everything is independent for $i = 1, \dots, n$. The model equations are

$$\begin{aligned}\mathbf{d}_1 &= \mathbf{\Lambda}_1 \mathbf{F} + \mathbf{e}_1 \\ \mathbf{d}_2 &= \mathbf{\Lambda}_2 \mathbf{F} + \mathbf{e}_2,\end{aligned}\tag{3.15}$$

where all expected values are zero, the $p \times p$ matrix $\mathbf{\Lambda}_1$ is diagonal with positive diagonal elements, $cov(\mathbf{F}) = \mathbf{\Phi}$, $cov(\mathbf{e}_1) = \mathbf{\Omega}_1$, $cov(\mathbf{e}_2) = \mathbf{\Omega}_2$ and $cov(\mathbf{e}_1, \mathbf{e}_2) = \mathbf{\Omega}_{1,2}$. The parameter vector for this model is

$$\boldsymbol{\theta} = (\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Phi}, \mathbf{\Omega}_1, \mathbf{\Omega}_2, \mathbf{\Omega}_{1,2}).$$

Of course, only the non-redundant elements of the covariance matrices are intended as part of the parameter vector.

The centered original model is then re-parameterized in two different ways, leading to two further surrogate models. These models will cleverly be called Model One and Model Two. Figure 3.5 indicates the process.

In Model One, the number one figures prominently, because it looks like the non-zero factor loadings in $\mathbf{\Lambda}_1$ have all been set to one. That is, a factor loading appears to have been set to one for each factor. Actually, it is a re-parameterization accomplished by a change of variables. Letting $\mathbf{F}' = \mathbf{\Lambda}_1 \mathbf{F}$ yields

$$\begin{aligned}\mathbf{d}_1 &= \mathbf{F}' + \mathbf{e}_1 \\ \mathbf{d}_2 &= \mathbf{\Lambda}_2 \mathbf{\Lambda}_1^{-1} \mathbf{\Lambda}_1 \mathbf{F} + \mathbf{e}_2 \\ &= \mathbf{\Lambda}'_2 \mathbf{F}' + \mathbf{e}_2,\end{aligned}\tag{3.16}$$

⁵For $\mathbf{d}_{i,1}$, $\mathbf{d}_{i,2}$, \mathbf{F}_i , $\mathbf{e}_{i,1}$ and $\mathbf{e}_{i,2}$

Figure 3.5: Surrogate Models



where $\Lambda'_2 = \Lambda_2 \Lambda_1^{-1}$. The covariance matrix of the transformed factors is

$$\Phi' = \text{cov}(\Lambda_1 \mathbf{F}) = \Lambda_1 \Phi \Lambda_1^\top,$$

and the covariance matrices of the error terms are the same as for the centered original model. The parameters of Model One are

$$\begin{aligned} \theta_1 &= (\Lambda'_2, \quad \Phi', \quad \Omega'_1, \quad \Omega'_2, \quad \Omega'_{1,2}) \\ &= (\Lambda_2 \Lambda_1^{-1}, \quad \Lambda_1 \Phi \Lambda_1^\top, \quad \Omega_1, \quad \Omega_2, \quad \Omega_{1,2}). \end{aligned} \quad (3.17)$$

In Model Two, the factors are scaled to have variance one. Since they are already have expected value zero, this means they are standardized. Let \mathbf{V} denote a diagonal matrix with the variances of the factors (the diagonal elements of Φ) on the main diagonal. Transforming the factors by $\mathbf{F}'' = \mathbf{V}^{-1/2} \mathbf{F}$,

$$\begin{aligned} \mathbf{d}_1 &= \Lambda_1 \mathbf{F} + \mathbf{e}_1 \\ &= \Lambda_1 \mathbf{V}^{1/2} \mathbf{V}^{-1/2} \mathbf{F} + \mathbf{e}_1 \\ &= \Lambda_1'' \mathbf{F}'' + \mathbf{e}_1, \end{aligned}$$

and

$$\begin{aligned} \mathbf{d}_2 &= \Lambda_2 \mathbf{F} + \mathbf{e}_2 \\ &= \Lambda_2 \mathbf{V}^{1/2} \mathbf{V}^{-1/2} \mathbf{F} + \mathbf{e}_2 \\ &= \Lambda_2'' \mathbf{F}'' + \mathbf{e}_2. \end{aligned}$$

Summarizing, the equations for Model Two are

$$\begin{aligned} \mathbf{d}_1 &= \Lambda_1'' \mathbf{F}'' + \mathbf{e}_1 \\ \mathbf{d}_2 &= \Lambda_2'' \mathbf{F}'' + \mathbf{e}_2, \end{aligned} \quad (3.18)$$

where $\Lambda_1'' = \Lambda_1 \mathbf{V}^{1/2}$ and $\Lambda_2'' = \Lambda_2 \mathbf{V}^{1/2}$. The covariance matrix of the transformed factors for Model Two is

$$\Phi'' = \text{cov}(\mathbf{V}^{-1/2} \mathbf{F}) = \mathbf{V}^{-1/2} \Phi \mathbf{V}^{-1/2},$$

and the parameter vector is

$$\begin{aligned}\boldsymbol{\theta}_2 &= (\quad \Lambda_1'', \quad \Lambda_2'', \quad \Phi'', \quad \Omega_1'', \quad \Omega_2'', \quad \Omega_{1,2}'') \\ &= (\quad \Lambda_1 \mathbf{V}^{1/2}, \quad \Lambda_2 \mathbf{V}^{1/2}, \quad \mathbf{V}^{-1/2} \Phi \mathbf{V}^{-1/2}, \quad \Omega_1, \quad \Omega_2, \quad \Omega_{1,2}).\end{aligned}\tag{3.19}$$

The objective here is to show that $\boldsymbol{\theta}_1$ in Expression (3.17) and $\boldsymbol{\theta}_2$ in Expression (3.19) are connected by a one-to-one function; that is, $\boldsymbol{\theta}_1$ is a function of $\boldsymbol{\theta}_2$, and $\boldsymbol{\theta}_2$ is a function of $\boldsymbol{\theta}_1$.

To find $\boldsymbol{\theta}_1$ as a function of $\boldsymbol{\theta}_2$, it is enough to express Λ_2' and Φ' in terms of the elements of $\boldsymbol{\theta}_2$. We have

$$\begin{aligned}\Lambda_2'' \Lambda_1''^{-1} &= \Lambda_2 \mathbf{V}^{1/2} (\Lambda_1 \mathbf{V}^{1/2})^{-1} \\ &= \Lambda_2 \mathbf{V}^{1/2} \mathbf{V}^{-1/2} \Lambda_1^{-1} \\ &= \Lambda_2 \Lambda_1^{-1} \\ &= \Lambda_2'\end{aligned}\tag{3.20}$$

and

$$\begin{aligned}\Lambda_1'' \Phi'' \Lambda_1''^\top &= (\Lambda_1 \mathbf{V}^{1/2}) (\mathbf{V}^{-1/2} \Phi \mathbf{V}^{-1/2}) (\Lambda_1 \mathbf{V}^{1/2})^\top \\ &= \Lambda_1 \Phi \mathbf{V}^{-1/2} \mathbf{V}^{1/2} \Lambda_1^\top \\ &= \Lambda_1 \Phi \Lambda_1^\top \\ &= \Phi'.\end{aligned}\tag{3.21}$$

Notice that going in this direction, the assumption that Λ_1 is diagonal is not used. All that's necessary is the existence of an inverse. Also notice that by the invariance principle of maximum likelihood estimation, one could simply place hats on the parameter matrices of (3.20) and (3.21) to obtain estimates for Model One from those for Model Two, without re-fitting the model. Similarly, expressions (3.22), (3.23) and (3.24) below may be used to obtain Model Two estimates directly from Model One estimates.

To go from Model One to Model Two, a bit of background is required. Let \mathbf{A} and \mathbf{B} be diagonal (and square) matrices of the same size. Then $\mathbf{AB} = \mathbf{BA}$, and if all the elements are non-negative, $(\mathbf{AB})^{1/2} = \mathbf{A}^{1/2} \mathbf{B}^{1/2}$. Also, let \mathbf{A} be a square matrix, and let $dg(\mathbf{A})$ denote the diagonal matrix with diagonal elements equal to the diagonal elements of \mathbf{A} . For example, in the current problem, $dg(\Phi) = \mathbf{V}$.

Now, suppose the diagonal elements of Λ_1 are labelled $\lambda_1, \dots, \lambda_p$. Because Λ_1 is diagonal, the j th diagonal element of $\Phi' = \Lambda_1 \Phi \Lambda_1^\top$ is $\lambda_j \phi_{j,j} \lambda_j = \lambda_j^2 \phi_{j,j}$. This is also the j th diagonal element of $\Lambda_1 \mathbf{V} \Lambda_1$, which is diagonal because the product of diagonal matrices is diagonal. In short, $dg(\Phi') = \Lambda_1 \mathbf{V} \Lambda_1$.

The task is to find $\boldsymbol{\theta}_2$ as a function of $\boldsymbol{\theta}_1$. That is, we need to express Λ_1'' , Λ_2'' and Φ'' in terms of single-prime quantities. The variances and covariances in Ω_1'' , Ω_2'' and $\Omega_{1,2}''$ are automatic, because the transformations considered here do not affect these error matrices.

Using the special properties of diagonal matrices indicated above,

$$\begin{aligned}
 dg(\Phi')^{1/2} &= (\Lambda_1 \mathbf{V} \Lambda_1)^{1/2} \\
 &= (\Lambda_1 \mathbf{V}^{1/2} \mathbf{V}^{1/2} \Lambda_1)^{1/2} \\
 &= (\Lambda_1 \mathbf{V}^{1/2} \Lambda_1 \mathbf{V}^{1/2})^{1/2} \\
 &= \Lambda_1 \mathbf{V}^{1/2} \\
 &= \Lambda_1''
 \end{aligned} \tag{3.22}$$

and

$$\begin{aligned}
 \Lambda_2' dg(\Phi')^{1/2} &= (\Lambda_2 \Lambda_1^{-1}) (\Lambda_1 \mathbf{V}^{1/2}) \\
 &= \Lambda_2 \mathbf{V}^{1/2} \\
 &= \Lambda_2''
 \end{aligned} \tag{3.23}$$

and

$$\begin{aligned}
 dg(\Phi')^{-1/2} \Phi' dg(\Phi')^{-1/2} &= (\Lambda_1 \mathbf{V}^{1/2})^{-1} (\Lambda_1 \Phi \Lambda_1^\top) (\Lambda_1 \mathbf{V}^{1/2})^{-1} \\
 &= \mathbf{V}^{-1/2} \Lambda_1^{-1} \Lambda_1 \Phi \Lambda_1 \mathbf{V}^{-1/2} \Lambda_1^{-1} \\
 &= \mathbf{V}^{-1/2} \Phi \Lambda_1 \Lambda_1^{-1} \mathbf{V}^{-1/2} \\
 &= \mathbf{V}^{-1/2} \Phi \mathbf{V}^{-1/2} \\
 &= \Phi''
 \end{aligned} \tag{3.24}$$

This establishes that the parameters of Models One and Two are one to one. In Figure 3.5, there could be a two-headed arrow between Model One and Model Two⁶. As a corollary, we have the following useful rule.

Rule 2i: The Equivalence Rule For a centered factor analysis model with at least one reference variable for each factor, suppose that surrogate models are obtained by either standardizing the factors, or by setting the factor loading of a reference variable equal to one for each factor. Then the parameters of one surrogate model are identifiable if and only if the parameters of the other surrogate model are identifiable.

⁶The student may be like, Okay, this is all correct, but how would anyone even think of some of these functions, especially the formula for Φ'' in (3.24)? The key is that surprisingly, if you standardize \mathbf{F}' you get \mathbf{F}'' . This makes it easy to write the double-prime matrices in terms of the single-prime matrices. Going in the other direction, try a change of variables in which Λ_1'' is absorbed into \mathbf{F}'' , effectively setting a factor loading to one for each factor. The change of variables is $\mathbf{F}''' = \Lambda_1'' \mathbf{F}''$, which happens to equal \mathbf{F}' .

This is so remarkable that it bears repeating. If you set a factor loading to one for each factor in the centered original model, you get Model One. If you standardize the factors in the centered original model, you get Model Two. If you standardize the factors in Model One, you get Model Two. If you set a factor loading to one for each factor in Model Two, you get Model One. It feels like a projection of some kind.

3.3.2 Choosing a Surrogate Model

So, the two standard surrogate models are equivalent. Their identifiability status is the same, and they contain the same information about the parameters of the original model. In actual data analysis, which one should you use?

An advantage of unit factor loadings Certainly, when factor loadings are set to one it's easier to calculate Σ by hand as a function of the surrogate model parameters. It's also easier to solve for model parameters in terms of σ_{ij} quantities – again, if the calculations are done by hand. For models to which the standard identifiability rules do not apply, this can be very helpful.

An advantage of standardized factors Recall that the identifiable parameters of a surrogate model are actually identifiable *functions* of the parameters of the original model. It's helpful if these functions correspond directly to something you want to know about. When a factor loading is seemingly set to one, the other factor loadings (which would appear on the other arrows coming from that factor) are actually *ratios* of factor loadings, with the invisible factor loading in the denominator. Thus, all the other factor loadings are relative to the one that disappeared, making the invisible factor loading a kind of reference quantity. When such ratios of factor loadings are of interest, setting a factor loading to one for each factor is a good choice.

On the other hand, the variances and covariances of the factors under the surrogate model are the original quantities multiplied by the reference factor loadings. This preserves nothing of interest from the original model, apart from the signs of the covariances.

In contrast, consider the covariances between factors when the factors are standardized. For a general factor analysis model, suppose that the factor F_j has been standardized. Using double primes for consistency with the notation of Section 3.3.1, let $F_j'' = \frac{1}{\sqrt{\phi_{j,j}}}F_j$. As well, the factor F_ℓ has been standardized by $F_\ell'' = \frac{1}{\sqrt{\phi_{\ell,\ell}}}F_\ell$. Then the covariance between the transformed factors is

$$\begin{aligned}\phi_{j,\ell}'' &= \text{cov}(F_j'', F_\ell'') \\ &= \text{cov}\left(\frac{1}{\sqrt{\phi_{j,j}}}F_j, \frac{1}{\sqrt{\phi_{\ell,\ell}}}F_\ell\right) \\ &= \frac{1}{\sqrt{\phi_{j,j}\phi_{\ell,\ell}}}\text{cov}(F_j, F_\ell) \\ &= \frac{\phi_{j,\ell}}{\sqrt{\phi_{j,j}\phi_{\ell,\ell}}} \\ &= \text{Corr}(F_j, F_\ell).\end{aligned}$$

The covariances between factors under the surrogate model are not just correlations, which they must be since the factors have variance one. They are *the* correlations — that is, they are exactly the correlations between factors under the original model. As such, those $\phi_{j,\ell}''$ quantities are very easy to understand and interpret. Confidence intervals are

meaningful. This is a significant advantage to standardizing the factors, though it's more helpful for pure factor analysis than for general structural equation models with a causal structure in the latent variables.

While it might be tempting to set a factor loading to one for a factor and also standardize that same factor, it's a very bad idea. You can do it, but this reduction of the original parameter space cannot be accomplished by a change of variables. Consequently the connection of the resulting model parameters to the parameters of the original model would be mysterious. Furthermore, doing both at once usually implies equality constraints on Σ that do not follow from the original model, invalidating the goodness of fit test. It's something you just should not do.

To summarize, setting a factor loading to one for each factor (Model One) is attractive because it makes calculations easier. Standardizing factors (Model Two) is attractive because the resulting covariances between factors are the correlations between factors under the original model. As the following example shows, it is possible to enjoy the benefits of both surrogate models. If identifiability is unclear and you prefer the interpretability of the model with standardized factors, you can safely show identifiability for Model One and then Fit Model Two to the data.

Example 3.3.1 *The Political Democracy Example*

This data set is discussed by Bollen [10] and other authors. Based on news reports and other sources, a panel of experts rated a sample of 72 developing countries on the following variables.

d_1 : Freedom of the press in 1960

d_2 : Freedom of political opposition in 1960

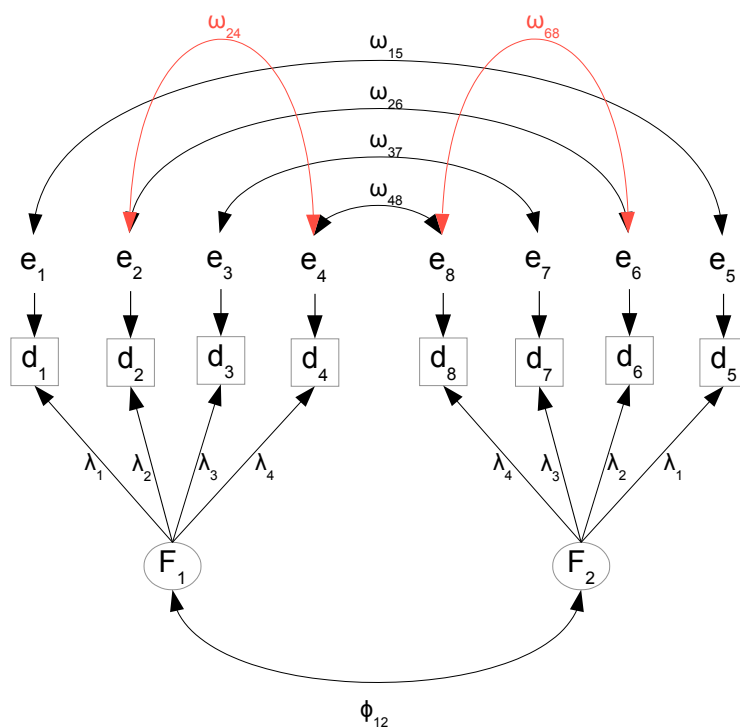
d_3 : Fairness of elections in 1960

d_4 : Effectiveness of the elected legislature in 1960

The variables d_5 through d_8 represent the same quantities for the year 1965. There are two hypothesized factors, strength of political democracy in 1960, and strength of political democracy in 1965. Figure 3.6 shows a path diagram, which in my humble opinion is an improvement on Bollen's Figure 7.3 on page 235 — even though they contain the same information. The factor F_1 is political democracy in 1960, and F_2 is political democracy in 1965. The factor loadings are hypothesized to be the same in 1960 and 1965, though the variances of the error terms might not be. Though the variables d_5 through d_8 correspond directly to d_1 through d_4 , they are sorted in the opposite order to allow for the curved arrows between arrow terms.

It is those curved arrows, representing covariances between error terms, that make this model unusual. There are *two* sources of covariance between the 1960 and 1965 variables, only one of which the covariance between factors. Even so, it turns out that all the parameters are identifiable.

Figure 3.6: Political Democracy Factor Model



The curved arrows between error terms reflect really thoughtful, good modelling. For example, freedom of the press in 1960 and freedom of the press in 1965 are assessed based on similar information from mostly the same sources, so that both observed variables are impacted by similar sources of bias. The latent variables involved are not part of the model, so they are represented by a covariance between error terms. The same applies to the other three pairs of variables (d_2 and d_6 , d_3 and d_7 , d_4 and d_8). Covariances within years are in red just for visual contrast. The measurement of d_2 and d_4 have something extra in common, as do d_6 and d_8 . I'm not sure what it is. Anyway, this is good. Most people just assume error terms uncorrelated without really thinking about it.

We are interested in a surrogate model with standardized factors, and we need to verify identifiability before trying to fit the model. Identifiability will be a lot easier to check for a surrogate model with $\lambda_1 = 1$. The [equivalence rule](#) says that it's okay to check one model and then feel comfortable fitting the other one.

Without the curved arrows, this model would be identifiable at a glance by the three-variable rule. With the curved arrows, it will be possible to get the job done by combining rules and a few simple calculations. Bear in mind that once a parameter has been identified, it may be used in the solutions for other parameters.

Here are the model equations for the model with factor loadings set to one.

$$\begin{aligned}
 d_1 &= F_1 + e_1 \\
 d_2 &= \lambda_2 F_1 + e_2 \\
 d_3 &= \lambda_3 F_1 + e_3 \\
 d_4 &= \lambda_4 F_1 + e_4 \\
 d_5 &= F_2 + e_5 \\
 d_6 &= \lambda_2 F_2 + e_6 \\
 d_7 &= \lambda_3 F_2 + e_7 \\
 d_8 &= \lambda_4 F_2 + e_8
 \end{aligned}$$

First, apply the three-variable rule to the submodel with F_1 , d_1 , d_2 and d_3 ⁷. The parameters λ_2 , λ_3 , ϕ_{11} , ω_{11} , ω_{22} and ω_{33} are identified.

Adding d_4 to the system is non-standard, because of the covariance between e_4 and e_2 . However,

$$\begin{aligned}
 \sigma_{14} &= \text{cov}(d_1, d_4) \\
 &= \text{cov}(F_1 + e_1)(\lambda_4 F_1 + e_4) \\
 &= \lambda_4 \text{Var}(F_1) + 0 + 0 + 0 \\
 &= \lambda_4 \phi_{11},
 \end{aligned}$$

and $\lambda_4 = \sigma_{14}/\phi_{11}$ is identified. Then,

$$\begin{aligned}
 \sigma_{24} &= \text{cov}(d_2, d_4) \\
 &= \text{cov}(\lambda_2 F_1 + e_2)(\lambda_4 F_1 + e_4) \\
 &= \lambda_2 \lambda_4 \text{Var}(F_1) + 0 + 0 + \text{cov}(e_2, e_4) \\
 &= \lambda_2 \lambda_4 \phi_{11} + \omega_{24},
 \end{aligned}$$

and $\omega_{24} = \sigma_{24} - \lambda_2 \lambda_4 \phi_{11}$ is identified.

Repeating these operations for the submodel with F_2 , d_5 , d_6 , d_7 and d_8 , the variance parameters $\omega_{55}, \dots, \omega_{88}$ are identified. Also, it is clear that if the factor loadings for 1965 were different from 1960, they would be identified as well.

Now we turn to the sources of covariance between the 1960 and 1965 measurements.

$$\begin{aligned}
 \sigma_{18} &= \text{cov}(d_1, d_8) \\
 &= \text{cov}(F_1 + e_1)(\lambda_4 F_2 + e_8) \\
 &= \lambda_4 \text{cov}(F_1, F_2) \\
 &= \lambda_4 \phi_{12}.
 \end{aligned}$$

Then, $\phi_{12} = \sigma_{18}/\lambda_4$ is identified.

⁷What about the curved arrows? There are no curved arrows connecting e_1 , e_2 and e_3 , so the calculations for this subsystem, if we had to re-do them, would be unaffected.

Now it's straightforward to solve for the remaining covariances between errors.

$$\begin{aligned}
 \sigma_{15} &= \text{cov}(d_1, d_5) \\
 &= \text{cov}(F_1 + e_1)(F_2 + e_5) \\
 &= \text{cov}(F_1, F_2) + \text{cov}(e_1, e_5) \\
 &= \phi_{12} + \omega_{15} \\
 \implies \omega_{15} &= \sigma_{15} - \phi_{12},
 \end{aligned}$$

and

$$\begin{aligned}
 \sigma_{26} &= \text{cov}(d_2, d_6) \\
 &= \text{cov}(\lambda_2 F_1 + e_2)(\lambda_2 F_2 + e_6) \\
 &= \lambda_2^2 \text{cov}(F_1, F_2) + \text{cov}(e_2, e_6) \\
 &= \lambda_2^2 \phi_{12} + \omega_{26} \\
 \implies \omega_{26} &= \sigma_{26} - \lambda_2^2 \phi_{12},
 \end{aligned}$$

and similarly for ω_{37} and ω_{48} .

I got a bit carried away here, and showed elementary details that you are probably able to do in your head. This may obscure the fact that establishing identifiability for this interesting model is really pretty easy, especially when working with the surrogate model in which factor loadings are set to one. It's not necessary to calculate the whole covariance matrix Σ , and all the calculations that are really needed could be done on a sheet of scratch paper.

3.4 The Reference Variable rule

This rule comes from applying the [equivalence rule](#) to the reference variable rule for unstandardized factors, on page 286, so that it holds for both the common surrogate factor analysis models. It says that under some other conditions that are fairly mild and easy to satisfy, the parameters of a model with three observable variables per factor will be identifiable, provided that one of the variables is a reference variable. The other two variables may be influenced by all the factors. Here is the rule.

Rule 2c: The Reference Variable Rule The parameters of a factor analysis model are identifiable except possibly on a set of volume zero in the parameter space, provided

- The number of observable variables (including reference variables) is at least three times the number of factors.
- There is at least one reference variable for each factor.
- For each factor, either the variance equals one and the sign of the reference variable's factor loading is known, or the factor loading of the reference variable is equal to one.
- Divide the observable variables into sets. The first set contains one reference variable for each factor. The number of variables in the second set and the number in the third set is also equal to the number of factors. The fourth set may contain any number of additional variables, including zero. The error terms for the variables in the first three sets may have non-zero covariance within sets, but not between sets. The error terms for the variables in the fourth set may have non-zero covariance within the set, and with the error terms of sets two and three, but they must have zero covariance with the error terms of the reference variables.

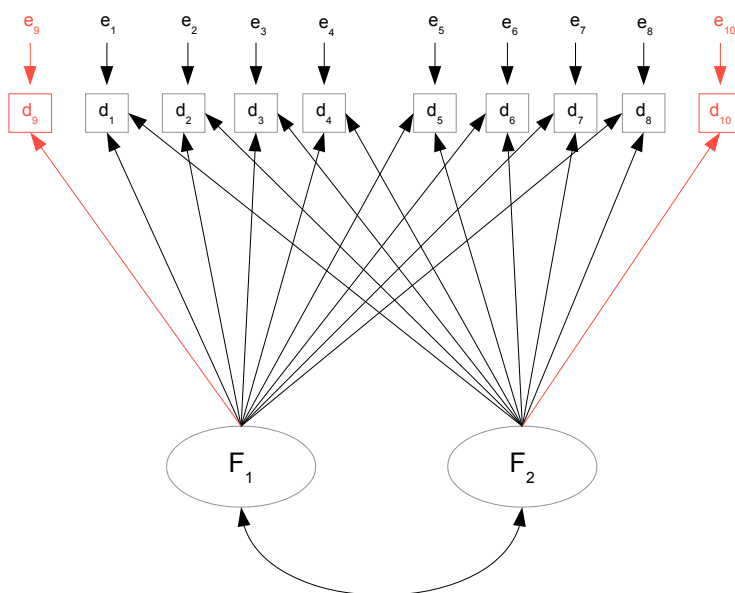
The last condition is unusually long. It describes patterns of permissible covariances between error terms. That's important and we will get back to it, but for now just observe that the condition is satisfied for models in which all the error terms are independent – something that is almost the default for factor analysis models⁸.

The rule with independent errors Figure 3.7 illustrates the reference variable rule with independent errors, and also gives an idea of the modelling flexibility the rule permits. The black part of the model is a direct copy of the unrestricted exploratory factor analysis model of Figure 2.1 in Chapter 2. Then, reference variables for the factors (the observable variables d_9 and d_{10}) have been added in red. The resulting model is immediately identifiable, assuming the factors are standardized or the factor loadings on the red arrows are set to one.

This shows that with a few extra variables of the right kind, the parameters of an exploratory factor analysis can be estimated without any fuss. If the factors are standardized, the covariances between factors are the correlations between factors under the original model. The factor loadings under the surrogate model are positive multiples of the the corresponding factor loadings under the original model. While the actual values of the original factor loadings are not knowable, it is possible to estimate and test whether their signs are positive, negative or zero. That's enough for many purposes. All the technical gymnastics from Chapter 2, like rotation to simple structure, viewing the resulting factor solution as a scientific theory and invoking Occam's Razor from the philosophy of

⁸Independent errors are universal in exploratory factor analysis, and many confirmatory factor analysis models seem to have inherited this feature. In Chapter 2 on page 214, independent errors are traced to Spearman's (1904) "[Law of the Universal Unity of the Intellectual Function.](#)" [60]

Figure 3.7: Adding reference variables to an unrestricted factor model



science to justify it on the grounds of simplicity — all of that is unnecessary if you have the right kind of data set. The reference variable rule tells you what kind of data set you need.

There is a general point here. Lack of identifiability is often a problem with the study design, not the model. This makes sense. Identifiability is literally about what can be known. Naturally, there is an intimate connection to research design.

One other observation is that while the black part of Figure 3.7 is an exploratory factor analysis model, the whole analysis can't be completely exploratory. You really need to have a good idea of what the factors are before designing measurement procedures (reference variables) that clearly tap one factor but not any of the others.

Statement of the model Rule 2c goes on and on about covariances between error terms. To clarify the discussion, a full statement of the model will be helpful. This is an adaptation of Model 3.1 on page 286. Independently for $i = 1, \dots, n$,

$$\begin{aligned}
 \mathbf{d}_{i,1} &= \Lambda_1 \mathbf{F}_i + \mathbf{e}_{i,1} \\
 \mathbf{d}_{i,2} &= \Lambda_2 \mathbf{F}_i + \mathbf{e}_{i,2} \\
 \mathbf{d}_{i,3} &= \Lambda_3 \mathbf{F}_i + \mathbf{e}_{i,3} \\
 \mathbf{d}_{i,4} &= \Lambda_4 \mathbf{F}_i + \mathbf{e}_{i,4},
 \end{aligned} \tag{3.25}$$

where

- $\mathbf{d}_{i,1}$, $\mathbf{d}_{i,2}$ and $\mathbf{d}_{i,3}$ are $p \times 1$ observable random vectors.

- $\mathbf{d}_{i,4}$ need not be present. If it is present, it is an $m \times 1$ observable random vector.
- \mathbf{F}_i (F for Factor) is a $p \times 1$ latent random vector with expected value zero $cov(\mathbf{F}_i) = \mathbf{\Phi}$.
- $\mathbf{\Lambda}_1$ is a $p \times p$ diagonal of constants, with non-zero diagonal elements. The diagonal elements may be assumed positive.
- $\mathbf{\Lambda}_2$ and $\mathbf{\Lambda}_3$ are $p \times p$ non-singular matrices of constants.
- $\mathbf{\Lambda}_4$, if it is present, is an m by p matrix of constants.
- $\mathbf{e}_{i,1}, \dots, \mathbf{e}_{i,4}$ are vectors of error terms, with expected value zero, covariance matrix $cov(\mathbf{e}_{i,j}) = \mathbf{\Omega}_{j,j}$ for $j = 1, \dots, 4$, and
 - $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) = cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,3}) = cov(\mathbf{e}_{i,2}, \mathbf{e}_{i,3}) = \mathbf{O}$, all $p \times p$ matrices.
 - $cov(\mathbf{e}_{i,1}, \mathbf{e}_{i,4}) = \mathbf{O}$, a $p \times m$ matrix.
 - $cov(\mathbf{e}_{i,2}, \mathbf{e}_{i,4}) = \mathbf{\Omega}_{2,4}$ and $cov(\mathbf{e}_{i,3}, \mathbf{e}_{i,4}) = \mathbf{\Omega}_{3,4}$.
- Either the diagonal elements of $\mathbf{\Lambda}_1$ or the diagonal elements of $\mathbf{\Phi}$ are equal to one.

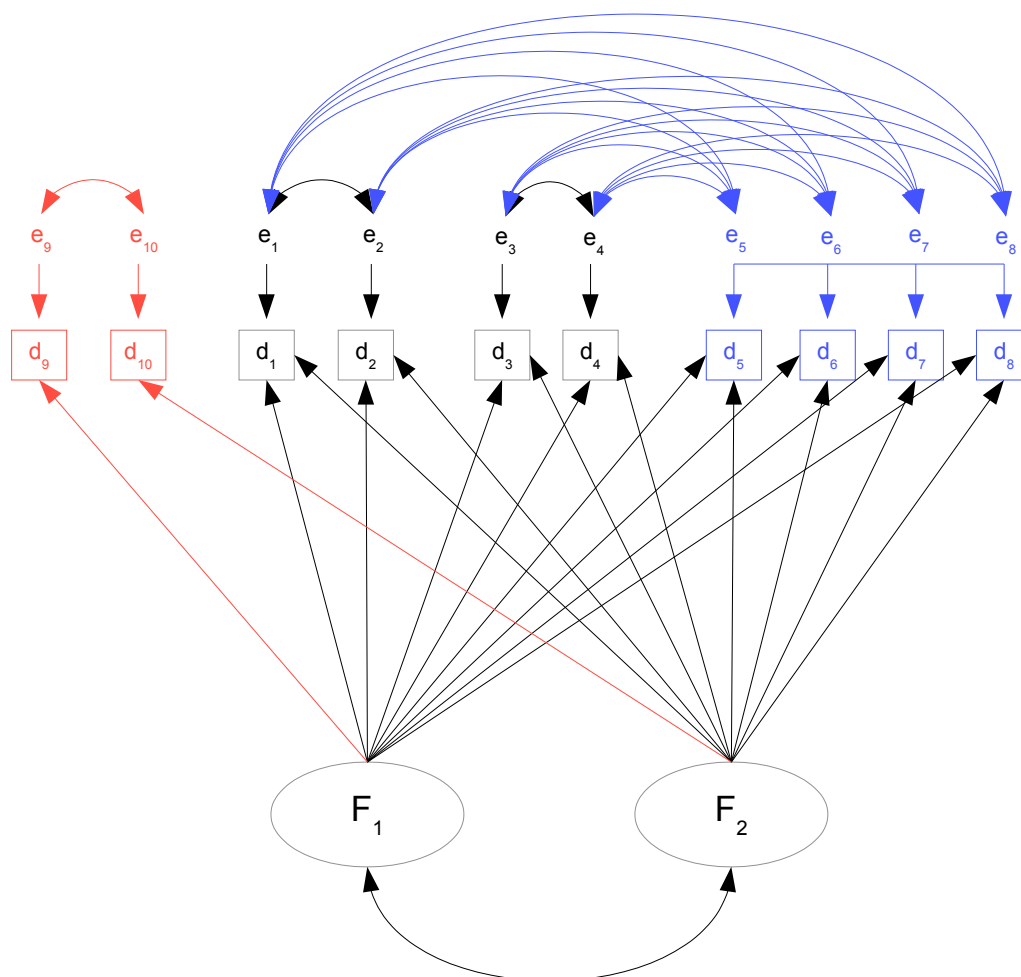
What's happening here is that the reference variables for the factors are being placed in $\mathbf{d}_{i,1}$, and then the remaining observable variables are being allocated to $\mathbf{d}_{i,2}$, $\mathbf{d}_{i,3}$, and possibly $\mathbf{d}_{i,4}$, depending on the potential for non-zero covariance between their error terms.

Figure 3.8 is a re-arranged version of Figure 3.7, showing the covariances between errors that the rule allows. The reference variables d_9 and d_{10} are grouped together in $\mathbf{d}_{i,1}$, while $\mathbf{d}_{i,2}$ contains d_1 and d_2 , and $\mathbf{d}_{i,3}$ contains d_3 and d_4 . The remaining observed variables, d_5 through d_8 , are placed in $\mathbf{d}_{i,4}$. With the colour coding, perhaps you can see it. e_9 and e_{10} are correlated, e_1 and e_2 are correlated, e_3 and e_4 are correlated, e_5 through e_8 are correlated, and there are four blue connectors running to each of e_1 , e_2 , e_3 and e_4 .

Correlated error terms To understand how error terms might be correlated, consider what an error term represents. In a path diagram, suppose that a variable y has three arrows pointing toward it from x_1 , x_2 and x_3 , and one more arrow coming from e , an error term. The model is saying that y is influenced by the x variables, but it's not completely determined by them. There are other, unmeasured variables that affect y . We don't know what they all are, or even how many there are. Anyway, we roll them together and call them e . That is, the error term in a model equation is *everything else* that affects the endogenous variable, apart from the other variables on the right side of the equation.

Thinking of an error term as a giant linear combination of unmeasured and perhaps even unimagined variables (probably not a bad approximation), it is clear that if any variables appear in more than one such linear combination, or if some of the variables in two different linear combinations have non-zero covariance, then the error terms will have non-zero covariance as well. This is how the curved arrows between error terms arise. "Everything else" includes some of the same influences, or related influences.

Figure 3.8: Allowable covariances between error terms



When observable variables are recorded at roughly the same time and by the same method, then correlated errors of measurement are practically unavoidable. For example, suppose that a sample of high school students takes a standardized test, consisting of sub-tests on mathematical and verbal material. Scores on the sub-tests will be two different observable variables. Some students will suffer from test anxiety more than others, some will be more test-wise than others, some will have gotten more sleep the night before, and some students will simply be having a better day than others. The list goes on. The point is that these unmeasured factors are not explicitly part of the model, but they will influence performance on both the math test and the verbal test. They are a source of covariance between the two measures, over and above any covariance between the *factors* (say, verbal ability and mathematical ability) that the tests seek to measure. All this would be represented by a curved, double-headed arrow between the error terms.

If the variables in a study come from questionnaires, the case for correlated error terms

is even stronger. Consider a questionnaire with a lot of questions about the respondent's workplace. Mixed together are questions from several sub-scales that seek to assess the quality of relations with co-workers, the perceived overall fairness of management, opportunities for advancement, and the respondent's job satisfaction. In the model, these sub-scales are going to be separate observed variables, each with its own error term. The respondent's current mood will certainly affect all the responses, as may happy or unhappy events outside the workplace. Some respondents will not really believe their responses will not get back to the employer, and will play it safe by saying that everything is great — on *all* the questions. Others will take the opportunity to vent their frustrations, and paint a picture of everything that is darker than the one they actually experience from day to day.

Also, one should not minimize the extent to which social science research (including market research and behavioural economics) is a social transaction between the participant and the investigator. Many people answering questionnaires certainly seek to represent themselves in a favourable light [19],[52] and often politely tell the investigator what they think the investigator wants to hear [49]. All these dynamics (which are only rarely what the investigator wants to study) push the responses to clusters of questions up or down together. In the path diagram they are represented by curved, double-headed arrows connecting error terms.

It would be nice if all error terms could have covariances with one another that are unknown parameters, and not assumed zero. This is how it goes in ordinary multivariate regression, with all variables observable. Once there are latent variables, however, identifiability becomes an issue. Certainly, if all the error terms in a factor analysis model have non-zero covariance with each other, then the [parameter count rule](#) establishes that all the parameters of the model cannot be identifiable. So, what should we do?

One alternative is to assume the covariances are zero, and hope. Just hope that the processes involving the variables in the model are a lot stronger than the processes leading to correlated error terms. The model is not quite correct and everyone knows it, but it should not be too misleading. I think it's fair to say that almost all the usual factor analysis models with independent error terms are based on this kind of hope. Too often, the model does not fit; this can include negative variance estimates, the so-called [Heywood case](#) described on page 226. Note that the negative variance in Example 1.5.1 was produced by correlated error terms.

There is another, better solution: careful research design. This means doing some thinking about the model to *before* collecting the data. The first thing to note is that some error terms can legitimately be assumed to have zero covariance – on the basis of reasonable modelling, not just hope. For example, suppose that a medical technician records the height of a patient, and also asks about occupation (later to be converted into a numerical index of occupational prestige). There is surely measurement error in both operations, but no particular reason to suspect that the errors might vary systematically with one another. Again, suppose a participant in a study fills out several questionnaires designed to assess racism and other social attitudes. The error terms are correlated, without a doubt. But if the person also grants access to her cell phone data, then a racism measure derived from Facebook likes (again imperfect, as always) could arguably

have an error term independent of the error terms of the self-report data.

As another example, here's a quote from page 84 in Section 0.10.3 on the double measurement design in Chapter 0: "...farmers who overestimate their number of pigs may also overestimate their number of cows. On the other hand, if the number of pigs is counted once by the farm manager at feeding time and on another occasion by a research assistant from an areal photograph, then it would be fair to assume that the errors of measurement for the different methods are uncorrelated." There are more examples in the BMI Health Study (Section 0.10.4 of Chapter 0, page 89). The point is that error terms need not *always* be correlated. If two observable variables are measured by different methods, on different occasions and ideally by different personnel, it's usually reasonable to assume that their errors are independent.

This is where the reference variable rule comes in. Like the [double measurement rule](#), it allows correlated errors *within* certain sets of observed variables, as long as there is zero covariance *between* sets — and identifiability is still preserved. It requires advance planning, and the data collection will inevitably be more demanding. However, it's not really a lot to ask. In experimental research (with random assignment of cases to treatment conditions), it is quite common to plan the data collection and statistical analysis at the same time, and to take a lot of care about the details of procedure. The same thing applies to good research using strictly observational data. It's not enough to just hand out a bunch of questionnaires.

Example 3.4.1 *Student Mental Health*

Let's give some content to Figure 3.8. The result will be a re-arrangement of the observed variables, with some of the curved, double-headed arrows eliminated. Suppose it's a study of student mental health. The investigators believe that anxiety and depression are the two main mental health problems that many young people face. They mean long-lasting, chronic anxiety and depression, not just getting anxious or sad about something, and then the feeling passes. The investigators are interested in how these traits are related to one another. Specifically, they want to estimate the correlation between true (not just reported) long-term anxiety and true long-term depression.

The participants are volunteer High School students. They all take part in a one-on-one interview with a clinical psychologist, who asks some very carefully chosen questions, and assesses them on level of persisting anxiety and level of persisting depression. I am willing to believe that the anxiety assessment reflects true anxiety plus error, and is not directly influenced by true depression. I also can believe that the depression assessment reflects true depression plus error, and is not directly influenced by true anxiety. So both clinical assessments are reference variables.

Regardless of what the clinical psychologist might claim, it's unavoidable that common extraneous factors will affect both assessments. For example, regardless of how skilled and non-threatening the psychologist might be, some people will just be less likely to report symptoms; it's a matter of personal style. The measurement errors of the two clinical

assessments are correlated, but we can live with it. The variables in the first set are:

d_9 : Clinical rating of anxiety.

d_{10} : Clinical rating of depression.

Using security camera recordings of students eating lunch in the cafeteria (with everyone's permission, of course), the investigators record four social behaviour variables during a designated twenty-minute period. Correlated errors within this set are very likely.

d_1 : Speaking time (not on phone).

d_2 : Listening time (head turned toward speaker).

d_5 : Number of smiles/laughs while not on cell phone solo⁹.

d_6 : Time solo on cell phone.

The following variables are obtained from school records. Measurement errors may not be correlated within this set, but we will be conservative, and assume they might be. In any case, it will be testable.

d_3 : Grade point average last academic session.

d_4 : Attendance last academic session.

d_7 : Hours per week playing school sports.

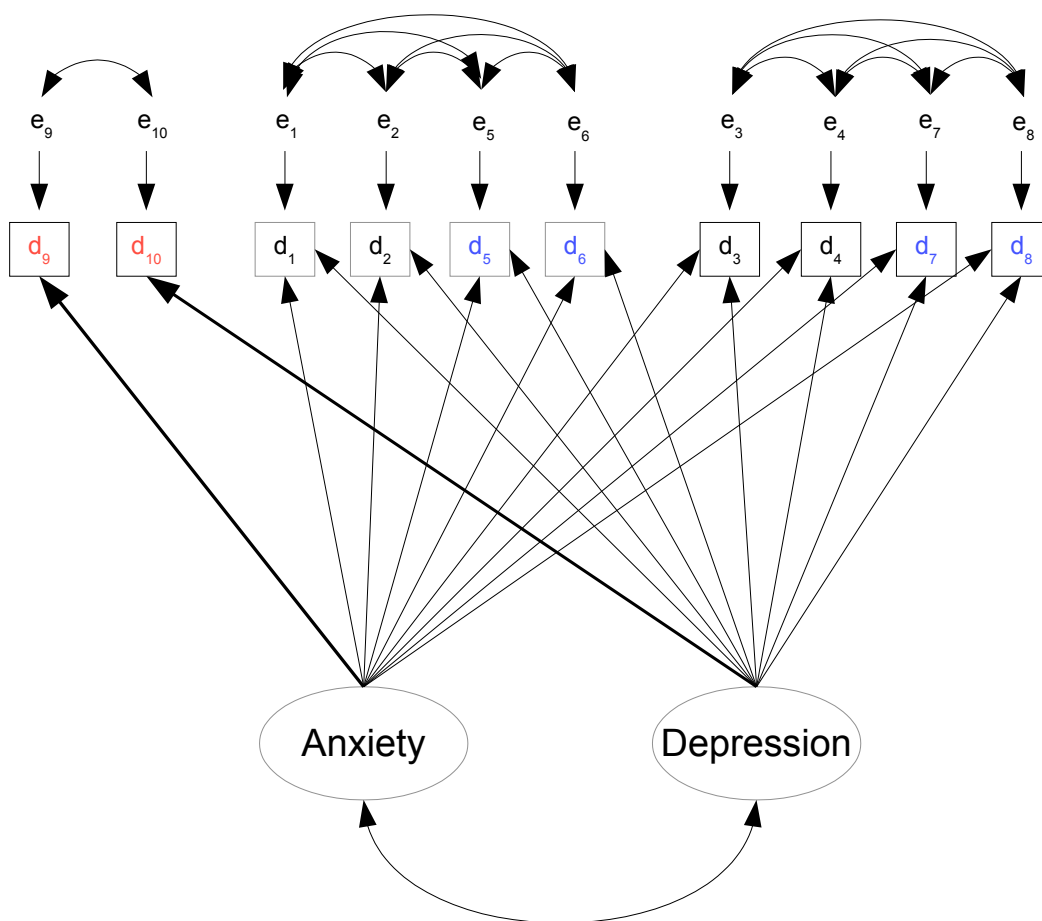
d_8 : Hours per week spent on extra-curricular activities, not including school sports.

Comparing the variable numbering and colour coding to Figure 3.8, it can be seen that two blue variables (d_5 and d_6) have been grouped with the social behaviour variables, and the other two blue ones (d_7 and d_8) have been grouped with the school record variables. The flexibility of the reference variable rule has been exploited to assemble a model that makes substantive sense, and still has identifiable parameters because it's a special case of what's allowed. The result is the model of Figure 3.9. This is a good way to apply the reference variable rule in practice. The proof requires three sets of observed variables, each with as many observed variables as there are factors, and it allows an additional set with as many variables as you like. But in practice, one may have an arbitrary number of variable sets, each with error terms correlated only within the set — provided the following conditions are met.

- One set consists of a reference variable for each factor.
- Two or more of the other sets of variables have at least as many variables as there are factors.

⁹If two people are looking at a phone together, it's not "solo," and if they smile or laugh it would be counted

Figure 3.9: Model for the student mental health example (Example 3.4)



Again, the sets of observed variables are defined by having error terms that are correlated with one another, and uncorrelated with the error terms of variables in the other sets. The uncorrelated error terms are to be justified by specific features of the research design. This is both an opportunity and an obligation.

The reference variable rule is much stronger than the three-variable rules (also called three-indicator rules) given in other textbooks I have seen. For example, in Bollen’s classic text [10] the “three-indicator” rule on p. 244 is exactly our [three-variable rule](#) (Rule 2b). All the observed variables are reference variables, and the covariance matrix of the error terms is diagonal. The result is a very restrictive model like the one in Figure 3.4, where observable variables can be influenced by only one factor. Surely it is better to *hypothesize* that certain factor loadings are zero and then test the hypothesis, than to simply assume that they are zero. Of course the assumption of independent errors is hard to justify as well, for most data sets. The reference variable rule is a welcome alternative.

Overfitting There is potential for abuse here. Suppose that as usual, data are collected without much thought to the confirmatory factor analysis model that will be fit. The error terms all could be correlated; who knows? All the factors could potentially affect all the observable variables; who knows? So the data analyst (who knows about the reference variable rule) picks some variables to be reference variables for the factors, assumes all the error terms to be independent, and runs the software. The model does not fit. So he picks some different variables as reference variables, and also semi-arbitrarily groups the observable variables into clusters, allowing non-zero covariance between error terms within a cluster. Now the fit is a lot better. The chi-squared test for lack of fit might even be non-significant. If it is still significant and the investigator keeps trying different combinations, then sooner or later, one of the models will almost surely fit the data. It is sort of like ordinary p-hacking¹⁰ in reverse. The data analyst keeps trying different things until the result is *not* statistically significant.

Has something real been discovered, or is it just an exploitation of random features of the data? The boundary between data snooping and legitimate exploratory data analysis is often fuzzy, and this is no exception. The solution, if you engage in this kind of practice, is replication and cross-validation. An example will be given in Section 3.8.

3.5 More Identification Rules

Combining the two-variable rule for unstandardized factors (page 288) with the [equivalence rule](#) yields

Rule 2d: Two-variable Rule The parameters of a factor analysis model are identifiable provided

- There are at least two factors.
- There are at least two reference variables for each factor.
- For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading of at least one reference variable is equal to one.
- Each factor has a non-zero covariance with at least one other factor.
- Errors are independent of one another and of the factors.

The two-variable rule requires at least two factors, each with two reference variables. In practice, factors that influence only two observable variables are often part of a larger system, and there might be only one such factor in the model. The following shows how a factor with two reference variables can be combined with a model whose parameters have already been identified in some other way.

¹⁰Simonsohn et al. [59] deserve credit for the catchy term “p-hacking. I do not necessarily endorse their work on the “p-curve.”

Rule 2e: Two-variable Addition Rule A factor with just two reference variables may be added to a measurement model whose parameters are identifiable, and the parameters of the combined model will be identifiable provided

- The errors for the two additional reference variables are independent of one another and of the error terms already in the model.
- For each factor, either the variance equals one and the sign of one factor loading is known, or the factor loading of at least one reference variable is equal to one.
- In the existing model with identifiable parameters,
 - There is at least one reference variable for each factor, and
 - At least one factor has a non-zero covariance with the new factor.

The proof of this rule will be given for standardized factors; the [equivalence rule](#) says that it also applies when a factor loading is set to one for each factor. Assume that there are already p factors and k observable variables in the model. The additional factor is F_{p+1} , and its reference variables are d_{k+1} and d_{k+2} .

In the existing model, there is a factor that has non-zero covariance with F_{p+1} . Without loss of generality, label this factor F_1 , and let its reference variable be d_1 . We have

$$\begin{aligned} d_1 &= \lambda_1 F_1 + e_1 \\ d_{k+1} &= \lambda_{k+1} F_{p+1} + e_{k+1} \\ d_{k+2} &= \lambda_{k+2} F_{p+1} + e_{k+2}. \end{aligned} \tag{3.26}$$

The new parameters that need to be identified are λ_{k+1} , λ_{k+2} , ω_{k+1} , ω_{k+2} , and the covariances between the existing factors and the new factor: $\phi_{j,p+1}$ for $j = 1, \dots, p$.

The covariance matrix of $\begin{pmatrix} d_1 \\ d_{k+1} \\ d_{k+2} \end{pmatrix}$ is

$$\begin{pmatrix} \sigma_{1,1} & \sigma_{1,k+1} & \sigma_{1,k+2} \\ & \sigma_{k+1,k+1} & \sigma_{k+1,k+2} \\ & & \sigma_{k+2,k+2} \end{pmatrix} = \begin{pmatrix} \lambda_1^2 + \omega_1 & \lambda_1 \lambda_{k+1} \phi_{1,p+1} & \lambda_1 \lambda_{k+2} \phi_{1,p+1} \\ & \lambda_{k+1}^2 + \omega_{k+1} & \lambda_{k+1} \lambda_{k+2} \\ & & \lambda_{k+2}^2 + \omega_{k+2} \end{pmatrix}.$$

Since the signs of λ_1 and λ_{k+1} are known, the sign of $\phi_{1,p+1}$ can be determined from $\sigma_{1,k+1}$. Also, note that since λ_1 is already identified, it may be used along with the $\sigma_{i,j}$ to solve for new parameters. Then,

$$\frac{\sigma_{1,k+1} \sigma_{1,k+2}}{\sigma_{k+1,k+2}} = \frac{\lambda_1^2 \lambda_{k+1} \lambda_{k+2} \phi_{1,p+1}^2}{\lambda_{k+1} \lambda_{k+2}} = \lambda_1^2 \phi_{1,p+1}^2.$$

Assuming λ_1 and λ_{k+1} are positive (which they can always be, by naming the factors appropriately), $\phi_{1,p+1} = \text{sign}(\sigma_{1,k+1}) \sqrt{\frac{\sigma_{1,k+1} \sigma_{1,k+2}}{\lambda_1^2 \sigma_{k+1,k+2}}}$. Since $\phi_{1,p+1}$ is now identified, it can be

used to solve for other parameters, and

$$\begin{aligned}\lambda_{k+1} &= \frac{\sigma_{1,k+1}}{\lambda_1 \phi_{1,p+1}} \\ \lambda_{k+2} &= \frac{\sigma_{1,k+2}}{\lambda_1 \phi_{1,p+1}} \\ \omega_{k+1} &= \sigma_{k+1,k+1} - \lambda_{k+1}^2 \\ \omega_{k+2} &= \sigma_{k+2,k+2} - \lambda_{k+2}^2.\end{aligned}$$

To identify the covariances of the other factors with F_{p+1} , place the primary reference variables for those factors into positions $2, \dots, p$ of the covariance matrix of observable variables. Then, for $j = 2, \dots, p$,

$$\text{cov}(d_j, d_{k+1}) = \sigma_{j,k+1} = \lambda_j \lambda_{k+1} \phi_{j,p+1} \implies \phi_{j,p+1} = \frac{\sigma_{j,k+1}}{\lambda_j \lambda_{k+1}}.$$

This establishes the two-variable addition rule.

The Combination Rule The two-variable addition rule reflects how parameter identifiability is usually established in practice for big measurement models. Parts of the model are identified, and then they are combined with other factors and variables to produce larger sub-models whose parameters are identifiable. Then the sub-models are combined. The combination rule says that sub-models with identifiable parameters may be combined, provided that the error terms of the two models have zero covariance.

Rule 2f: Combination Rule Suppose that two factor analysis models are based on non-overlapping sets of observable variables from the same data set, and that the parameters of both models are identifiable. The two models may be combined into a single model provided that the error terms of the first model are independent of the error terms in the second model. The additional parameters of the combined model are the covariances between the two sets of factors. These are all identifiable, except possibly on a set of volume zero in the parameter space.

Proof. Let the first model have p_1 factors and k_1 observable variables, and let the second model have p_2 factors and k_2 observable variables. Separate the first set of observable variables into two subsets, with p_1 variables in the first subset, and $k_1 - p_1$ variables in the other subset. Do the same thing for the other model. The criteria for separating the variables into subsets will be described presently. The model equations are now

$$\begin{aligned}\mathbf{d}_1 &= \mathbf{\Lambda}_1 \mathbf{F}_1 + \mathbf{e}_1 \\ \mathbf{d}_2 &= \mathbf{\Lambda}_2 \mathbf{F}_1 + \mathbf{e}_2 \\ \mathbf{d}_3 &= \mathbf{\Lambda}_3 \mathbf{F}_2 + \mathbf{e}_3 \\ \mathbf{d}_4 &= \mathbf{\Lambda}_4 \mathbf{F}_2 + \mathbf{e}_4\end{aligned}$$

The matrix of factor loadings $\mathbf{\Lambda}_1$ is $p_1 \times p_1$. The variables in \mathbf{d}_1 are selected to ensure that $\mathbf{\Lambda}_1$ has an inverse. The variables in \mathbf{d}_3 are selected so that $\mathbf{\Lambda}_3$ has an inverse. Suppose it is impossible to select a subset of observable variables so that $\mathbf{\Lambda}_1$ has an inverse. If so, the columns of the *combined* matrix of factor loadings for the first model are linearly dependent. This holds only a set of volume zero in the parameter space. The same applies to the second model.

In the combined model, the only new parameters are contained in the $p_1 \times p_2$ matrix $cov(\mathbf{F}_1, \mathbf{F}_2)$, which will be denoted by $\mathbf{\Phi}_{12}$. We have

$$\begin{aligned} cov(\mathbf{d}_1, \mathbf{d}_3) &= cov(\mathbf{\Lambda}_1 \mathbf{F}_1 + \mathbf{e}_1, \mathbf{\Lambda}_3 \mathbf{F}_2 + \mathbf{e}_3) \\ &= \mathbf{\Lambda}_1 cov(\mathbf{F}_1, \mathbf{F}_2) \mathbf{\Lambda}_3^\top + \mathbf{O} + \mathbf{O} + \mathbf{O} \\ &= \mathbf{\Lambda}_1 \mathbf{\Phi}_{12} \mathbf{\Lambda}_3^\top \end{aligned}$$

Since the matrices $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_3$ are already identified, they may be used to solve for $\mathbf{\Phi}_{12}$. Denoting $cov(\mathbf{d}_1, \mathbf{d}_3)$ by $\mathbf{\Sigma}_{13}$,

$$\begin{aligned} \mathbf{\Lambda}_1^{-1} \mathbf{\Sigma}_{13} (\mathbf{\Lambda}_3^\top)^{-1} &= \mathbf{\Lambda}_1^{-1} \mathbf{\Lambda}_1 \mathbf{\Phi}_{12} \mathbf{\Lambda}_3^\top (\mathbf{\Lambda}_3^\top)^{-1} \\ &= \mathbf{\Phi}_{12}, \end{aligned}$$

completing the proof.

Note that if the factor analysis sub-models have been identified using any of the rules given so far in this chapter, then there is at least one reference variable for each factor. In this case, $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_3$ are diagonal matrices with non-zero diagonal elements, and both inverses exist. If factor loadings have been set to one in the surrogate models, then $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_3$ are identity matrices. In practice, the part of the combination rule that says “except possibly on a set of volume zero” does not come into play.

The Extra Variables Rule The extra variables rule says that if the parameters of a factor analysis model are identifiable, more observable variables may be added to the model without adding any new factors. Identifiability is preserved, provided that the error terms for the new variables are uncorrelated with the error terms for observable variables already in the model (as well as being uncorrelated with the factors, of course). It is okay for the error terms of the additional variables to be correlated with one another. Straight arrows with factor loadings on them may point from each existing factor to each new variable. It is not necessary to include all such arrows. There are no restriction on the factor loadings of the variables that are being added to the model. There are no restriction on the covariances of error terms for the new set of variables, except that they must not be correlated with error terms already in the model.

The extra variable rule and the [reference variable rule](#) have something in common. They both allow inclusion of an additional set of observable variables that are influenced by all factors, and whose error terms need not be independent. When both rules apply, the reference variable rule may be preferable, because it allows some covariances between the error terms of the new variables and the error terms of variables already in the model;

hence; it is more flexible. On the other hand, to add more observable variables to a non-standard model like the one in the political democracy Example 3.3.2, the extra variables rule is the way to go.

Rule 2g: Extra Variables Rule If the parameters of a factor analysis model are identifiable, then a set of additional observable variables (without any new factors) may be added to the model. In the path diagram, straight arrows with factor loadings on them may point from each existing factor to each new variable. Error terms for the new variables may have non-zero covariances with each other. If the error terms of the new set have zero covariance with the error terms of the initial set and with the factors, then the parameters of the combined model are identifiable, except possibly on a set of volume zero in the parameter space.

Proof. In the initial model, there are p factors and k_1 observed variables. All parameters of the initial model are identifiable. The observed variables of the initial model are divided into two subsets, one with p variables, and the other with $k_1 - p$ variables. The model equations are

$$\begin{aligned}\mathbf{d}_1 &= \mathbf{\Lambda}_1 \mathbf{F} + \mathbf{e}_1 \\ \mathbf{d}_2 &= \mathbf{\Lambda}_2 \mathbf{F} + \mathbf{e}_2 \\ \mathbf{d}_3 &= \mathbf{\Lambda}_3 \mathbf{F} + \mathbf{e}_3,\end{aligned}$$

where the observed variables from the initial model are in \mathbf{d}_1 and \mathbf{d}_2 , and the new variables are in \mathbf{d}_3 . The variables in \mathbf{d}_1 are chosen so that the $p \times p$ matrix $\mathbf{\Lambda}_1$ has an inverse. This will be impossible if and only if the entire matrix of factor loadings for the initial model has columns that are linearly dependent, a condition that holds on a set of volume zero in the parameter space.

We have $cov(\mathbf{F}) = \mathbf{\Phi}$ and

$$cov \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{\Omega}_{11} & \mathbf{\Omega}_{12} & \mathbf{0} \\ & \mathbf{\Omega}_{22} & \mathbf{0} \\ & & \mathbf{\Omega}_{33} \end{pmatrix}.$$

The parameters to be identified are in the matrices $\mathbf{\Lambda}_3$ and $\mathbf{\Omega}_{33}$. The covariance matrix of the observable variables is

$$\begin{aligned}cov \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \end{pmatrix} &= \mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} & \mathbf{\Sigma}_{13} \\ & \mathbf{\Sigma}_{22} & \mathbf{\Sigma}_{23} \\ & & \mathbf{\Sigma}_{33} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{\Phi} + \mathbf{\Omega}_{11} & \mathbf{\Phi} \mathbf{\Lambda}_2^\top & \mathbf{\Phi} \mathbf{\Lambda}_3^\top \\ & \mathbf{\Lambda}_2 \mathbf{\Phi} \mathbf{\Lambda}_2^\top + \mathbf{\Omega}_{22} & \mathbf{\Lambda}_2 \mathbf{\Phi} \mathbf{\Lambda}_3^\top \\ & & \mathbf{\Lambda}_3 \mathbf{\Phi} \mathbf{\Lambda}_3^\top + \mathbf{\Omega}_{33} \end{pmatrix}.\end{aligned}$$

The parameters of the initial model are all identifiable, so they may be used to solve for the matrices Λ_3 and Ω_{33} . This is straightforward:

$$\begin{aligned}\Lambda_3 &= \Sigma_{13}^\top \Phi^{-1} \\ \Omega_{33} &= \Sigma_{33} - \Lambda_3 \Phi \Lambda_3^\top \quad \blacksquare\end{aligned}$$

The Error-free Rule Starting with a factor analysis model with identifiable parameters, add an observable variable *to the factors*. Often it's an observed exogenous variable (like sex or a dummy variable for experimental condition) that is hypothesized to affect some of the latent variables in a general structural equation model. It is convenient to make such variables part of the latent variable model.

Suppose parameters of an existing factor analysis model with p factors) are all identifiable. Add an observable scalar variable x that is independent of the error terms, and may have non-zero covariances with the factors. Thinking of x as an additional factor, we are adding a row (and column) to Σ , and a row (and column) to Φ . There are $p + 1$ additional parameters that need to be identified. One of these is the variance of x , which is obtained immediately as $\phi_{p+1,p+1} = \sigma_{k+1,k+1}$. The other new parameters are covariances between x and the factors, which are identified as follows.

As in a couple of earlier proofs, the observed variables from the existing model are divided into two vectors \mathbf{d}_1 and \mathbf{d}_2 , yielding the model equations

$$\begin{aligned}\mathbf{d}_1 &= \Lambda_1 \mathbf{F} + \mathbf{e}_1 \\ \mathbf{d}_2 &= \Lambda_2 \mathbf{F} + \mathbf{e}_2\end{aligned}$$

where the variables in \mathbf{d}_1 are chosen so that the $p \times p$ matrix Λ_1 has an inverse. This will be impossible if and only if the entire matrix of factor loadings for the existing model has columns that are linearly dependent, a condition that holds on a set of volume zero in the parameter space.

Let Σ_{x,d_1} denote the vector of covariances between x and the variables in \mathbf{d}_1 , and let $\Phi_{x,F}$ denote the vector of covariances between x and the other factors. Σ_{x,d_1} is part of the last row (column) of Σ , and $\Phi_{x,F}$ is part of the last row (column) of Φ . We have

$$\begin{aligned}\Sigma_{x,d_1} &= \text{cov}(x, \mathbf{d}_1) \\ &= \text{cov}(x, \Lambda_1 \mathbf{F} + \mathbf{e}_1) \\ &= \Lambda_1 \text{cov}(x, \mathbf{F} + \text{cov}(x, \mathbf{e}_1)) \\ &= \Lambda_1 \Phi_{x,F} + \mathbf{0},\end{aligned}$$

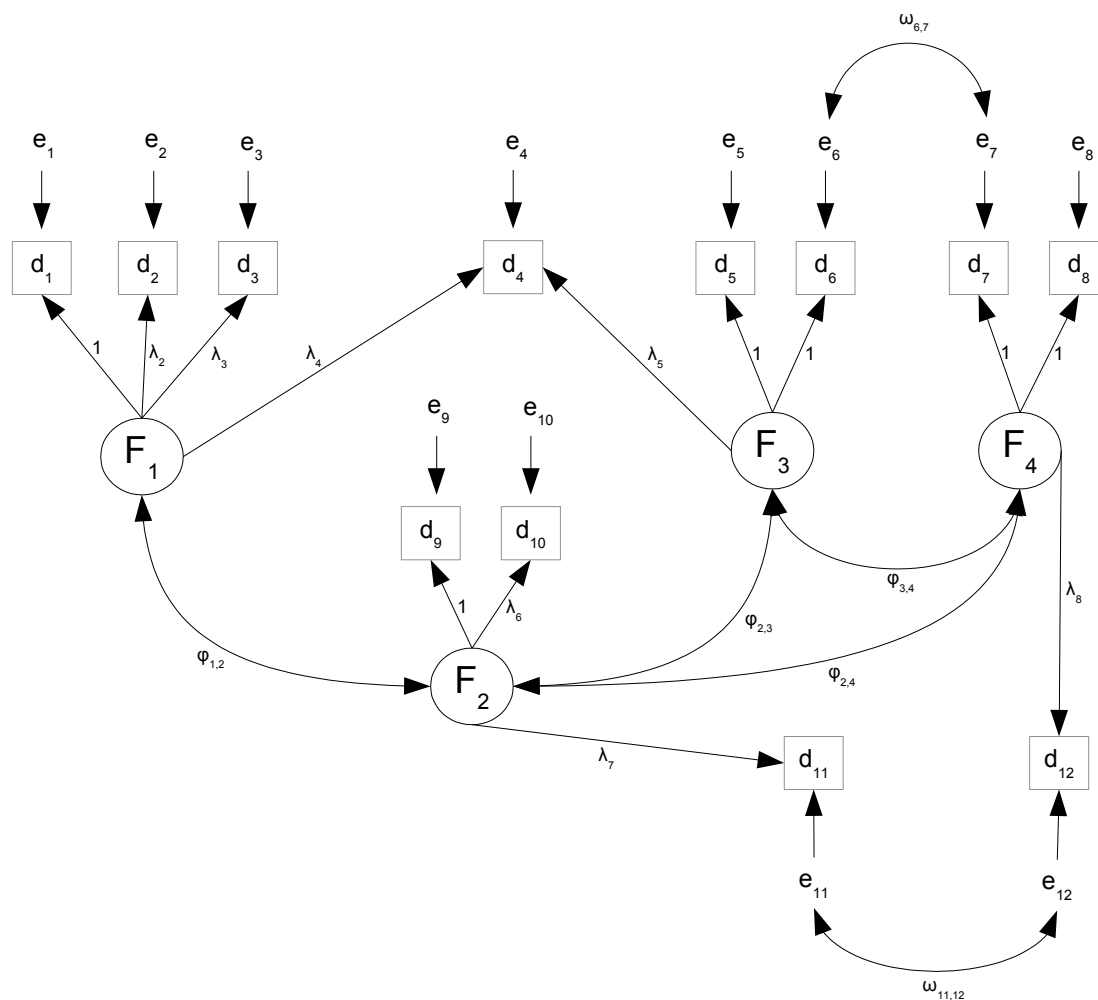
so that $\Phi_{x,F} = \Lambda_1^{-1} \Sigma_{x,d_1}$. Since Λ_1 is already identified, this completes the proof of the error-free rule. The rule will be stated as it applies to a vector of new observed variables.

Rule 2h: The Error-free Rule A set of observable variables may be added to the factors of a measurement model whose parameters are identifiable, provided that the new observed variables are independent of the error terms that are already in the model. The parameters of the resulting model are identifiable, except possibly on a set of volume zero in the parameter space.

3.6 Putting the Rules Together

Figure 3.10 shows a big, hairy confirmatory factor analysis model. Trying to establish

Figure 3.10: A Confirmatory Factor Analysis Model



identifiability by solving covariance structure equations would be a huge clerical task; instead, we will use the identifiability rules. See Appendix D for a collection of the identifiability rules in outline form.

There are twelve observable variables, so that Σ has $12(12+1)/2 = 78$ unique elements. The number one on some of the straight arrows tells us that this is a surrogate model in which at least one factor loading has been set to one for each factor. Counting parameters, there are 4 variances of the factors (denoted $\phi_{j,j}$), 4 possibly non-zero covariances between factors, and 7 factor loadings that are not fixed to the value one. There are 12 error variances (denoted $\omega_{j,j}$), and 2 possibly non-zero covariances between error terms. In all,

that's 78 covariance structure equations in $4 + 4 + 7 + 12 + 2 = 29$ unknown parameters. Because there are more covariance structure equations than parameters, the model passes the test of the [parameter count rule](#), and identifiability cannot be ruled out.

We will establish identifiability in two ways, first without using the [reference variable rule](#), and then using it.

Without using the reference variable rule The strategy will be to apply the rules to parts of the model, and then put the sub-models together. First, consider the sub-model involving F_1 , d_1 , d_2 and d_3 . Its parameters are identifiable by the [three-variable rule](#), provided that λ_2 and λ_3 are both non-zero. This could be verified empirically by testing $H_0 : Corr(d_1, d_2) = 0$ and $H_0 : Corr(d_1, d_3) = 0$. We have identified six parameters: $\phi_{1,1}$, λ_2 , λ_3 , $\omega_{1,1}$, $\omega_{2,2}$ and $\omega_{3,3}$.

Next, look at the part involving F_3 , F_4 , and d_5 through d_8 . The [double measurement rule](#) covers this, including the covariance between e_6 and e_7 . Just consider d_6 and d_7 to be part of the same "set" of measurements, perhaps conducted at the same time by the same personnel. This identifies eight more parameters: $\phi_{3,3}$, $\phi_{4,4}$, $\phi_{3,4}$, $\omega_{5,5}$, $\omega_{6,6}$, $\omega_{7,7}$, $\omega_{8,8}$ and $\omega_{6,7}$.

Now put the two sub-models together using the [combination rule](#). Notice that the variables d_4 and d_{12} are not included yet; they are being saved for later. Also, the zero covariance between F_1 and the other factors presents no obstacle. No new parameters have been identified in this case, but merging the sub-models helps with the next step.

The next step is to add the part involving F_2 , d_9 and d_{10} to the combined sub-model. The [two-variable addition rule](#) allows this, provided $\phi_{1,2}$, $\phi_{2,3}$ and $\phi_{2,4}$ are not all zero. According to the model, if $\phi_{1,2}$ were zero, then d_9 and d_{10} would be uncorrelated with d_1 , d_2 and d_3 ; this is testable. The conditions $\phi_{2,3} \neq 0$ and $\phi_{2,4} \neq 0$ could be verified in a similar way, and only one of the three covariances with F_2 needs to be non-zero for the two-variable addition rule to apply. In this way, seven more parameters are identified: $\phi_{2,2}$, $\phi_{1,2}$, $\phi_{2,3}$, $\phi_{2,4}$, λ_6 , $\omega_{9,9}$ and $\omega_{10,10}$.

At this point, we have a (big) sub-model whose parameters are all identifiable, and which includes all the factors. Now use the [extra variables rule](#) to add the remaining observable variables d_4 , d_{11} and d_{12} , quickly checking that their error terms are not correlated with any of the error terms already in the model. Eight more parameters are identified: λ_4 , λ_5 , λ_7 , λ_8 , $\omega_{4,4}$, $\omega_{11,11}$, $\omega_{12,12}$ and $\omega_{11,12}$.

That does it. There were 29 parameters to identify, and we identified $6 + 8 + 7 + 8 = 29$. Notice how, at several points in the argument, empirical tests were proposed to verify that the true parameter vector was in a region of the parameter space where the parameters involved were identifiable. One can extend this dual strategy of identification checking and empirical testing, by testing sub-models for fit, and then testing fit again as the sub-models are combined. This way, if the final model does not fit the data, you probably will have a good idea why.

Checking identifiability using the reference variable rule The rule requires that the number of observable variables be at least three times the number of factors. The model

has four factors and twelve observable variables, so the first requirement is satisfied — just barely. The next requirement is that every factor have at least one reference variable. A quick glance verifies this condition. In fact, every factor has at least two reference variables. At least one reference variable for every factor has a factor loading of one, so this is a nice unstandardized surrogate model; the third condition of the rule is satisfied.

The model has only two non-zero covariances between error terms, so as long as d_6 and d_7 go in the same set of variables and d_{11} and d_{12} do in the same set, all the parameters are identifiable except possibly on a set of volume zero in the parameter space. Let's take a closer look at this issue.

Referring back to Model (3.25) on page 306, observe that the lowerdimensional set of parameter values where identifiability fails is the set where the square sub-matrices Λ_2 and Λ_3 do not have inverses. Mathematically, this could happen just because of the values that the factor loadings happen to have, and there's really nothing we can do about it. More concerning would be if it happened because of definite zeros in a model we more or less believe, like the model of Figure 3.10. To check this, I wrote down the factor matrix, shown in (3.27). The rows are re-arranged (there is more than one way to do it) so that Λ_2 and Λ_3 both have inverses, provided that most of the λ_j are non-zero. The only exception I see is that λ_4 could be zero. The other way of proving identifiability (without the reference variable rule) also requires that most of the λ_j be non-zero.

$$\begin{pmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{pmatrix} = \begin{array}{c|cccc} & F_1 & F_2 & F_3 & F_4 \\ \hline d_1 & 1 & 0 & 0 & 0 \\ d_9 & 0 & 1 & 0 & 0 \\ d_6 & 0 & 0 & 1 & 0 \\ d_7 & 0 & 0 & 0 & 1 \\ \hline d_2 & \lambda_2 & 0 & 0 & 0 \\ d_{10} & 0 & \lambda_6 & 0 & 0 \\ d_5 & 0 & 0 & 1 & 0 \\ d_8 & 0 & 0 & 0 & 1 \\ \hline d_4 & \lambda_4 & 0 & \lambda_5 & 0 \\ d_3 & \lambda_3 & 0 & 0 & 0 \\ d_{11} & 0 & \lambda_7 & 0 & 0 \\ d_{12} & 0 & 0 & 0 & \lambda_8 \end{array} \quad (3.27)$$

Most of the time, it is not necessary to write down the complete factor matrix in order to verify that the [reference variable rule](#) applies — but it's quite informative here. The main lesson is that while the model of Figure 3.10 seems to have a lot of arrows, it is actually a very sparse special case of the model (Model 3.25) that underlies the reference variable rule. In (3.27), 25 factor loadings are set to zero or one, while they are unconstrained under Model (3.25). These all represent testable null hypotheses¹¹ rather than assumptions. In addition, Figure 3.10 has only two potentially non-zero covariances between error terms, while Model (3.25) allows $3 \times p(p-1) = 36$. In all, that's $25 + 34 = 59$ ways in which the

¹¹A factor loading will equal one under this surrogate model if and only if two factor loadings are equal under the original model.

model of Figure 3.10 might fail to fit the data, while Model (3.25) could fit very well.

This raises a question. What should be done if the model does not fit? If one is using the reference variable rule, the answer is pretty obvious. Fit a model with the factor loadings in Λ_2 and Λ_3 unconstrained, and test the 25 null hypotheses with z -tests. Any null hypothesis that is rejected points to a constraint on the parameter values that is contributing to the lack of fit. If the model with unconstrained factor loadings still does not fit, a second line of attack is to start testing hypotheses about covariances between error terms. This is tough to do in an honest way without more information about how the data were collected. The observable variables may not naturally divide themselves into subsets whose error terms can be assumed independent, because the study may not have been planned with this in mind.

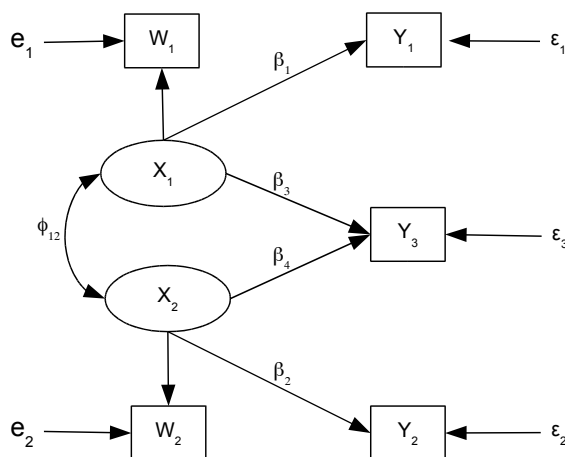
This is all possible with the reference variable rule in hand. Without the rule, it would be hard to know what to do. The only real choice would be to start guessing and trying to solve equations. Good luck.

More examples of applying the rules

Example 3.6.1 *A latent variable regression*

This example is based on the fact that a regression model with latent explanatory variables and observed response variables may be viewed as a confirmatory factor analysis model. Figure 3.11 reproduces Figure 19 on page 117. The [reference variable rule](#) does not apply

Figure 3.11: Regression with latent explanatory variables as a confirmatory factor analysis (Reproduction of Figure 19)

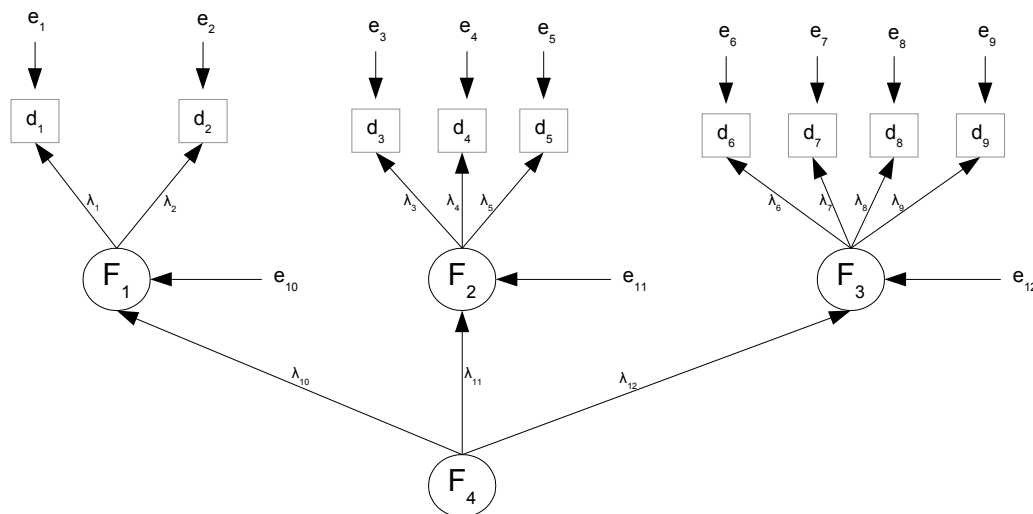


because there are two factors and only $5 < 6$ observable variables, but the parameters are immediately identifiable by the [two-variable rule](#), except at points in the parameter space where $\phi_{1,2} = 0$. Detailed calculations like the ones in Chapter 0 are usually unnecessary if you know some identifiability rules.

Example 3.6.2 *A second-order factor analysis*

Figure 3.12 shows a simple *second-order factor analysis* model. The idea behind higher order factor analysis is that the observed variables reflect a set of unobservable factors, and those factors in turn reflect the operation of another set of factors at a deeper level. In principle, there could be third-order factors influencing the second-order factors, and

Figure 3.12: Second-order factor analysis



so on.

In a higher-order factor analysis model, the higher-order factors (second order and above) have no direct influence on the observed variables. Perhaps surprisingly, it is still possible to apply the identifiability rules we have. In Figure 3.12, none of the factor loadings is explicitly set to one, so assume the factors are standardized, and that the sign of one factor loading is known for each factor. This includes the set λ_{10} , λ_{11} and λ_{12} .

To check identifiability, adopt a two-stage approach. First, look at the first-order factors F_1 , F_2 and F_3 . Imagine curved, double headed arrows connecting them. The covariances will be determined by λ_{10} , λ_{11} and λ_{12} , but ignore F_4 and the straight arrows from F_4 to the first-order factors for now.

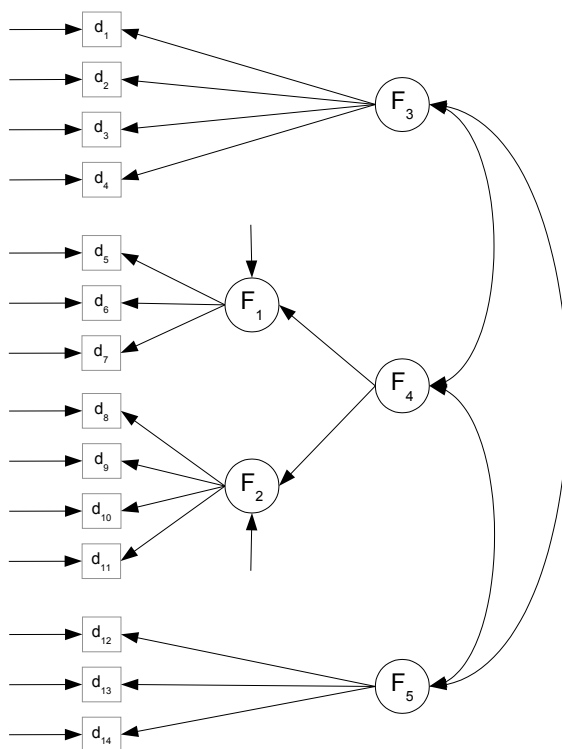
It's clear that the system involving F_2 , F_3 and d_3 through d_8 is identified by the [three-variable rule](#), and then the system involving F_1 , d_1 and d_2 can be brought in with the [two-variable addition rule](#). The factor loadings λ_1 through λ_9 and the error variances ω_1 through ω_9 have all been identified, as have the covariances (correlations) of F_1 , F_2 and F_3 .

Now think of F_1 through F_3 as observed variables, and let their correlation matrix (identified by the argument above) play the role of Σ . By the [three-variable rule](#), the factor loadings λ_{10} through λ_{12} are all identifiable, provided they are non-zero. That's it, and that's how it goes in general. The stages in the identifiability proof follow the stages in the model.

Example 3.6.3 *Another higher-order model*

Figure 3.13 shows another confirmatory factor analysis model. This one is a sort of hybrid, with both first-order and second-order features. A picture like this could arise

Figure 3.13: Mixed first-order and second-order factor analysis



quite naturally in the course of model development. The investigator has several factors in mind, and several observed variables designated to measure each one. For example in Figure 3.13, d_1 through d_4 could be measures of left-right political orientation, d_5 through d_{11} could be measures of academic performance (which would be called “intelligence” by some), and d_{12} through d_{14} could be measures of self-esteem. To check uni-dimensionality, the investigator carries out separate exploratory factor analyses (yes, *exploratory*) on the three subsets of observable variables. If everything is okay, a single-factor model should fit each one.

It works out okay for political orientation and self esteem, but for d_5 through d_{11} , two factors are required. After rotation, it looks like d_5 through d_7 load primarily on one factor, while d_8 through d_{11} load on the other. The first set of variables depend on solving puzzles and math problems, while the second set depend on knowing the definitions of words and on reading a brief passage and then answering questions about it. One could call these factors “Math” and “Verbal,” and nobody would argue.

Unfortunately, the factors are orthogonal, because it’s a generic exploratory factor analysis model. It may fit the data, but only because of the arrows running from F_1 to

d_8 through d_{11} , and from F_2 to d_5 through d_7 . This crossover pattern is not identifiable (the extra variables rule does not apply), and it's incompatible with the path diagram in Figure 3.13. Also, separate Math and Verbal factors do not accord with the investigator's theory or research questions, which are about a single thing called "intelligence." Figure 3.13 shows a really nice solution, which allows the Math and Verbal factors to be somewhat distinct, but correlated because they both reflect a second-order factor — and that factor is what the investigator wants to study.

Two comments are in order. First, I did not think of this cute data analysis trick. I saw it in a low-grade empirical research paper, and I am still searching for the source of the idea so I can give proper credit. Second, the forgoing discussion points out the fact that like most statistical methods, confirmatory factor analysis is often used in an exploratory way. In practice, the user will try quite a few models until finding one that fits the data adequately, and then carry out a boatload of statistical tests. In the end, only one model and a few of the tests will be reported, and the discussion will make it seem like it was planned all along. There is lots of opportunity for overfitting, and for apparent findings that actually reflect coincidences in the data. The solution is to replicate the results on a second, independent set of data. Without this kind of cross-validation, the so-called "conclusions" should be treated as data-driven hypotheses. Again, this situation is not limited to confirmatory factor analysis and structural equation models. It is true of most statistical applications.

Now consider identifiability for Figure 3.13. Parameters of the first-order system involving F_1 and F_2 (with a curved, double-headed arrow between the factors) and d_5 through d_{11} are identifiable by the [three-variable rule](#). Now bring d_1 through d_4 and d_{12} through d_{14} into the first-order model, using the [error-free rule](#). That is, treat these observable variables as factors that are measured without error. The result is an ordinary second-order factor analysis model in which the second-order factors are F_3 , F_4 and F_5 . The system involving F_3 and F_5 is identified¹² by the [three-variable rule](#). The system involving F_1 , F_2 and F_4 is then brought in with the [two-variable addition rule](#).

All the parameters are identifiable except on a set of volume zero in the parameter space, so it's mission accomplished — sort of. In this case, the set of volume zero where identifiability fails happens to include some interesting points, namely the points where $Cov(F_3, F_4) = 0$ and $Cov(F_4, F_5) = 0$. At least one of these covariances needs to be nonzero in order for the two-variable addition rule to work in the last stage of the proof.

The whole point of the study is probably the connections between F_3 , F_4 and F_5 . If the investigator tries to test the null hypothesis that all three covariances are zero using a likelihood ratio test, the process will fail. It will be impossible to fit the restricted model, because the likelihood function will have a non-unique maximum on an infinite connected set. If in reality part of the null hypothesis is true, with both $Cov(F_3, F_4) = 0$ and $Cov(F_4, F_5) = 0$, then there could easily be numerical difficulties in fitting the *unrestricted* model. Fortunately, the model says that $Cov(F_3, F_4) = 0$ if and only if the matrix of covariances between $(d_1, d_2, d_3, d_4)^\top$ and $(d_5, \dots, d_{11})^\top$ consists only of zeros. This can

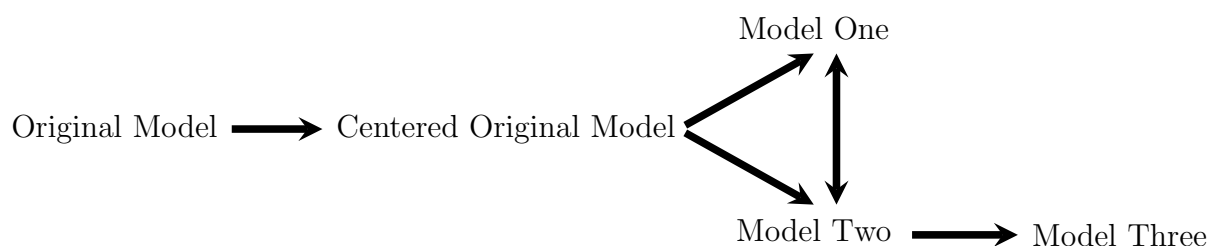
¹²That is, the parameters are a function of the variances and covariances of the first-order factors, which in turn are functions of the variances and covariances of the observable variables.

be tested with off-the-shelf canonical correlation methods (see R's `CCP` package), and $Cov(F_4, F_5) = 0$ can be diagnosed in a similar way.

3.7 Standardized Observed Variables

Standardizing the observed variables is familiar from exploratory factor analysis; see Chapter 2. In confirmatory factor analysis, it is a change of variables that leads to another level of surrogate model, beyond the standard choices of standardizing factors or setting factor loadings to one. Figure 3.14 shows the most common situation. Standardizing

Figure 3.14: Standardizing the Observed Variables



the observed variables just means dividing them by their standard deviations, since they already have expected value zero under the common centered surrogate models. While this operation may be applied at any point in the re-parameterization process, it is most commonly applied to a model with standardized factors (Model Two). The surrogate model with both standardized factors and standardized observed variables will be called Model Three.

Consider the model equations $\mathbf{d} = \mathbf{\Lambda}\mathbf{F} + \mathbf{e}$, with the double primes (if any) hidden, and not bothering to separate the data vector into \mathbf{d}_1 and \mathbf{d}_2 . Let $\mathbf{W} = dg(\mathbf{\Sigma})$, where as usual, $\mathbf{\Sigma} = cov(\mathbf{d})$. Then,

$$\begin{aligned}
 \mathbf{z} &= \mathbf{W}^{-1/2}\mathbf{d} \\
 &= \mathbf{W}^{-1/2}(\mathbf{\Lambda}\mathbf{F} + \mathbf{e}) \\
 &= (\mathbf{W}^{-1/2}\mathbf{\Lambda})\mathbf{F} + (\mathbf{W}^{-1/2}\mathbf{e}) \\
 &= \mathbf{\Lambda}'''\mathbf{F} + \mathbf{e}''',
 \end{aligned}$$

where

$$\mathbf{\Lambda}''' = \mathbf{W}^{-1/2}\mathbf{\Lambda} \quad \text{and} \quad cov(\mathbf{e}''') = \mathbf{\Omega}''' = \mathbf{W}^{-1/2}\mathbf{\Omega}\mathbf{W}^{-1/2}. \quad (3.28)$$

One thing to notice about standardizing the observed variables is that while it affects $\mathbf{\Lambda}$ and $\mathbf{\Omega}$, the covariances between factors in the matrix $\mathbf{\Phi}$ are unaffected. This is a fortunate, since observed variables are usually standardized only if the factors are standardized, and when the factors are standardized, covariances between factors equal correlations under

the original model. The nice interpretation is preserved – so at least, standardizing the observed variables does no harm.

It can also do some good. We will now see that when the observed variables are standardized, the factor loading for a reference variable is the correlation of the reference variable with the latent variable it measures, under the original model. Also, the variance of the error term (for all observed variables, not just reference variables) is the proportion of variance in that variable that is due to error — again, under the original model.

Correlations between factors and their reference variables Let d_ℓ be a reference variable for factor j , so that under the centered original model, $d_\ell = \lambda_{\ell,j}F_j + e_\ell$. Choosing explicitness over simplicity, we will employ the notation of Section 3.3.1, and use double primes to indicate quantities under Model Two, in which the factors have been standardized. We have

$$\begin{aligned} \text{cov}(F_j, d_\ell) &= \text{cov}(F_j, \lambda_{\ell,j}F_j + e_\ell) \\ &= \text{cov}(F_j, \lambda''_{\ell,j}F_j + e''_\ell) \\ &= \lambda_{\ell,j}\text{cov}(F_j, F_j) + \text{cov}(F_j, e_\ell) \\ &= \lambda_{\ell,j}\phi_{j,j}, \end{aligned}$$

so that

$$\begin{aligned} \text{corr}(F_j, d_\ell) &= \frac{\lambda_{\ell,j}\phi_{j,j}}{\sqrt{\phi_{j,j}}\sqrt{\sigma_{\ell,\ell}}} \\ &= \frac{\lambda_{\ell,j}\sqrt{\phi_{j,j}}}{\sqrt{\sigma_{\ell,\ell}}}. \end{aligned} \tag{3.29}$$

Consider the model with both F_j and d_ℓ standardized. Recalling how we got there,

$$\begin{aligned} d_\ell &= \lambda_{\ell,j}F_j + e_\ell \\ &= \lambda_{\ell,j}\sqrt{\phi_{j,j}} \left(\frac{1}{\sqrt{\phi_{j,j}}} \right) F_j + e_\ell \\ &= \lambda''_{\ell,j}F''_j + e_\ell. \end{aligned}$$

Then standardizing d_ℓ as well,

$$\begin{aligned} z_\ell &= \frac{1}{\sqrt{\sigma_{\ell,\ell}}}\lambda''_{\ell,j}F''_j + \frac{1}{\sqrt{\sigma_{\ell,\ell}}}e_\ell \\ &= \lambda'''_{\ell,j}F''_j + e'''_\ell. \end{aligned}$$

Now un-wrap $\lambda_{\ell,j}'''$, the factor loading of the reference variable under this “completely standardized” model¹³.

$$\begin{aligned}\lambda_{\ell,j}''' &= \frac{1}{\sqrt{\sigma_{\ell,\ell}}} \lambda_{\ell,j}'' \\ &= \frac{1}{\sqrt{\sigma_{\ell,\ell}}} \lambda_{\ell,j} \sqrt{\phi_{j,j}},\end{aligned}$$

which is exactly expression (3.29) for the correlation between the factor and its reference variable, under the original model. Squaring the factor loading yields the reliability of the reference variable — the proportion of variance in the reference variable that arises from the quantity it is measuring, and not error. It is always helpful when the parameters of a surrogate model correspond to something important about the original model.

Uniqueness As discussed in Chapter 2, the uniqueness of an observed variable is the proportion of its variance that comes from error (the unique factor) and not the common factors. For reference variables, the uniqueness is one minus the reliability.

Let $\boldsymbol{\lambda}_\ell$ denote row ℓ of the factor matrix $\mathbf{\Lambda}$ in the original model. This is the row corresponding to the observed variable d_ℓ . If d_ℓ is a reference variable, $\boldsymbol{\lambda}_\ell$ has only one non-zero element, but that need not be the case here. If the observed variables are *not* standardized, $d_\ell = \boldsymbol{\lambda}_\ell \mathbf{F} + e_\ell$. When the observed variables are standardized (whether or not the factors are standardized as well), $z_\ell = \boldsymbol{\lambda}_\ell''' \mathbf{F} + e_\ell'''$, and

$$\begin{aligned}\text{Var}(z_\ell) &= \text{cov}(\boldsymbol{\lambda}_\ell''' \mathbf{F} + e_\ell''') \\ &= \boldsymbol{\lambda}_\ell''' \boldsymbol{\Phi} \boldsymbol{\lambda}_\ell'''^\top + \omega_{\ell,\ell}'''.\end{aligned}\tag{3.30}$$

By (3.28), $\omega_{\ell,\ell}''' = \omega_{\ell,\ell} / \sigma_{\ell,\ell}$. It is exactly the uniqueness of d_ℓ under the original model. That is, it is the proportion of variance in the observed variable d_ℓ that comes from error (the unique factor) and not the common factors. For example, if the value of such a parameter is something like 0.85, it means that the variable in question is 85% noise. Uniqueness is worth estimating, and standardizing the observed variables makes the process more convenient.

Reduction of the parameter space Because $\text{var}(z_\ell = 1)$, expression (3.30) says that $\omega_{\ell,\ell}''' = \text{Var}(e_\ell''') = 1 - \boldsymbol{\lambda}_\ell''' \boldsymbol{\Phi} \boldsymbol{\lambda}_\ell'''^\top$. That is, the variances of the error terms are functions of the other parameters in the model. The dimension of the parameter space has been reduced by k , the number of observed variables. We will now see that for almost all models used in practice, this reduction of the parameter space has no effect on identifiability or model fit.

¹³Some software, including lavaan, calls models and their estimates “completely” standardized when both the factors and the observable variables are standardized

A modest assumption As in 3.30, $d_\ell = \lambda_\ell \mathbf{F} + e_\ell$ implies $\text{Var}(d_\ell) = \sigma_{\ell,\ell} = \lambda_\ell \Phi \lambda_\ell^\top + \omega_{\ell,\ell}$. That is, the variances of the observed variables are something plus an $\omega_{\ell,\ell}$. Suppose that this is the only place in Σ where $\omega_{\ell,\ell}$ appears, and also suppose that for $\ell = 1, \dots, j$, the error variance $\omega_{\ell,\ell}$ is not subject to any constraints, such as some of them being equal to one another or to other model parameters. This is typical of most models used in practice, and it leads to some useful conclusions.

Identifiability When solving covariance structure equations to prove identifiability, it is natural to set the diagonal elements of Ω aside and solve for the other parameters first. If it works, one can then obtain the error variances by subtraction. When the observed variables are standardized, the whole process is the same except that the last step is omitted. This implies that the identifiability status of a model is not changed if the observed variables are standardized — given the “modest assumption” of the paragraph above.

Equal diagonals Recall the meaning of $\Sigma(\theta)$. It’s just the covariance matrix of the observable variables (that is, Σ), written as a function of the model parameters θ .

As mentioned back on page 171, maximum likelihood estimation often proceeds by minimizing the objective function $g(\theta) = \text{tr}(\widehat{\Sigma}\Sigma(\theta)^{-1}) - \log|\widehat{\Sigma}\Sigma(\theta)^{-1}| - k$, which is equivalent to minimizing the minus log likelihood. The function $g(\theta)$ is a lot like a distance between $\Sigma(\theta)$ and $\widehat{\Sigma}$ ¹⁴. Other things being equal, anything that brings $\Sigma(\theta)$ closer to $\widehat{\Sigma}$ will reduce the value of $g(\theta)$. In particular, for any fixed values of the matrices Λ and Φ (and regardless of the off-diagonal elements of Ω), letting $\omega_{\ell,\ell} = \widehat{\sigma}_{\ell,\ell} - \lambda_\ell \Phi \lambda_\ell^\top$ for $\ell = 1, \dots, k$ will make the main diagonals of $\Sigma(\theta)$ and $\widehat{\Sigma}$ coincide, resulting in a lower value of $g(\theta)$. This also holds when $\Lambda = \widehat{\Lambda}$ and $\Phi = \widehat{\Phi}$. The conclusion is that for $\ell = 1, \dots, k$, we have $\widehat{\sigma}_{\ell,\ell} = \widehat{\lambda}_\ell \widehat{\Phi} \widehat{\lambda}_\ell^\top + \widehat{\omega}_{\ell,\ell}$. The right-hand side is a diagonal element of $\Sigma(\widehat{\theta})$, so that $dg(\Sigma(\widehat{\theta})) = dg(\widehat{\Sigma})$. Another way to express this is

$$\mathbf{W}(\widehat{\theta}) = \widehat{\mathbf{W}}. \quad (3.31)$$

This equality will come in handy very shortly. Once again, it holds when the error variances $\omega_{\ell,\ell}$ appear only in the diagonal of Ω , and are otherwise unconstrained¹⁵.

¹⁴It is non-negative, and it equals zero if and only if $\Sigma(\theta) = \widehat{\Sigma}$. I’m not sure whether it obeys the triangle inequality. This gap makes the argument less rigorous.

¹⁵Of course the model implies some constraints on the $\omega_{\ell,\ell}$. Since they are variances, they are must be non-negative. Also, if the covariance matrix Ω has any non-zero off-diagonal elements, the fact that it must be non-negative definite places additional limitations on the possible values of $\omega_{\ell,\ell}$. However, these constraints are not automatically enforced in a numerical search for the MLE, unless the user explicitly specifies inequality constraints. The result is that as the numerical optimization forces $dg(\Sigma(\widehat{\theta}))$ toward $dg(\widehat{\Sigma})$, an $\widehat{\omega}_{\ell,\ell}$ or two can easily become negative for some models and some data sets. This is the dreaded [Heywood case](#) (see p. 226). For a sufficiently large sample size, the consistency of maximum likelihood estimation guarantees that it cannot happen if the model is correct and the true parameter vector is in the interior of the parameter space. Negative variance estimates are a sign of poor model fit.

Estimation Parameter estimates for a model in which the observed variables are standardized may be obtained without re-fitting the model. The key is Expression (3.28), which is reproduced here for convenience. In most applications, $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ contain parameters from a surrogate model with standardized factors, so one could say they have invisible double primes.

$$\mathbf{\Lambda}''' = \mathbf{W}^{-1/2}\mathbf{\Lambda} \quad \text{and} \quad \text{cov}(\mathbf{e}''') = \mathbf{\Omega}''' = \mathbf{W}^{-1/2}\mathbf{\Omega}\mathbf{W}^{-1/2}.$$

It is tempting to just put hats on everything and invoke invariance, but you need to watch out. While $\widehat{\mathbf{W}} = dg(\widehat{\mathbf{\Sigma}})$ and $\widehat{\mathbf{\Sigma}}$ is an MLE, it's an MLE based on a generic multivariate normal model, not the same as the factor analysis model with $\mathbf{\Lambda}$ and $\mathbf{\Omega}$. What we really want is

$$\widehat{\mathbf{\Lambda}}''' = \mathbf{W}(\widehat{\boldsymbol{\theta}})^{-1/2}\widehat{\mathbf{\Lambda}} \quad \text{and} \quad \widehat{\mathbf{\Omega}}''' = \mathbf{W}(\widehat{\boldsymbol{\theta}})^{-1/2}\widehat{\mathbf{\Omega}}\mathbf{W}(\widehat{\boldsymbol{\theta}})^{-1/2}. \quad (3.32)$$

The distinction between $\widehat{\mathbf{W}}$ and $\mathbf{W}(\widehat{\boldsymbol{\theta}})$ does not matter when (3.31) holds, which is most of the time. Still, it's nice to know that lavaan uses (3.32). It took me a fair amount of work to verify this, because it's not that easy to come up with a model where (3.31) fails badly enough to have a noticeable effect.

To obtain standard errors and tests for a model with standardized observed variables, it is necessary to re-fit the model. There are two natural ways to proceed. The most obvious way is to literally standardize the observed variables; subtract off the sample means and then divide by the sample standard deviations¹⁶. The same results may be obtained by analyzing the sample correlation matrix rather than the covariance matrix. This will be illustrated in Section 3.8.

Testing goodness of fit When Expression (3.31) holds, standardizing the observed variables has no effect on the likelihood ratio test for model fit. This is established in the following theorem.

Theorem 3.1 *For a centered confirmatory factor analysis model, let $\boldsymbol{\theta}$ denote the parameter vector, and let $\mathbf{\Sigma} = \mathbf{\Sigma}(\boldsymbol{\theta})$ denote the $k \times k$ variance covariance matrix of the observable variables. The unique MLE of $\boldsymbol{\theta}$ is $\widehat{\boldsymbol{\theta}}$, and the sample variance-covariance matrix of the observable variables is $\widehat{\mathbf{\Sigma}}$. Let $\widehat{\mathbf{W}} = dg(\widehat{\mathbf{\Sigma}})$ and $\mathbf{W}(\widehat{\boldsymbol{\theta}}) = dg(\mathbf{\Sigma}(\widehat{\boldsymbol{\theta}}))$. If $\widehat{\mathbf{W}} = \mathbf{W}(\widehat{\boldsymbol{\theta}})$, then the test statistic of the likelihood ratio test for goodness of model fit is unchanged when the observed variables are standardized.*

Proof As given in (1.18), the test statistic for a model with unstandardized observed variables is $G^2 = n(\text{tr}\{\widehat{\mathbf{\Sigma}}\mathbf{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}\} - \log|\widehat{\mathbf{\Sigma}}\mathbf{\Sigma}(\widehat{\boldsymbol{\theta}})^{-1}| - k)$. In the model with standardized observed variables, $\widehat{\mathbf{\Sigma}}$ is replaced by the sample correlation matrix $\widehat{\mathbf{W}}^{-1/2}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{W}}^{-1/2}$, and

¹⁶Make sure you have n rather than $n - 1$ in the denominators of the standard deviations. This way, you are working with true MLEs.

$\Sigma(\hat{\theta})$ is replaced by $\mathbf{W}(\hat{\theta})^{-\frac{1}{2}} \Sigma(\hat{\theta}) \mathbf{W}(\hat{\theta})^{-\frac{1}{2}}$. The resulting test statistic is

$$\begin{aligned}
G_s^2 &= n \left(\text{tr} \left\{ \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \widehat{\mathbf{W}}^{-\frac{1}{2}} \left(\mathbf{W}(\hat{\theta})^{-\frac{1}{2}} \Sigma(\hat{\theta}) \mathbf{W}(\hat{\theta})^{-\frac{1}{2}} \right)^{-1} \right\} - \log \left| \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \widehat{\mathbf{W}}^{-\frac{1}{2}} \left(\mathbf{W}(\hat{\theta})^{-\frac{1}{2}} \Sigma(\hat{\theta}) \mathbf{W}(\hat{\theta})^{-\frac{1}{2}} \right)^{-1} \right| - k \right) \\
&= n \left(\text{tr} \left\{ \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \widehat{\mathbf{W}}^{-\frac{1}{2}} \left(\widehat{\mathbf{W}}^{-\frac{1}{2}} \Sigma(\hat{\theta}) \widehat{\mathbf{W}}^{-\frac{1}{2}} \right)^{-1} \right\} - \log \left| \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \widehat{\mathbf{W}}^{-\frac{1}{2}} \left(\widehat{\mathbf{W}}^{-\frac{1}{2}} \Sigma(\hat{\theta}) \widehat{\mathbf{W}}^{-\frac{1}{2}} \right)^{-1} \right| - k \right) \\
&= n \left(\text{tr} \left\{ \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \underbrace{\widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\mathbf{W}}^{\frac{1}{2}}}_{\mathbf{I}} \Sigma(\hat{\theta})^{-1} \widehat{\mathbf{W}}^{\frac{1}{2}} \right\} - \log \left| \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \underbrace{\widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\mathbf{W}}^{\frac{1}{2}}}_{\mathbf{I}} \Sigma(\hat{\theta})^{-1} \widehat{\mathbf{W}}^{\frac{1}{2}} \right| - k \right) \\
&= n \left(\text{tr} \left\{ \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \Sigma(\hat{\theta})^{-1} \widehat{\mathbf{W}}^{\frac{1}{2}} \right\} - \log \left| \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \Sigma(\hat{\theta})^{-1} \widehat{\mathbf{W}}^{\frac{1}{2}} \right| - k \right) \\
&= n \left(\text{tr} \left\{ \widehat{\mathbf{W}}^{\frac{1}{2}} \widehat{\mathbf{W}}^{-\frac{1}{2}} \widehat{\Sigma} \Sigma(\hat{\theta})^{-1} \right\} - \log \left(\left| \widehat{\mathbf{W}}^{-\frac{1}{2}} \right| \left| \widehat{\Sigma} \Sigma(\hat{\theta})^{-1} \right| \left| \widehat{\mathbf{W}}^{\frac{1}{2}} \right| \right) - k \right) \\
&= n \left(\text{tr} \left\{ \widehat{\Sigma} \Sigma(\hat{\theta})^{-1} \right\} - \log \left(\frac{\left| \widehat{\Sigma} \Sigma(\hat{\theta})^{-1} \right| \left| \widehat{\mathbf{W}}^{\frac{1}{2}} \right|}{\left| \widehat{\mathbf{W}}^{-\frac{1}{2}} \right|} \right) - k \right) \\
&= n \left(\text{tr} \left\{ \widehat{\Sigma} \Sigma(\hat{\theta})^{-1} \right\} - \log \left| \widehat{\Sigma} \Sigma(\hat{\theta})^{-1} \right| - k \right). \\
&= G^2 \quad \blacksquare
\end{aligned}$$

Normal inference is unaffected by standardizing Theorem 3.1 depends on $\mathbf{W}(\hat{\theta})$ being equal to $\widehat{\mathbf{W}}$. As indicated in the discussion leading up to (3.31), this condition holds when the error variances $\omega_{\ell,\ell}$ appear only in the diagonal of $\mathbf{\Omega}$ and are not functions of one another or of other parameters in the model. Most confirmatory factor analysis models employed in practice enjoy this property. Likelihood ratio tests are differences in G^2 fit statistics between a restricted and an unrestricted model. Wald tests are asymptotically equivalent to likelihood ratio tests under the null hypothesis. Confidence intervals can be obtained by inverting tests. The result is that for most confirmatory factor analyses, inference based on the normal model is unaffected by standardizing the observable variables. The choice to standardize or not is entirely a matter of convenience and interpretability.

3.8 The Holzinger and Swineford Data with lavaan

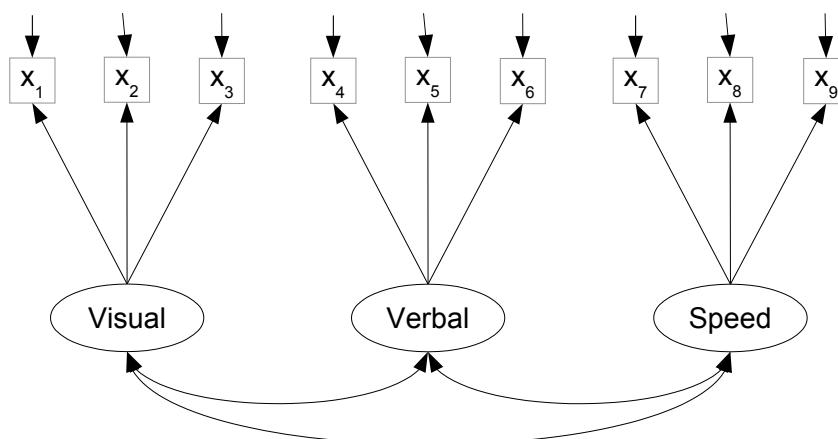
The Holzinger and Swineford (1939) data is a classic data set that is used in multiple textbooks and journal articles. It is included in the `lavaan` package, and is used in a confirmatory factor analysis example in the `lavaan` tutorial. The data were collected on students in grades seven and eight from two different schools. As in the `lavaan` tutorial, attention will be limited to nine tests of “mental ability” that are thought to reflect three factors: visualization (tests 1, 2 and 3), verbal or text processing (test 4, 5 and 6) and speed (tests 7, 8 and 9).

Visual		Verbal		Speed	
x_1	Visual Perception	x_4	Paragraph Comprehension	x_7	Addition
x_2	Cubes	x_5	Sentence Completion	x_8	Counting Dots
x_3	Lozenges	x_6	Word Meaning	x_9	Straight-Curved Capitals

The students actually took 24 tests; the full data set is available in the `MBESS` package.

Figure 3.15 shows a path diagram. It's pretty straightforward; all the parameters are identifiable at a glance by the [three-variable rule](#). As in Jöreskog's 1969 article [36], the analyses here will be limited to just the 145 children from the Grant-White school. This will provide a valuable cross-check of the numbers we obtain. Figure 3.15 represents Jöreskog's model (d).

Figure 3.15: Holzinger and Swineford Mental Test Data



As usual, the R code that follows is not just an example of how to do the job efficiently. Instead, it explores the capabilities of the software, and seeks to make connections between the computations and the ideas in the rest of the text. Students who imitate all these operations to do an assignment are missing the point. The examples you are most likely to want to follow tend to come near the end. The hope is by that point, you will know what's going on.

Acquiring the data

```
> rm(list=ls())
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
> # help(HolzingerSwineford1939)
> hs = HolzingerSwineford1939
> hs = subset(hs,school=='Grant-White'); dim(hs) # 145 rows, 15 columns
[1] 145 15
> print(head(hs),digits=3)
      id sex ageyr agemo      school grade  x1  x2  x3  x4  x5  x6  x7  x8  x9
157 201  1   13     0 Grant-White   7 3.83 4.75 0.50 3.33 4.25 1.43 3.00 4.10 4.33
158 202  2   11    10 Grant-White   7 5.50 5.50 2.12 2.67 4.25 1.43 2.83 4.90 5.42
```

159	203	1	12	6	Grant-White	7	5.67	6.00	2.75	3.67	4.75	2.71	2.17	4.30	6.33
160	204	1	11	11	Grant-White	7	4.83	5.75	1.12	3.00	4.75	1.57	4.96	5.15	4.00
161	205	1	12	5	Grant-White	7	2.67	6.25	1.25	2.67	6.25	3.43	4.87	6.10	4.44
162	206	2	12	6	Grant-White	7	5.00	6.25	2.50	3.33	5.75	2.57	4.09	5.65	5.58

Standardized factors, complete model specification First, a model will be specified using the full lavaan syntax, giving names to all the parameters. This is the way it was done in Chapters 0 and 1. It will be seen presently that there is an easier way to get the job done.

```
> swine1 = '
      # Measurement model
+      visual  =~ lambda1*x1 + lambda2*x2 + lambda3*x3
+      verbal  =~ lambda4*x4 + lambda5*x5 + lambda6*x6
+      speed   =~ lambda7*x7 + lambda8*x8 + lambda9*x9
+      # Variances of error terms
+      x1 ~~ omega1*x1; x2 ~~ omega2*x2; x3 ~~ omega3*x3
+      x4 ~~ omega4*x4; x5 ~~ omega5*x5; x6 ~~ omega6*x6
+      x7 ~~ omega7*x7; x8 ~~ omega8*x8; x9 ~~ omega9*x9
+      # Variances of factors equal one
+      visual  ~~ 1*visual; verbal  ~~ 1*verbal ; speed  ~~ 1*speed
+      # Covariances of factors
+      visual  ~~ phi12*verbal ; visual  ~~ phi13*speed
+                        verbal  ~~ phi23*speed
+
> smodel1 = lavaan(swine1, data=hs); summary(smodel1)
lavaan 0.6-7 ended normally after 19 iterations
```

Estimator	ML
Optimization method	NLMINB
Number of free parameters	21
Number of observations	145

Model Test User Model:

Test statistic	51.542
Degrees of freedom	24
P-value (Chi-square)	0.001

Parameter Estimates:

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

Latent Variables:

		Estimate	Std.Err	z-value	P(> z)
visual =~					
x1	(lmb1)	0.777	0.103	7.525	0.000
x2	(lmb2)	0.572	0.101	5.642	0.000
x3	(lmb3)	0.719	0.093	7.711	0.000
verbal =~					
x4	(lmb4)	0.971	0.079	12.355	0.000
x5	(lmb5)	0.961	0.083	11.630	0.000
x6	(lmb6)	0.935	0.081	11.572	0.000
speed =~					
x7	(lmb7)	0.679	0.087	7.819	0.000
x8	(lmb8)	0.833	0.087	9.568	0.000
x9	(lmb9)	0.719	0.086	8.357	0.000
Covariances:					
		Estimate	Std.Err	z-value	P(> z)
visual ~~					
verbal	(ph12)	0.541	0.085	6.355	0.000
speed	(ph13)	0.523	0.094	5.562	0.000
verbal ~~					
speed	(ph23)	0.336	0.091	3.674	0.000
Variances:					
		Estimate	Std.Err	z-value	P(> z)
.x1	(omg1)	0.715	0.126	5.675	0.000
.x2	(omg2)	0.899	0.123	7.339	0.000
.x3	(omg3)	0.557	0.103	5.409	0.000
.x4	(omg4)	0.315	0.065	4.870	0.000
.x5	(omg5)	0.419	0.072	5.812	0.000
.x6	(omg6)	0.406	0.069	5.880	0.000
.x7	(omg7)	0.600	0.091	6.584	0.000
.x8	(omg8)	0.401	0.094	4.248	0.000
.x9	(omg9)	0.535	0.089	6.010	0.000
visual		1.000			
verbal		1.000			
speed		1.000			

The output is pretty much self-explanatory to a reader who is familiar with the lavaan examples in Chapters 0 and 1. The estimated covariances (correlations) of the factors match Jöreskog's (1969, p. 192) values for Model (d). The factor loadings do not match, because Jöreskog standardizes the observed variables; we have not done that yet. The chi-squared test for model fit ($\chi^2(24) = 51.542$, $df = 21$, $p = 0.001$) indicates that the model is not fully compatible with the data¹⁷. A model based on the [reference variable](#)

¹⁷Jöreskog's value for the test of fit is 51.19. That's close, but not quite equal to the lavaan value. The reason is that in the formula for the likelihood ratio test test statistic (see Expression (1.18) on page 171) Jöreskog has a multiplier of $n - 1$ out in front, in place of n . This makes no difference asymptotically, of

`rule` performs better; we will get to that later.

The `cfa` function The same job can be accomplished with less work, using lavaan's `cfa` (confirmatory factor analysis) function with the default settings. Only the measurement part of the model needs to be given, and all the Greek letters are gone. There is a lot less typing. There are also fewer opportunities to make mistakes.

```
> swine2 = 'visual =~ x1 + x2 + x3
+          verbal =~ x4 + x5 + x6
+          speed  =~ x7 + x8 + x9
+          '
> smodel2 = cfa(swine2, data=hs); summary(smodel2)
lavaan 0.6-7 ended normally after 34 iterations
```

Estimator	ML
Optimization method	NLMINB
Number of free parameters	21
Number of observations	145

Model Test User Model:

Test statistic	51.542
Degrees of freedom	24
P-value (Chi-square)	0.001

Parameter Estimates:

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

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)
visual =~				
x1	1.000			
x2	0.736	0.155	4.760	0.000
x3	0.925	0.166	5.584	0.000
verbal =~				
x4	1.000			
x5	0.990	0.087	11.418	0.000
x6	0.963	0.085	11.377	0.000

course. To obtain Jöreskog's value from the lavaan output, $(n-1)/nG^2 = 144/145 * 51.542 = 51.18654$.

Jöreskog's likelihood approach is based on a Wishart distribution for a version of the sample covariance matrix with $n-1$ in the denominator. Some software, including SAS and Amos, follow Jöreskog's old LISREL software in this matter. Both lavaan and mplus, like this book, assume a multivariate normal likelihood for the original data, rather than starting with the covariance matrix.

```

speed =~
  x7          1.000
  x8          1.226    0.187    6.569    0.000
  x9          1.058    0.165    6.429    0.000

Covariances:
              Estimate Std.Err  z-value  P(>|z|)
visual ~~
  verbal      0.408    0.098    4.153    0.000
  speed       0.276    0.076    3.639    0.000
verbal ~~
  speed       0.222    0.073    3.022    0.003

Variances:
              Estimate Std.Err  z-value  P(>|z|)
.x1          0.715    0.126    5.675    0.000
.x2          0.899    0.123    7.339    0.000
.x3          0.557    0.103    5.409    0.000
.x4          0.315    0.065    4.870    0.000
.x5          0.419    0.072    5.812    0.000
.x6          0.406    0.069    5.880    0.000
.x7          0.600    0.091    6.584    0.000
.x8          0.401    0.094    4.248    0.000
.x9          0.535    0.089    6.010    0.000
visual       0.604    0.160    3.762    0.000
verbal       0.942    0.152    6.177    0.000
speed        0.461    0.118    3.910    0.000

```

Let's take a close look to see what we have. The number of free parameters equals 21, as in the summary of `smodel1`. The chi-squared statistics for model fit are the same. This is promising.

Now compare the estimated factor loadings under **Latent Variables** in the summaries of `smodel1` and `smodel2`. The numbers are different, but don't worry about that yet. The abbreviations for the parameter names are missing for `smodel2`; this is really no great loss. The output is quite readable if you understand `=~` as standing for "is measured by." Under **Variances**, note that when an observable variable is preceded by a dot, it means this is the estimated variance not of the variable, but of its error term. Comparison with the `smodel1` summary, which has labels, helps to confirm this. Once you get used to lavaan output, the parameter labels are really not necessary. If you wish, you can compromise by supplying names for just some of the parameters. This can be a convenient way to set two parameters equal; just give them the same name.

In the summary of `smodel2`, the estimated factor loadings for x_1 , x_4 and x_7 are all equal to one, and there are no standard errors or tests. By default, lavaan is fitting a surrogate model with a factor loading set to one for each factor; that's the model described as "Model One" in Section 3.3.1. The factor loadings of one could be mysterious for users who don't know about parameter identifiability, but lavaan is making a choice that's

designed to be helpful. It probably *is* helpful, most of the time.

Just to confirm the meaning of the parameter estimates, recall that under Model One, the factor loading for x_2 is $\lambda'_2 = \lambda_2/\lambda_1$. Under Model Two,

$$\begin{aligned}\frac{\lambda''_2}{\lambda''_1} &= \frac{\lambda_2\sqrt{\phi_{11}}}{\lambda_1\sqrt{\phi_{11}}} \\ &= \frac{\lambda_2}{\lambda_1} = \lambda'_2.\end{aligned}$$

By invariance, this equality must also be true of the MLEs. This means that $\widehat{\lambda}'_2 = 0.736$ (the factor loading for x_2 in the `smodel2` summary) can be recovered from the `smodel1` output, as follows. $\widehat{\lambda}''_2/\widehat{\lambda}''_1 =$

```
> 0.572/0.777
0.7361647
```

We are on the right track. It is clear that the default model fit in `smodel2` is the surrogate model in which a factor loading has been set to one for each factor.

Equal diagonals As given in (3.31), the main diagonal of the reproduced covariance matrix $\Sigma(\widehat{\theta})$ is able to match the main diagonal of the sample covariance matrix.

```
> # Checking that the diagonal of Sigma(thetahat) = diagonal of Sigmahat
> x = hs[,7:15]; n = dim(x)[1]
> Sigmahat = (n-1)/n * var(x)
> SigOfThetahat = fitted(smodel2)$cov
> rbind(diag(Sigmahat),diag(SigOfThetahat))
      x1      x2      x3      x4      x5      x6      x7      x8      x9
[1,] 1.318647 1.226379 1.073365 1.257212 1.341665 1.280142 1.0618 1.09438 1.051048
[2,] 1.318647 1.226379 1.073365 1.257212 1.341665 1.280142 1.0618 1.09438 1.051048
```

It worked perfectly, as it does in all but the most peculiar models. Even when they fit badly overall, confirmatory factor analysis models almost always fit the diagonal of $\widehat{\Sigma}$ perfectly.

Standardized parameter estimates One often encounters this expression in write-ups of confirmatory factor analysis and structural equation modelling. It's a bit misleading, because it's not the parameter estimates that are standardized. The statistics in question are parameter estimates for a model with the factors standardized — or both the factors and the observed variables standardized. The easiest way to get these numbers from lavaan is by adding the `standardized=TRUE` option to `summary`. Notice that the model does not to be re-fit.

```
> summary(smodel2, standardized=TRUE)
lavaan 0.6-7 ended normally after 34 iterations
```

Estimator	ML
Optimization method	NLMINB
Number of free parameters	21
Number of observations	145

Model Test User Model:

Test statistic	51.542
Degrees of freedom	24
P-value (Chi-square)	0.001

Parameter Estimates:

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

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
visual =~						
x1	1.000				0.777	0.677
x2	0.736	0.155	4.760	0.000	0.572	0.517
x3	0.925	0.166	5.584	0.000	0.719	0.694
verbal =~						
x4	1.000				0.971	0.866
x5	0.990	0.087	11.418	0.000	0.961	0.829
x6	0.963	0.085	11.377	0.000	0.935	0.826
speed =~						
x7	1.000				0.679	0.659
x8	1.226	0.187	6.569	0.000	0.833	0.796
x9	1.058	0.165	6.429	0.000	0.719	0.701

Covariances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
visual ~~						
verbal	0.408	0.098	4.153	0.000	0.541	0.541
speed	0.276	0.076	3.639	0.000	0.523	0.523
verbal ~~						
speed	0.222	0.073	3.022	0.003	0.336	0.336

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.x1	0.715	0.126	5.675	0.000	0.715	0.542
.x2	0.899	0.123	7.339	0.000	0.899	0.733
.x3	0.557	0.103	5.409	0.000	0.557	0.519

.x4	0.315	0.065	4.870	0.000	0.315	0.251
.x5	0.419	0.072	5.812	0.000	0.419	0.312
.x6	0.406	0.069	5.880	0.000	0.406	0.317
.x7	0.600	0.091	6.584	0.000	0.600	0.566
.x8	0.401	0.094	4.248	0.000	0.401	0.367
.x9	0.535	0.089	6.010	0.000	0.535	0.509
visual	0.604	0.160	3.762	0.000	1.000	1.000
verbal	0.942	0.152	6.177	0.000	1.000	1.000
speed	0.461	0.118	3.910	0.000	1.000	1.000

The `standardized=TRUE` option has added two columns to the `smodel2` summary output: `Std.lv` and `Std.all`. Naturally, `Std.lv` means that the latent variables (factors) have been standardized. These numbers perfectly match the `Estimate` column of the `smodel1` summary. The `Std.all` column gives estimates for a model where the observable variables as well as the latent variables are standardized. This is sometimes called the “completely standardized” model.

This time, the estimated factor loadings as well as the correlations between factors match Jöreskog’s (1969, p. 192) “(d) Restricted Oblique Solution” [36]. This confirms that the `Std.all` values are what we think they are – estimates for a model in which both the factors and the observed variables have been standardized.

Producing the numbers with matrix operations It is instructive to see how the `Std.lv` and `Std.all` values could have been obtained¹⁸ from the `smodel2` model fit. In the notation of Section 3.3.1, we are calculating double and triple-prime matrices from single-prime matrices.

First consider `Std.lv`. An application of the invariance principle to (3.22), (3.23) and (3.24) yields

$$\begin{aligned}
 \widehat{\Lambda}_1'' &= dg(\widehat{\Phi}')^{1/2} \\
 \widehat{\Lambda}_2'' &= \widehat{\Lambda}_2' dg(\widehat{\Phi}')^{1/2} \\
 \widehat{\Phi}'' &= dg(\widehat{\Phi}')^{-1/2} \widehat{\Phi}' dg(\widehat{\Phi}')^{-1/2}.
 \end{aligned}
 \tag{3.33}$$

How can one obtain those single-prime matrices? The parameter estimates in matrix form are located in “slots” in the fitted lavaan model object. Slots are like properties of the object, or something. There can be slots within slots. One can refer to a slot of an object using the `@` sign, as in `object@slotname`. In the object `smodel2`, the slot called `Model` is an object with 59 slots. One of these is named `GLIST`; it is a list containing the estimated parameter matrices we want. In the following, single primes are represented by `_p`, and double primes are represented by `_pp`. After some experimenting,

```
> Lambda_p = (smodel2@Model)@GLIST$lambda; Lambda_p
      [,1]      [,2]      [,3]
```

¹⁸In the lavaan code, it may be done in a slightly different but equivalent way. I have not looked at the source code.


```
[1,] 1.0000000 0.0000000 0.0000000
[2,] 0.7361563 0.0000000 0.0000000
[3,] 0.9247953 0.0000000 0.0000000
[4,] 0.0000000 1.0000000 0.0000000
[5,] 0.0000000 0.9897921 0.0000000
[6,] 0.0000000 0.9633398 0.0000000
[7,] 0.0000000 0.0000000 1.0000000
[8,] 0.0000000 0.0000000 1.225840
[9,] 0.0000000 0.0000000 1.057888
```

Comparing with the numbers in the `smodel2` summary, this is definitely $\widehat{\Lambda}'$. Of course, it doesn't have the observed variables in \mathbf{d}_1 and \mathbf{d}_2 separated. Extracting $\widehat{\Lambda}'_2$ and then the other estimated parameter matrices,

```
> Lambda2_p = Lambda_p[-c(1,4,7),]; Lambda2_p
      [,1]      [,2]      [,3]
[1,] 0.7361563 0.0000000 0.0000000
[2,] 0.9247953 0.0000000 0.0000000
[3,] 0.0000000 0.9897921 0.0000000
[4,] 0.0000000 0.9633398 0.0000000
[5,] 0.0000000 0.0000000 1.225840
[6,] 0.0000000 0.0000000 1.057888
> Omega = (smodel2@Model)@GLIST$theta; Omega
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]
[1,] 0.7148977 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000 0.0000000 0.0000000
[2,] 0.0000000 0.8991918 0.0000000 0.0000000 0.0000000 0.000000 0.0000000 0.0000000 0.0000000
[3,] 0.0000000 0.0000000 0.5570105 0.0000000 0.0000000 0.000000 0.0000000 0.0000000 0.0000000
[4,] 0.0000000 0.0000000 0.0000000 0.3153055 0.0000000 0.000000 0.0000000 0.0000000 0.0000000
[5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.4188895 0.000000 0.0000000 0.0000000 0.0000000
[6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.40603 0.0000000 0.0000000 0.0000000
[7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.6004945 0.0000000 0.0000000
[8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000 0.4011844 0.0000000
[9,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000 0.0000000 0.534789
> Phi_p = (smodel2@Model)@GLIST$psi; Phi_p
      [,1]      [,2]      [,3]
[1,] 0.6037495 0.4077212 0.2761904
[2,] 0.4077212 0.9419068 0.2215664
[3,] 0.2761904 0.2215664 0.4613051
```

Notice that except for Λ , lavaan is using a different Greek letter notation for the parameter matrices. Nobody cares.

The matrix $dg(\widehat{\Phi})^{1/2}$ appears four times in (3.33). To carry out the calculations, it is convenient to give it a simple name. Call it M .

```
> M = sqrt(diag(diag(Phi_p))); M # = Lambda1_pp, the factor loadings of the leading reference
      [,1]      [,2]      [,3]
[1,] 0.7770132 0.0000000 0.0000000
```

```

[2,] 0.0000000 0.9705188 0.0000000
[3,] 0.0000000 0.0000000 0.6791945
> Lambda2_pp = Lambda2_p %*% M; Lambda2_pp # the other factor loadings
      [,1]      [,2]      [,3]
[1,] 0.5720032 0.0000000 0.0000000
[2,] 0.7185782 0.0000000 0.0000000
[3,] 0.0000000 0.9606119 0.0000000
[4,] 0.0000000 0.9349394 0.0000000
[5,] 0.0000000 0.0000000 0.8325836
[6,] 0.0000000 0.0000000 0.7185117
> # Putting the full factor matrix Lambda_pp together,
> Lambda_pp = rbind(M[1,],Lambda2_pp[1:2,], M[2,],Lambda2_pp[3:4,], M[3,],Lambda2_pp[5:6,])
> Lambda_pp
      [,1]      [,2]      [,3]
[1,] 0.7770132 0.0000000 0.0000000
[2,] 0.5720032 0.0000000 0.0000000
[3,] 0.7185782 0.0000000 0.0000000
[4,] 0.0000000 0.9705188 0.0000000
[5,] 0.0000000 0.9606119 0.0000000
[6,] 0.0000000 0.9349394 0.0000000
[7,] 0.0000000 0.0000000 0.6791945
[8,] 0.0000000 0.0000000 0.8325836
[9,] 0.0000000 0.0000000 0.7185117

```

These numbers match the estimated factor loadings in the `Std.lv` column of the `smodel2` model summary. For example, the loading of x_8 on the `speed` factor is 0.833 in the `Std.lv` column, and it is 0.8325836 in the matrix `Lambda_pp` above.

```

> Phi_pp = solve(M) %*% Phi_p %*% solve(M); Phi_pp
      [,1]      [,2]      [,3]
[1,] 1.0000000 0.5406683 0.5233425
[2,] 0.5406683 1.0000000 0.3361288
[3,] 0.5233425 0.3361288 1.0000000

```

These estimated correlations match the contents of the `Std.lv` column under `Covariances`.

For the “completely standardized” estimates in the `Std.all` column — that is, for the estimates from a model with both the factors and the observed variables standardized, Expression (3.28) implies

$$\begin{aligned}\widehat{\Lambda}''' &= \widehat{\mathbf{W}}^{-1/2} \widehat{\Lambda}'' \\ \widehat{\Omega}''' &= \widehat{\mathbf{W}}^{-1/2} \widehat{\Omega} \widehat{\mathbf{W}}^{-1/2}.\end{aligned}$$

Calculating,

```

> # Standardized observed variables
> n = dim(hs)[1]; n
[1] 145

```

```

> Sigma = var(hs[,7:15]) * (n-1)/n # Actually Sigma-hat of course
> W = diag(diag(SigmaHat))
> Lambda_ppp = solve(sqrt(W)) %*% Lambda_pp; Lambda_ppp
      [,1]      [,2]      [,3]
[1,] 0.6766500 0.0000000 0.0000000
[2,] 0.5165187 0.0000000 0.0000000
[3,] 0.6935860 0.0000000 0.0000000
[4,] 0.0000000 0.8655649 0.0000000
[5,] 0.0000000 0.8293273 0.0000000
[6,] 0.0000000 0.8263318 0.0000000
[7,] 0.0000000 0.0000000 0.6591327
[8,] 0.0000000 0.0000000 0.7958731
[9,] 0.0000000 0.0000000 0.7008460
> Omega_ppp = solve(sqrt(W)) %*% Omega %*% solve(sqrt(W)); Omega_ppp
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]
[1,] 0.5421448 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[2,] 0.0000000 0.7332085 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[3,] 0.0000000 0.0000000 0.5189386 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[4,] 0.0000000 0.0000000 0.0000000 0.2507973 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.3122162 0.0000000 0.0000000 0.0000000 0.0000000
[6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.3171758 0.0000000 0.0000000 0.0000000
[7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.5655441 0.0000000 0.0000000
[8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.3665861 0.0000000
[9,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.508815

```

The numbers in these matrices match the `Std.all` output. For example, the estimated factor loading of z_2 on `visual` is 0.517 in the `Std.all` column, and it is 0.5165187 in `Lambda_ppp` above. The estimated variance of e_6''' is 0.317 in the `Std.all` column, and 0.3171758 in `Omega_ppp` — so, the x_2 variable is estimated to be around 32% noise.

It goes without saying (and yet I find myself saying it anyway) that in a practical data analysis job, these matrix calculations would almost never be necessary. It's much easier to just use `standardized=TRUE`, and let `lavaan` do the work. The purpose of doing the matrix calculations here was to show where those `Std.lv` and `Std.all` numbers come from, and to provide a bridge between the theory and what the software is producing. It's also nice to know how to get at those estimated parameter matrices. I didn't know about slots before I did this.

Fitting a model with standardized factors the easy way The main disadvantage of the `standardized=TRUE` option is that one gets estimates, but not standard errors. In the summary output for `smodel2`, the standard errors in the output still apply to the surrogate model in which some loadings were set to one. So for example, if you wanted a confidence interval for a correlation between factors, you would still have a bit of work to do. It is possible to choose a model with standardized factors at the model fitting stage, by using the `std.lv` (standardized latent variables) option, as follows.

```

> smodel3 = cfa(swine2, data=hs, std.lv=TRUE); summary(smodel3, standardized=TRUE)

```

lavaan 0.6-7 ended normally after 19 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	21
Number of observations	145

Model Test User Model:

Test statistic	51.542
Degrees of freedom	24
P-value (Chi-square)	0.001

Parameter Estimates:

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

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
visual =~						
x1	0.777	0.103	7.525	0.000	0.777	0.677
x2	0.572	0.101	5.642	0.000	0.572	0.517
x3	0.719	0.093	7.711	0.000	0.719	0.694
verbal =~						
x4	0.971	0.079	12.355	0.000	0.971	0.866
x5	0.961	0.083	11.630	0.000	0.961	0.829
x6	0.935	0.081	11.572	0.000	0.935	0.826
speed =~						
x7	0.679	0.087	7.819	0.000	0.679	0.659
x8	0.833	0.087	9.568	0.000	0.833	0.796
x9	0.719	0.086	8.357	0.000	0.719	0.701

Covariances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
visual ~~						
verbal	0.541	0.085	6.355	0.000	0.541	0.541
speed	0.523	0.094	5.562	0.000	0.523	0.523
verbal ~~						
speed	0.336	0.091	3.674	0.000	0.336	0.336

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.x1	0.715	0.126	5.675	0.000	0.715	0.542

.x2	0.899	0.123	7.339	0.000	0.899	0.733
.x3	0.557	0.103	5.409	0.000	0.557	0.519
.x4	0.315	0.065	4.870	0.000	0.315	0.251
.x5	0.419	0.072	5.812	0.000	0.419	0.312
.x6	0.406	0.069	5.880	0.000	0.406	0.317
.x7	0.600	0.091	6.584	0.000	0.600	0.566
.x8	0.401	0.094	4.248	0.000	0.401	0.367
.x9	0.535	0.089	6.010	0.000	0.535	0.509
visual	1.000				1.000	1.000
verbal	1.000				1.000	1.000
speed	1.000				1.000	1.000

The `Estimate` column now matches the `Std.lv` column; `std.lv=TRUE` had its intended effect. All is well.

Analyzing the correlation matrix There is also a `std.ov` option for the `cfa` function, and one would think the observed variables could be standardized by specifying `std.ov=TRUE`. However, as of this writing¹⁹ it does not quite work as expected. When the observed variables are standardized, they are divided by a sample standard deviation with $n - 1$ in the denominator, rather than n . This makes no difference asymptotically, so the estimates, tests and confidence intervals are just as good either way. However, if `std.lv` and `std.ov` are both `TRUE`, the numbers in the `Estimate` column don't quite don't quite match the `Std.all` column. It's a bit unsettling.

One option is to standardize the observed variables yourself, making sure you divide by n to get the true MLEs. This works, but it's awkward. It's easier to use the sample correlation matrix as input.

The lavaan software allows a sample covariance matrix and a sample size as input, in place of the raw data²⁰. When you give lavaan a correlation matrix, it treats it as a sample covariance matrix. There are two consequences, both a bit subtle. The first is that lavaan assumes the sample variances just happen to be all equal to one (an event of zero probability, by the way), and it treats the error variances (the $\omega_{j,j}'''$) as free parameters to be estimated, rather than using the fact that they are functions of the other parameters. This actually works out very well. The standard errors are correct, and it's a lot easier to obtain confidence intervals for uniquenesses and commonalities than it would be otherwise.

The second consequence of treating the correlation matrix as a covariance matrix is the issue of whether the sample variances and covariances have n in the denominator, or $n - 1$. Actually, this question should not apply to sample correlations. Thinking of a sample correlation as a sample covariance divided by a product of sample standard deviations, the denominators, whether they are n or $n - 1$, are already cancelled. To lavaan, though, it's just a sample covariance matrix. Many sample covariance matrices (for example, the ones produced by R's `var` function) have $n - 1$ in the denominators,

¹⁹September 2021, lavaan version 0.6-7.

²⁰This is really handy for re-analyzing published data, because books and journal articles often display covariance matrices or correlation matrices even when they do not provide access to the raw data.

so that the estimates are unbiased, but not quite true MLEs. The `cfa` function has an option to deal with this. By default, `sample.cov.rescale` is set to `TRUE`, meaning please correct the input sample covariance matrix, multiplying all entries by $(n-1)/n$. If the input matrix is a sample correlation matrix, you want `sample.cov.rescale=FALSE`. Here's how it goes with the Holzinger-Swineford data. I think factor analysis is nicer with standardized observed variables, so to me, this is a good example of how to do a confirmatory factor analysis with lavaan²¹.

```
> # Analyze the correlation matrix
> x = hs[,7:15] # Columns 7 through 15 of the data frame: Just the x variables.
> xcorr = cor(x); round(xcorr,3)
      x1    x2    x3    x4    x5    x6    x7    x8    x9
x1 1.000 0.326 0.449 0.342 0.309 0.317 0.104 0.308 0.487
x2 0.326 1.000 0.417 0.228 0.159 0.195 0.066 0.168 0.248
x3 0.449 0.417 1.000 0.328 0.287 0.347 0.075 0.239 0.373
x4 0.342 0.228 0.328 1.000 0.719 0.714 0.209 0.104 0.314
x5 0.309 0.159 0.287 0.719 1.000 0.685 0.254 0.198 0.356
x6 0.317 0.195 0.347 0.714 0.685 1.000 0.179 0.121 0.272
x7 0.104 0.066 0.075 0.209 0.254 0.179 1.000 0.587 0.418
x8 0.308 0.168 0.239 0.104 0.198 0.121 0.587 1.000 0.528
x9 0.487 0.248 0.373 0.314 0.356 0.272 0.418 0.528 1.000
> smodel4 = cfa(swine2, sample.cov=xcorr, sample.nobs=145,
+ std.lv=TRUE, sample.cov.rescale=FALSE)
> summary(smodel4, standardized=TRUE)
lavaan 0.6-7 ended normally after 20 iterations
```

Estimator	ML
Optimization method	NLMINB
Number of free parameters	21
Number of observations	145

Model Test User Model:

Test statistic	51.542
Degrees of freedom	24
P-value (Chi-square)	0.001

Parameter Estimates:

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

Latent Variables:

²¹The reader may be thinking “Well, it’s about time!”

```

                Estimate Std.Err  z-value  P(>|z|)  Std.lv  Std.all
visual =~
  x1            0.677    0.090    7.525    0.000    0.677    0.677
  x2            0.517    0.092    5.642    0.000    0.517    0.517
  x3            0.694    0.090    7.711    0.000    0.694    0.694
verbal =~
  x4            0.866    0.070   12.355    0.000    0.866    0.866
  x5            0.829    0.071   11.630    0.000    0.829    0.829
  x6            0.826    0.071   11.572    0.000    0.826    0.826
speed =~
  x7            0.659    0.084    7.819    0.000    0.659    0.659
  x8            0.796    0.083    9.568    0.000    0.796    0.796
  x9            0.701    0.084    8.357    0.000    0.701    0.701

Covariances:
                Estimate Std.Err  z-value  P(>|z|)  Std.lv  Std.all
visual ~~
  verbal        0.541    0.085    6.355    0.000    0.541    0.541
  speed        0.523    0.094    5.562    0.000    0.523    0.523
verbal ~~
  speed        0.336    0.091    3.674    0.000    0.336    0.336

Variances:
                Estimate Std.Err  z-value  P(>|z|)  Std.lv  Std.all
.x1            0.542    0.096    5.675    0.000    0.542    0.542
.x2            0.733    0.100    7.339    0.000    0.733    0.733
.x3            0.519    0.096    5.409    0.000    0.519    0.519
.x4            0.251    0.051    4.870    0.000    0.251    0.251
.x5            0.312    0.054    5.812    0.000    0.312    0.312
.x6            0.317    0.054    5.880    0.000    0.317    0.317
.x7            0.566    0.086    6.584    0.000    0.566    0.566
.x8            0.367    0.086    4.248    0.000    0.367    0.367
.x9            0.509    0.085    6.010    0.000    0.509    0.509
  visual        1.000                    1.000    1.000
  verbal        1.000                    1.000    1.000
  speed        1.000                    1.000    1.000
# That's it!

```

There we go. The `Estimate` and `Std.lv` both match `Std.all`. These numbers are very interpretable. For example, the estimate of 0.733 for $\omega_{2,2}$ (triple prime deleted) means we estimate that x_2 is around 73.3% noise. Because maximum likelihood estimates are asymptotically normal, an approximate 95% confidence interval to go with this estimate is just the estimate plus or minus 1.96 times the standard error.

```

> c(0.733-1.96*0.1, 0.733+1.96*0.1)
[1] 0.537 0.929

```

This confidence interval is produced automatically by `parameterEstimates(smodel14)`.

Also, notice that in obedience to Theorem 3.1, the chi-squared statistic for lack of model fit is still 51.542. It is unaffected by standardizing the observed variables.

A confidence interval for uniqueness, the hard way Suppose you do not standardize the observed variables, and you want a point estimate and confidence interval for the uniqueness of x_2 — *under the original model*. Assume the lavaan model `swine1` on page 333, the very explicit model with standardized factors. Under this model (with double primes, as in Section 3.3.1),

$$\begin{aligned} \text{Var}(x_2) &= \text{Var}(\lambda_2'' F_1'' + e_2) \\ &= \lambda_2''^2 \text{Var}(F_1'') + \omega_2 \\ &= \lambda_2''^2 \phi_{1,1}'' + \omega_2. \end{aligned}$$

Bearing in mind that $\lambda_2'' = \lambda_2 \phi_{1,1}^{1/2}$ and $\phi_{1,1}'' = 1$, the proportion of unexplained variance under the surrogate model is

$$\begin{aligned} \frac{\omega_2}{\lambda_2''^2 \phi_{1,1}'' + \omega_2} &= \frac{\omega_2}{\left(\lambda_2 \phi_{1,1}^{1/2}\right)^2 \times 1 + \omega_2} \\ &= \frac{\omega_2}{\lambda_2^2 \phi_{1,1} + \omega_2}. \end{aligned} \tag{3.34}$$

For the centered original model, $\text{Var}(x_2) = \lambda_2^2 \phi_{1,1} + \omega_2$, so that the proportion of unexplained variance is exactly Expression (3.34). Remarkably, this identifiable function of the original model parameters is the same under the surrogate model with standardized factors. It's not the kind of thing you can depend on in general.

In any case, we want a point estimate and confidence interval for $\frac{\omega_2}{\lambda_2''^2 + \omega_2}$. The point estimate can be easily obtained from numbers in the output of `summary(smodel1)`. On a test or quiz, you could do it with a calculator.

```
> 0.899/(0.572^2 + 0.899)
[1] 0.7331689
```

That's exactly the $\hat{\omega}_2$ of 0.733 from the model with both factors and observed variables standardized. We also want a confidence interval, something that cannot be calculated from the `summary(smodel1)` output.

As a nice smooth function of asymptotically normal MLEs, $\frac{\hat{\omega}_2}{\hat{\lambda}_2''^2 + \hat{\omega}_2}$ is asymptotically normal. All we need is a standard error — an estimated standard deviation. We can get it easily using lavaan's `:=` syntax for estimating non-linear functions of model parameters. Include this line at the end of the `swine1` model string:

```
x2uniqueness := omega2 / (lambda2^2 + omega2)
```

The lovely `:=` feature is only available if you explicitly provide names (labels) for the parameters. It is incompatible with the shorthand syntax of model `smodel2`.

One is tempted to just copy-paste the entire `swine1` string, and add the line at the end and call the result `swine1b` or something. This is not ideal, because it's not a good idea to have more than one slightly different version of the same code floating around. What if you found an error in `swine1`, or decided to change it for some other reason? Would you remember to make the same change(s) in `swine1b`? Here's a better option.

```
> swine1b = paste(swine1, "x2uniqueness := omega2 / (lambda2^2 + omega2)")
```

Take a look at the result.

```
> cat(swine1b)
# Measurement model
visual =~ lambda1*x1 + lambda2*x2 + lambda3*x3
verbal =~ lambda4*x4 + lambda5*x5 + lambda6*x6
speed =~ lambda7*x7 + lambda8*x8 + lambda9*x9
# Variances of error terms
x1 ~~ omega1*x1; x2 ~~ omega2*x2; x3 ~~ omega3*x3
x4 ~~ omega4*x4; x5 ~~ omega5*x5; x6 ~~ omega6*x6
x7 ~~ omega7*x7; x8 ~~ omega8*x8; x9 ~~ omega9*x9
# Variances of factors equal one
visual ~~ 1*visual; verbal ~~ 1*verbal; speed ~~ 1*speed
# Covariances of factors
visual ~~ phi12*verbal; visual ~~ phi13*speed
                    verbal ~~ phi23*speed
x2uniqueness := omega2 / (lambda2^2 + omega2)
```

The non-linear function is added neatly to the end. If `swine1` changes, `swine1b` will also be changed when the code is re-run. Now fit the new model and look at the results.

```
> smodel1b = lavaan(swine1b, data=hs); summary(smodel1b)
```

The estimate and standard error for `x2uniqueness` appears at the end of the `summary` output. everything else is the same as the output of `summary(smodel1)`. Showing just the last part,

Defined Parameters:

	Estimate	Std.Err	z-value	P(> z)
x2uniqueness	0.733	0.081	9.078	0.000

The estimated uniqueness (73% noise) is exactly the same as the $\hat{\omega}_2'''$ obtained from `smodel14`, the model with both factors and observed variables standardized. The standard errors are a bit different, 0.081 for `x2uniqueness`, versus 0.10 for $\hat{\omega}_2'''$ in `smodel14`. The reason is that by default, lavaan uses the multivariate delta method (See Appendix A, page 564) to estimate the standard deviations of non-linear functions of the parameter estimates. These numbers are close to the ones that come from re-parameterization, in the sense that the difference goes to zero in probability as the sample size tends to infinity. They need not be the same for finite sample sizes, but they are equally valid.

Another option is to use the `se = "bootstrap"` option in the `cfa` or `lavaan` function. This yields standard errors based on the bootstrap, which is distribution-free. Because the bootstrap is a randomization technique, the standard errors will be slightly different every time you run your code, unless you set the seed of the random number generator with the `set.seed` function.

A model that fits Let's not get too carried away here. We got the lavaan software to do what we want, but the model still does not fit ($\chi^2 = 51.542$, $df = 24$, $p = 0.001$). This means that the estimates, and especially the tests and confidence intervals, are open to question. Jöreskog's analysis in [36] includes several models that fit the data, including model (c), described as the "Reference variables solution." This is exactly the model of the [reference variable rule](#), except that it's a special case with the errors independent²².

Starting over for completeness,

```
> rm(list=ls())
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-7
lavaan is BETA software! Please report any bugs.
> hs = subset(HolzingerSwineford1939, school=='Grant-White')
> x = hs[,7:15]; xcorr = cor(x)
```

Now specify the reference variable model. The reference variables (x_1 , x_4 and x_7) are out front, while the other observed variables, which are influenced by all factors, are grouped together, identically in each line of the model string.

```
> swine3 = '
+         visual =~ x1 + x2+x3+x5+x6+x8+x9
+         verbal =~ x4 + x2+x3+x5+x6+x8+x9
+         speed  =~ x7 + x2+x3+x5+x6+x8+x9
+         '
```

Now fit the model, analyzing the correlation matrix because it is the easiest way to standardize the observed variables.

```
> smodel5 = cfa(swine3, sample.cov=xcorr, sample.nobs=145,
+               std.lv=TRUE, sample.cov.rescale=FALSE)
> summary(smodel5) # standardized=TRUE is not necessary.
lavaan 0.6-7 ended normally after 32 iterations
```

Estimator	ML
Optimization method	NLMINB
Number of free parameters	33

²²Actually, I discovered the reference variable rule by attempting to generalize Jöreskog's model (c). Others may have known about this rule, but I did not.

Number of observations 145

Model Test User Model:

Test statistic 9.846
 Degrees of freedom 12
 P-value (Chi-square) 0.629

Parameter Estimates:

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

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)
visual =~				
x1	0.708	0.087	8.144	0.000
x2	0.538	0.123	4.362	0.000
x3	0.674	0.125	5.392	0.000
x5	-0.033	0.093	-0.350	0.726
x6	0.013	0.092	0.137	0.891
x8	0.415	0.115	3.612	0.000
x9	0.557	0.113	4.916	0.000
verbal =~				
x4	0.871	0.070	12.434	0.000
x2	-0.031	0.118	-0.265	0.791
x3	0.042	0.119	0.354	0.724
x5	0.808	0.090	8.939	0.000
x6	0.819	0.090	9.055	0.000
x8	-0.298	0.111	-2.673	0.008
x9	-0.061	0.110	-0.552	0.581
speed =~				
x7	0.782	0.096	8.161	0.000
x2	-0.075	0.107	-0.700	0.484
x3	-0.086	0.108	-0.790	0.429
x5	0.128	0.074	1.729	0.084
x6	-0.007	0.075	-0.093	0.926
x8	0.731	0.109	6.690	0.000
x9	0.413	0.096	4.321	0.000

Covariances:

	Estimate	Std.Err	z-value	P(> z)
visual ~~				
verbal	0.543	0.112	4.850	0.000

speed	0.240	0.148	1.626	0.104
verbal ~~				
speed	0.284	0.116	2.439	0.015

Variances:

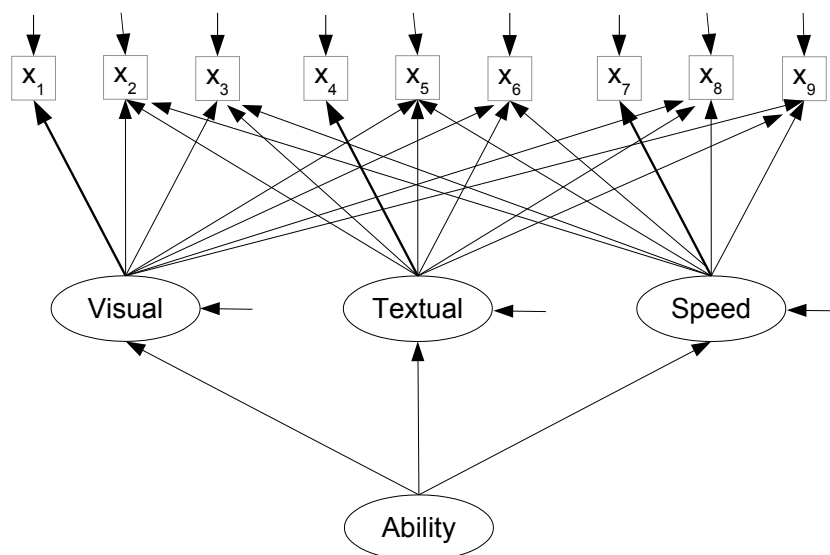
	Estimate	Std.Err	z-value	P(> z)
.x1	0.499	0.091	5.498	0.000
.x2	0.740	0.102	7.245	0.000
.x3	0.535	0.095	5.661	0.000
.x5	0.302	0.054	5.589	0.000
.x6	0.322	0.054	5.924	0.000
.x8	0.317	0.095	3.323	0.001
.x9	0.456	0.071	6.421	0.000
.x4	0.241	0.052	4.626	0.000
.x7	0.388	0.113	3.428	0.001
visual	1.000			
verbal	1.000			
speed	1.000			

This model fits ($\chi^2 = 9.846$, $df = 12$, $p = 0.629$). The estimated factor loadings (and the associated tests) suggest that the model of Figure 3.15 did not fit because x_8 (Counting Dots) and x_9 (Straight-curved Capitals) are positively influenced by the speed factor. There is also evidence that x_8 may be *negatively* influenced by the verbal factor F_2 .

It is interesting that under the model of Figure 3.15, the estimated correlation between the visual and speed factors is substantial ($\hat{\phi}_{1,3} = 0.523$) and undeniably significant ($z = 5.562$, $p \approx 0$). See for example the output of `summary(smodel4, standardized=TRUE)`. This is an important conclusion, because it might reflect something fundamental about cognition and the human nervous system. However, for the model that fits the data, there is not enough evidence to conclude a non-zero correlation ($\hat{\phi}_{1,3} = 0.24$, $z = 1.626$, $p = 0.104$): see the output of `summary(smodel5)` starting on page 349. When a model does not fit the data, conclusions from the significance tests are highly suspect. This issue is discussed in Chapter 7.

A second-order model It is fairly reasonable to hypothesize that there is a general factor underlying the visual, verbal and speed factors; it might be called mental ability. Figure 3.16 shows the path diagram. The top part is Jöreskog's [36] Reference Variables model c (also `smodel4`), identified by the [reference variable rule](#). In the lower part, the curved arrows representing correlations between factors have been replaced by the hypothesized second-order ability factor, with arrows pointing from the ability factor to the first-order visual, verbal and speed factors. There are also arrows that seem to come from nowhere, pointing at the first-order factors. These represent error terms. One might think that their variances would introduce three additional parameters, but because the factors are standardized (including ability), the variances are functions of the second-order factor loadings.

Figure 3.16: A second-order model for the Holzinger and Swineford Data



There are three of these second-order factor loadings. They replace the three correlations between first-order factors. Expression (3.13) on page 293 in the proof of the [three-variable rule](#) shows that there is a one-to-one connection between the second-order factor loadings and the correlations between first-order factors, provided that the sign of at least second-order loadings is known. Here, there is no problem; theoretically, they are all positive.

To incorporate the second-order ability factor into the model, it's enough to add a line that says ability is measured by visual, verbal and speed.

```
> swine4 = paste(swine3, "ability =~ visual + verbal + speed"); cat(swine4)

visual =~ x1 + x2+x3+x5+x6+x8+x9
verbal  =~ x4 + x2+x3+x5+x6+x8+x9
speed   =~ x7 + x2+x3+x5+x6+x8+x9
ability =~ visual + verbal + speed
```

For the sake of interpretability, I wanted to stay with a “completely standardized” model, in which both the observed and latent variables are standardized.

```
> smodel6 = cfa(swine4, sample.cov=xcorr, sample.nobs=145,
+               std.lv=TRUE, sample.cov.rescale=FALSE)
```

Before looking at any output, let's consider what to expect. First, since the parameters of `smodel5` and `smodel6` are one-to-one, the fit should be the same and we should get the same chi-squared value of 9.846 with 12 degrees of freedom. Second, all the estimated

first-order factor loadings (and consequently, the error variances) should be the same in the two fitted models. Third, the invariance principle of maximum likelihood estimation²³ dictates a very specific connection between the estimated second-order factor loadings and the estimated correlations between first-order factors. To make this explicit, denote the ability factor by F_0 , and write the second-order model equations as follows.

$$\begin{aligned} F_1 &= \gamma_1 F_0 + \epsilon_1 \\ F_2 &= \gamma_2 F_0 + \epsilon_2 \\ F_3 &= \gamma_3 F_0 + \epsilon_3 \end{aligned} \tag{3.35}$$

Then, basically transcribing material from 3.12 on page 292, we must have

$$\hat{\phi}_{1,2} = \hat{\gamma}_1 \hat{\gamma}_2 \quad \hat{\phi}_{1,3} = \hat{\gamma}_1 \hat{\gamma}_3 \quad \hat{\phi}_{2,3} = \hat{\gamma}_2 \hat{\gamma}_3, \tag{3.36}$$

where the $\hat{\phi}_{i,j}$ are from the `Covariances` part of the output from `summary(smodel5)`; the output begins on page 349.

Now we know what to expect from `summary(smodel6)`.

```
> summary(smodel6, standardized=TRUE)
```

```
lavaan 0.6-7 ended normally after 43 iterations
```

Estimator	ML
Optimization method	NLMINB
Number of free parameters	33
Number of observations	145

```
Model Test User Model:
```

Test statistic	9.846
Degrees of freedom	12
P-value (Chi-square)	0.629

```
Parameter Estimates:
```

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

```
Latent Variables:
```

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
visual =~						
x1	0.520	0.170	3.063	0.002	0.708	0.708
x2	0.396	0.127	3.116	0.002	0.538	0.538

²³Roughly, the MLE of a function is that function of the MLE.

x3	0.496	0.141	3.508	0.000	0.674	0.674
x5	-0.024	0.068	-0.355	0.723	-0.033	-0.033
x6	0.009	0.068	0.137	0.891	0.013	0.013
x8	0.305	0.127	2.397	0.017	0.415	0.415
x9	0.409	0.139	2.937	0.003	0.557	0.557
verbal =~						
x4	0.522	0.286	1.827	0.068	0.871	0.871
x2	-0.019	0.071	-0.264	0.792	-0.031	-0.031
x3	0.025	0.073	0.343	0.731	0.042	0.042
x5	0.484	0.264	1.832	0.067	0.808	0.808
x6	0.491	0.265	1.851	0.064	0.819	0.819
x8	-0.178	0.097	-1.847	0.065	-0.298	-0.298
x9	-0.036	0.064	-0.568	0.570	-0.061	-0.061
speed =~						
x7	0.731	0.105	6.971	0.000	0.782	0.782
x2	-0.070	0.099	-0.707	0.480	-0.075	-0.075
x3	-0.080	0.100	-0.803	0.422	-0.086	-0.086
x5	0.119	0.070	1.706	0.088	0.128	0.128
x6	-0.007	0.070	-0.093	0.926	-0.007	-0.007
x8	0.683	0.105	6.495	0.000	0.731	0.731
x9	0.386	0.096	4.031	0.000	0.413	0.413
ability =~						
visual	0.923	0.575	1.605	0.108	0.678	0.678
verbal	1.336	1.133	1.179	0.238	0.801	0.801
speed	0.379	0.182	2.085	0.037	0.355	0.355
Variances:						
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.x1	0.499	0.091	5.498	0.000	0.499	0.499
.x2	0.740	0.102	7.245	0.000	0.740	0.740
.x3	0.535	0.095	5.661	0.000	0.535	0.535
.x5	0.302	0.054	5.589	0.000	0.302	0.302
.x6	0.322	0.054	5.924	0.000	0.322	0.322
.x8	0.317	0.095	3.323	0.001	0.317	0.317
.x9	0.456	0.071	6.421	0.000	0.456	0.456
.x4	0.241	0.052	4.626	0.000	0.241	0.241
.x7	0.388	0.113	3.428	0.001	0.388	0.388
.visual	1.000				0.540	0.540
.verbal	1.000				0.359	0.359
.speed	1.000				0.874	0.874
ability	1.000				1.000	1.000

Comparing this to the `smode15` output that begins on page 349, we do get the same chi-squared fit test value of 9.846 with 12 degrees of freedom, so that is okay. However, the estimated first-order factor loadings are quite different. For example, the estimated loading that links the visual factor to x_1 is 0.520 for `smode16`, compared to 0.708 for `smode15`. It's way off.

After a while, I finally saw a hint pointing to the source of the problem. At the end of the `smodel6` output directly above, there are dots in front of `visual`, `verbal` and `speed`. This indicates that we are not looking at estimated variances of variables, but at estimated variances of *error terms*. It would appear that the variances of the first-order factors were not set to one after all. Instead, the variances of the error terms (that is, ϵ_1 , ϵ_2 and ϵ_3 in Expression 3.35) were set to one. I verified this by doing some calculations on numbers from the output. The result is a surrogate model that, while it's technically correct and has identifiable parameters, is just strange and does not correspond to anything we want.

On the other hand, the `Std.lv` and `Std.all` columns do contain the desired estimates. Unlike the `std.lv=TRUE` option in the `cfa` function, the `standardized=TRUE` option in `summary` is working as expected. The estimated factor loadings match `summary(smodel5)` perfectly.

To check (3.36), we first obtain $\hat{\phi}_{1,2} = 0.543$, $\hat{\phi}_{1,3} = 0.240$ and $\hat{\phi}_{2,3} = 0.284$ from the `Covariances` part of `summary(smodel5)`. Then, obtaining $\hat{\gamma}_1 = 0.678$ (not 0.923), $\hat{\gamma}_2 = 0.801$ and $\hat{\gamma}_3 = 0.355$ from `summary(smodel6)`,

```
> 0.678*0.801 # Should equal phihat12 = 0.543
[1] 0.543078
> 0.678*0.355 # Should equal phihat13 = 0.240
[1] 0.24069
> 0.801*0.355 # Should equal phihat13 = 0.284
[1] 0.284355
```

So, the `Std.all` column clearly has the estimates from a model with both the observed variables and the latent variables (not the error terms of the latent variables) standardized. It means, for example, that the estimated correlation between the ability factor and the visual factor equals 0.678 — the same as the factor loading. This is also the estimated correlation for the original model.

There is a slightly easier way to get these numbers, and that is to use the default surrogate model with a factor loading set to one for each factor, including second-order factors. The model string `swine4` is fine as it is.

```
> cat(swine4)
      visual =~ x1 + x2+x3+x5+x6+x8+x9
      verbal  =~ x4 + x2+x3+x5+x6+x8+x9
      speed   =~ x7 + x2+x3+x5+x6+x8+x9
      ability =~ visual + verbal + speed
```

The `cfa` function call is simpler.

```
> smodel7 = cfa(swine4, data=hs)
> summary(smodel7, standardized=TRUE)
lavaan 0.6-7 ended normally after 48 iterations
```

Estimator	ML
Optimization method	NLMINB

Number of free parameters 33

Number of observations 145

Model Test User Model:

Test statistic 9.846

Degrees of freedom 12

P-value (Chi-square) 0.629

Parameter Estimates:

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

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
visual =~						
x1	1.000				0.813	0.708
x2	0.733	0.191	3.841	0.000	0.596	0.538
x3	0.859	0.196	4.378	0.000	0.699	0.674
x5	-0.046	0.133	-0.350	0.727	-0.038	-0.033
x6	0.018	0.128	0.137	0.891	0.014	0.013
x8	0.534	0.162	3.302	0.001	0.434	0.415
x9	0.702	0.167	4.197	0.000	0.571	0.557
verbal =~						
x4	1.000				0.977	0.871
x2	-0.035	0.134	-0.265	0.791	-0.035	-0.031
x3	0.045	0.126	0.354	0.724	0.044	0.042
x5	0.958	0.112	8.534	0.000	0.936	0.808
x6	0.948	0.109	8.684	0.000	0.926	0.819
x8	-0.319	0.120	-2.666	0.008	-0.312	-0.298
x9	-0.064	0.115	-0.552	0.581	-0.062	-0.061
speed =~						
x7	1.000				0.806	0.782
x2	-0.103	0.147	-0.698	0.485	-0.083	-0.075
x3	-0.110	0.140	-0.786	0.432	-0.089	-0.086
x5	0.184	0.108	1.696	0.090	0.148	0.128
x6	-0.010	0.105	-0.093	0.926	-0.008	-0.007
x8	0.949	0.200	4.747	0.000	0.765	0.731
x9	0.525	0.135	3.882	0.000	0.423	0.413
ability =~						
visual	1.000				0.678	0.678
verbal	1.418	0.862	1.644	0.100	0.801	0.801
speed	0.518	0.238	2.173	0.030	0.355	0.355

Variances:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
.x1	0.657	0.120	5.498	0.000	0.657	0.499
.x2	0.907	0.125	7.245	0.000	0.907	0.740
.x3	0.575	0.101	5.661	0.000	0.575	0.535
.x5	0.405	0.073	5.589	0.000	0.405	0.302
.x6	0.412	0.069	5.924	0.000	0.412	0.322
.x8	0.347	0.104	3.323	0.001	0.347	0.317
.x9	0.480	0.075	6.421	0.000	0.480	0.456
.x4	0.303	0.065	4.626	0.000	0.303	0.241
.x7	0.412	0.120	3.428	0.001	0.412	0.388
.visual	0.357	0.233	1.532	0.126	0.540	0.540
.verbal	0.343	0.375	0.914	0.361	0.359	0.359
.speed	0.568	0.163	3.486	0.000	0.874	0.874
ability	0.304	0.208	1.460	0.144	1.000	1.000

Notice how all the leading factor loadings are set to one, including for the second-order factor. The `Std.all` column has the same numbers obtained from `smodel6`. If we want the estimates for a completely standardized model and don't care about standard errors, this is all we need. If necessary, one could define custom non-linear functions of the parameters using the `:=` notation, as on page 348, and get standard errors based on the delta method.

Constraining the error variances It is possible (but not very convenient) to actually fit a model with the variances of the first-order factors set to one. This is accomplished by constraining the variances of the error terms that feed into the first-order factors in Figure 3.16. The model equations for the latent variable part are given in Expression (3.35); they are repeated below for convenience.

$$\begin{aligned} F_1 &= \gamma_1 F_0 + \epsilon_1 \\ F_2 &= \gamma_2 F_0 + \epsilon_2 \\ F_3 &= \gamma_3 F_0 + \epsilon_3 \end{aligned}$$

Denote $Var(\epsilon_j)$ by ψ_j . With the variance of F_0 (ability) equal to one, we have $Var(F_j) = \gamma_j^2 + \psi_j$, so that $Var(F_j)$ will equal one provided $\psi_j = 1 - \gamma_j^2$ for $j = 1, 2, 3$. Here is the lavaan model string. It will be considered one piece at a time.

```
> swine5 = '
+       # Measurement model
+       visual =~ NA*x1 + x2+x3+x5+x6+x8+x9
+       verbal  =~ NA*x4 + x2+x3+x5+x6+x8+x9
+       speed   =~ NA*x7 + x2+x3+x5+x6+x8+x9
+       ability =~ NA*visual + gamma1*visual +
+               gamma2*verbal + gamma3*speed
+       # Variances
```

```

+      ability ~~ 1*ability
+      visual  ~~ psi1*visual; verbal  ~~ psi2*verbal; speed  ~~ psi3*speed
+      # Constraints to make variances of 1st order factors = 1
+      psi1 == 1 - gamma1^2
+      psi2 == 1 - gamma2^2
+      psi3 == 1 - gamma3^2
+      '

```

The first minor hurdle to overcome is that the `std.lv=TRUE` option would standardize the ϵ_j rather than the F_j , which is not what we want. However, if `std.lv=TRUE` is *not* specified, then the `cfa` function will set the leading factor loadings to one for each factor, whether or not parameter names are provided. It would be possible to specify the model more completely and use the `lavaan` function as in the `swine1` model string on page 333, but that's a lot of typing. Here's a better way. Look at the first three lines of the measurement model.

```

visual =~ NA*x1 + x2+x3+x5+x6+x8+x9
verbal =~ NA*x4 + x2+x3+x5+x6+x8+x9
speed  =~ NA*x7 + x2+x3+x5+x6+x8+x9

```

Pre-multiplying the reference variables by `NA` has the effect of *freeing* the factor loading — making it a free parameter to be estimated. The next statement shows that this facility can co-exist with providing a name for the factor loading, by naming the variable twice. This is similar to how starting values are specified — see page 80. It's not enough to name the parameter, when you are using `cfa`.

```

ability =~ NA*visual + gamma1*visual +
          gamma2*verbal + gamma3*speed

```

After fixing the variance of ability equal to one, we give names to the variances of ϵ_1 , ϵ_2 and ϵ_3 . The rule is that if you want to use a parameter in a constraint, you must name it.

```

ability  ~~ 1*ability
visual   ~~ psi1*visual; verbal  ~~ psi2*verbal; speed  ~~ psi3*speed

```

Last come the constraints, set with double equals signs.

```

psi1 == 1 - gamma1^2
psi2 == 1 - gamma2^2
psi3 == 1 - gamma3^2

```

Analyzing the correlation matrix in order to obtain standardized observed variables while avoiding the `std.ov` option²⁴,

²⁴You are forgiven if you forgot that `std.ov` divides by a sample standard deviation with $n - 1$ in the denominator.

```
> smodel8 = cfa(swine5, sample.cov=xcorr, sample.nobs=145, sample.cov.rescale=FALSE)
> summary(smodel8, standardized=TRUE)
```

lavaan 0.6-7 ended normally after 190 iterations

Estimator	ML
Optimization method	NLMINB
Number of free parameters	36
Number of observations	145

Model Test User Model:

Test statistic	9.846
Degrees of freedom	12
P-value (Chi-square)	0.629

Parameter Estimates:

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

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
visual =~						
x1	0.708	0.087	8.144	0.000	0.708	0.708
x2	0.538	0.123	4.362	0.000	0.538	0.538
x3	0.674	0.125	5.392	0.000	0.674	0.674
x5	-0.033	0.093	-0.350	0.726	-0.033	-0.033
x6	0.013	0.092	0.137	0.891	0.013	0.013
x8	0.415	0.115	3.612	0.000	0.415	0.415
x9	0.557	0.113	4.916	0.000	0.557	0.557
verbal =~						
x4	0.871	0.070	12.434	0.000	0.871	0.871
x2	-0.031	0.118	-0.264	0.791	-0.031	-0.031
x3	0.042	0.119	0.354	0.724	0.042	0.042
x5	0.808	0.090	8.939	0.000	0.808	0.808
x6	0.819	0.090	9.055	0.000	0.819	0.819
x8	-0.298	0.111	-2.673	0.008	-0.298	-0.298
x9	-0.061	0.110	-0.552	0.581	-0.061	-0.061
speed =~						
x7	0.782	0.096	8.161	0.000	0.782	0.782
x2	-0.075	0.107	-0.700	0.484	-0.075	-0.075
x3	-0.086	0.108	-0.790	0.429	-0.086	-0.086
x5	0.128	0.074	1.729	0.084	0.128	0.128
x6	-0.007	0.075	-0.093	0.926	-0.007	-0.007

x8		0.731	0.109	6.690	0.000	0.731	0.731
x9		0.413	0.096	4.321	0.000	0.413	0.413
ability = ~							
visual	(gmm1)	0.678	0.228	2.971	0.003	0.678	0.678
verbal	(gmm2)	0.801	0.244	3.284	0.001	0.801	0.801
speed	(gmm3)	0.355	0.149	2.385	0.017	0.355	0.355

Variances:

		Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
ability		1.000				1.000	1.000
.visual	(psi1)	0.540	0.309	1.746	0.081	0.540	0.540
.verbal	(psi2)	0.359	0.390	0.920	0.358	0.359	0.359
.speed	(psi3)	0.874	0.105	8.295	0.000	0.874	0.874
.x1		0.499	0.091	5.498	0.000	0.499	0.499
.x2		0.740	0.102	7.245	0.000	0.740	0.740
.x3		0.535	0.095	5.661	0.000	0.535	0.535
.x5		0.302	0.054	5.589	0.000	0.302	0.302
.x6		0.322	0.054	5.924	0.000	0.322	0.322
.x8		0.317	0.095	3.323	0.001	0.317	0.317
.x9		0.456	0.071	6.421	0.000	0.456	0.456
.x4		0.241	0.052	4.626	0.000	0.241	0.241
.x7		0.388	0.113	3.428	0.001	0.388	0.388

Constraints:

		Slack
psi1	- (1-gamma1^2)	0.000
psi2	- (1-gamma2^2)	0.000
psi3	- (1-gamma3^2)	0.000

The `Estimate` column is identical to the `Std.all` column, so it worked. This example shows that explicitly constraining error variances is an effective way to standardize endogenous latent variables. However, it can be tedious for large, multistage models. By using `parameterEstimates(smodel8)`, one could automatically obtain 95% confidence intervals for all the parameters, including the ones (ψ_1 , ψ_2 and ψ_3) that have been made functionally dependent on other parameters.

Testing equal factor loadings Constraints are a handy way to specify null hypotheses for likelihood ratio tests. They may be placed in the model string, but it's preferable to give them in the `cfa` or `lavaan` statement. That way, the same model string can be used to specify the full model and the restricted model.

Suppose we wish to test equality of the second order factor loadings; the null hypothesis is $H_0 : \gamma_1 = \gamma_2 = \gamma_3$. The model under this null hypothesis is expressed as

```
> smodel8H0 = cfa(swine5, sample.cov=xcorr,
+               sample.nobs=145, sample.cov.rescale=FALSE,
+               constraints = 'gamma1 == gamma2
+               gamma2 == gamma3')
```

It's often advisable to look at a summary of the restricted model, just to be sure that nothing obvious has gone wrong. That step is not shown here. Then, the `anova` function generates a likelihood ratio test. If $p < 0.5$, the null hypothesis given in the restricted model is rejected at the 0.05 significance level.

```
> anova(smodel8H0, smodel8)
Chi-Squared Difference Test
```

	Df	AIC	BIC	Chisq	Chisq diff	Df diff	Pr(>Chisq)
smodel8	12	3273.5	3371.7	9.8461			
smodel8H0	14	3273.5	3365.8	13.8616	4.0155	2	0.1343

So, there's insufficient evidence to conclude that the second-order factor loadings are different. Or, one could say that the results are consistent with equal second-order factor loadings²⁵.

To do this test with `smodel17` (in which leading factor loadings equal one), realize that under `smodel8`, the γ_j are exactly the correlations of F_0 with F_j . They are also the correlations of F_0 with F_j , under the original model, and under the model of `smodel17`. So for the model of `smodel17`, we seek to test the null hypothesis of equal correlations. This implies some constraints on the parameters that are not at all intuitive. The result is either a good homework problem or a place where I need to show my work²⁶.

For `smodel17`, the equations of the latent part of the model are

$$\begin{aligned} F_1 &= F_0 + \epsilon_1 \\ F_2 &= \gamma_2 F_0 + \epsilon_2 \\ F_3 &= \gamma_3 F_0 + \epsilon_3, \end{aligned}$$

with $Var(F_0) = \phi$, $Var(\epsilon_j) = \psi_j$ for $j = 1, 2, 3$, and of course F_0 independent of the ϵ_j . The variances of the first-order factors are

$$Var(F_1) = \phi + \psi_1, \quad Var(F_2) = \gamma_2^2 \phi + \psi_2, \quad Var(F_3) = \gamma_3^2 \phi + \psi_3,$$

and

$$\begin{aligned} Cov(F_0, F_1) &= Cov(F_0, F_0 + \epsilon_1) = \phi \\ Cov(F_0, F_2) &= Cov(F_0, \gamma_2 F_0 + \epsilon_2) = \gamma_2 \phi \\ Cov(F_0, F_3) &= Cov(F_0, \gamma_3 F_0 + \epsilon_3) = \gamma_3 \phi, \end{aligned}$$

²⁵As usual in applied statistics, we are not actively accepting the null hypothesis. For example, if we say that the results are consistent with equal second-order factor loadings, what we really mean is that they are not inconsistent with equal factor loadings. That is, the null hypothesis was not rejected.

²⁶Maybe it's both. Some students are surprised when they discover that the answers to many homework problems are directly in the textbook. It's a sneaky way to encourage students to read the text.

so that

$$\begin{aligned} \text{Corr}(F_0, F_1) &= \frac{\phi}{\sqrt{\phi(\phi+\psi_1)}} = \frac{\sqrt{\phi}}{\sqrt{\phi+\psi_1}} \\ \text{Corr}(F_0, F_2) &= \frac{\gamma_2\phi}{\sqrt{\phi(\gamma_2^2\phi+\psi_2)}} = \frac{\gamma_2\sqrt{\phi}}{\sqrt{\gamma_2^2\phi+\psi_2}} \\ \text{Corr}(F_0, F_3) &= \frac{\gamma_3\phi}{\sqrt{\phi(\gamma_3^2\phi+\psi_3)}} = \frac{\gamma_3\sqrt{\phi}}{\sqrt{\gamma_3^2\phi+\psi_3}}. \end{aligned}$$

Since $\text{Corr}(F_0, F_1) > 0$, the null hypothesis of equal correlations implies $\gamma_2 > 0$ and $\gamma_3 > 0$. Using this,

$$\begin{aligned} \text{Corr}(F_0, F_1) = \text{Corr}(F_0, F_2) &\iff \frac{\sqrt{\phi}}{\sqrt{\phi+\psi_1}} = \frac{\gamma_2\sqrt{\phi}}{\sqrt{\gamma_2^2\phi+\psi_2}} \\ &\iff \frac{1}{\sqrt{\phi+\psi_1}} = \frac{\gamma_2}{\sqrt{\gamma_2^2\phi+\psi_2}} \\ &\iff \gamma_2\sqrt{\phi+\psi_1} = \sqrt{\gamma_2^2\phi+\psi_2} \\ &\iff \gamma_2^2(\phi+\psi_1) = \gamma_2^2\phi+\psi_2 \\ &\iff \gamma_2^2\phi+\gamma_2^2\psi_1 = \gamma_2^2\phi+\psi_2 \\ &\iff \psi_2 = \gamma_2^2\psi_1. \end{aligned} \tag{3.37}$$

Similarly,

$$\begin{aligned} \text{Corr}(F_0, F_1) = \text{Corr}(F_0, F_3) &\iff \frac{\sqrt{\phi}}{\sqrt{\phi+\psi_1}} = \frac{\gamma_3\sqrt{\phi}}{\sqrt{\gamma_3^2\phi+\psi_3}} \\ &\iff \frac{1}{\sqrt{\phi+\psi_1}} = \frac{\gamma_3}{\sqrt{\gamma_3^2\phi+\psi_3}} \\ &\iff \gamma_3\sqrt{\phi+\psi_1} = \sqrt{\gamma_3^2\phi+\psi_3} \\ &\iff \gamma_3^2(\phi+\psi_1) = \gamma_3^2\phi+\psi_3 \\ &\iff \gamma_3^2\phi+\gamma_3^2\psi_1 = \gamma_3^2\phi+\psi_3 \\ &\iff \psi_3 = \gamma_3^2\psi_1. \end{aligned} \tag{3.38}$$

Finally,

$$\begin{aligned} \text{Corr}(F_0, F_2) = \text{Corr}(F_0, F_3) &\iff \frac{\gamma_2\sqrt{\phi}}{\sqrt{\gamma_2^2\phi+\psi_2}} = \frac{\gamma_3\sqrt{\phi}}{\sqrt{\gamma_3^2\phi+\psi_3}} \\ &\iff \frac{\gamma_2}{\sqrt{\gamma_2^2\phi+\psi_2}} = \frac{\gamma_3}{\sqrt{\gamma_3^2\phi+\psi_3}} \\ &\iff \gamma_2\sqrt{\gamma_3^2\phi+\psi_3} = \gamma_3\sqrt{\gamma_2^2\phi+\psi_2} \\ &\iff \gamma_2^2(\gamma_3^2\phi+\psi_3) = \gamma_3^2(\gamma_2^2\phi+\psi_2) \\ &\iff \gamma_2^2\gamma_3^2\phi+\gamma_2^2\psi_3 = \gamma_2^2\gamma_3^2\phi+\gamma_3^2\psi_2 \\ &\iff \gamma_2^2\psi_3 = \gamma_3^2\psi_2 \end{aligned} \tag{3.39}$$

Only two of these constraints are necessary; any two imply the remaining one. One thing that's clear from all this so far is that even though the calculations are elementary, this is a lot of work to set up just one null hypothesis. When the interest is in correlations, a model with standardized variables is preferable. Since most of the work has been done, let's proceed.

In order to impose constraints on parameters in lavaan, the parameters involved must be named in the model string. It's convenient to assemble a new model string by adding to `swine3`.

```
> cat(swine3)
      visual =~ x1 + x2+x3+x5+x6+x8+x9
      verbal  =~ x4 + x2+x3+x5+x6+x8+x9
      speed   =~ x7 + x2+x3+x5+x6+x8+x9
> part2 = '# Second order measurement model
+         ability =~ visual + gamma2*verbal + gamma3*speed
+         # Variances of error terms (epsilons)
+         visual ~~ psi1*visual; verbal ~~ psi2*verbal; speed ~~ psi3*speed '
> swine6 = paste(swine3,part2)
> cat(swine6)
      visual =~ x1 + x2+x3+x5+x6+x8+x9
      verbal  =~ x4 + x2+x3+x5+x6+x8+x9
      speed   =~ x7 + x2+x3+x5+x6+x8+x9
      # Second order measurement model
      ability =~ visual + gamma2*verbal + gamma3*speed
      # Variances of error terms (epsilons)
      visual  ~~ psi1*visual; verbal  ~~ psi2*verbal; speed  ~~ psi3*speed
```

To test this code, I verified that it produced the same fit and parameter estimates as `smodel7` (starting on page 355), except with a few extra labels. Then I tried to fit the model with the constraints (3.37) and (3.38).

```
> # Constraints are equivalent to equal correlations of F0 with F_j. This is H0.
> smodel7H0 = cfa(swine6, data = hs, constraints = 'psi2 == gamma2^2 * psi1
+                                                  psi3 == gamma2^3 * psi1 ' )
```

It took a long time to run, which is almost always a bad sign. Then,

Warning messages:

```
1: In lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable, :
lavaan WARNING: the optimizer warns that a solution has NOT been found!
2: In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats, :
lavaan WARNING:
The variance-covariance matrix of the estimated parameters (vcov)
does not appear to be positive definite! The smallest eigenvalue
(= -2.656966e-21) is smaller than zero. This may be a symptom that
the model is not identified.
3: In lav_object_post_check(object) :
lavaan WARNING: some estimated lv variances are negative
```

The parameters are identifiable in most of the parameter space, and the regions where they are not identifiable do not correspond to the constraints. So, we can discount suggestion that possibly “the model is not identified”—though typos can accidentally specify a model that is not what one intends, and whose parameters are not identifiable. The warning about negative variance estimates is helpful. Let’s look at a summary.

```
> summary(smodel7H0, standardized=TRUE)
```

```
lavaan 0.6-7 ended normally after 1336 iterations
```

Estimator	ML
Optimization method	NLMINB
Number of free parameters	33
Number of observations	145

Model Test User Model:

Test statistic	671.784
Degrees of freedom	14
P-value (Chi-square)	0.000

Parameter Estimates:

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

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
visual =~						
x1	1.000				NA	NA
x2	-0.002	0.001	-3.092	0.002	NA	NA
x3	-0.004	0.000	-9.062	0.000	NA	NA
x5	-0.005	0.000	-10.974	0.000	NA	NA
x6	-0.001	0.001	-2.249	0.025	NA	NA

```

      x8          -0.001    0.000      -1.795    0.073      NA      NA
      x9          -0.004    0.000     -10.413    0.000      NA      NA
verbal =~
      x4          1.000
      x2          2.116    0.000  32578611.185    0.000      NA      NA
      x3          3.977    0.000  39039739.233    0.000      NA      NA
      x5          8.504    0.000  15717415.576    0.000      NA      NA
      x6          5.219    0.000  94329059.781    0.000      NA      NA
      x8          -0.891    0.000  -5466418.974    0.000      NA      NA
      x9          1.521    0.000  14915452.630    0.000      NA      NA
speed =~
      x7          1.000
      x2          0.116    0.000   18821.703    0.000    3.528    0.815
      x3          0.217    0.000   34832.277    0.000    0.410    0.335
      x5          0.394    0.000   11697.369    0.000    0.765    0.620
      x6          0.317    0.000   63860.916    0.000    1.390    0.934
      x8          0.101    0.000    7754.555    0.000    1.117    0.747
      x9          0.218    0.000   19810.814    0.000    0.357    0.337
ability =~
      visual          1.000
      verbal (gmm2)    0.004    0.000   190959.565    0.000    0.768    0.649
      speed  (gmm3) -14577.264
                                -1.000    -1.000

Variances:
      Estimate  Std.Err  z-value  P(>|z|)  Std.lv  Std.all
. visual (psi1) -129.680   0.000 -20649480.153  0.000    NA      NA
. verbal (psi2)  -0.002   0.001   -1.772    0.076    NA      NA
. speed  (psi3)  -0.000
.x1          131.055   0.000  20972882.910  0.000  131.055  95.304
.x2          1.334   0.000  4375786.419  0.000   1.334   0.894
.x3          0.973   0.000  847631.935  0.000   0.973   0.640
.x5          0.441   0.000  33863.868  0.000   0.441   0.199
.x6          1.047   0.000  893652.826  0.000   1.047   0.468
.x8          0.992   0.000  929543.299  0.000   0.992   0.888
.x9          0.820   0.000  498369.776  0.000   0.820   0.584
.x4          1.533   0.000  13735738.550  0.000   1.533   1.001
.x7          6.310   0.000  5932823.378  0.000   6.310   0.336
ability          0.000
                                1.000   1.000

Constraints:
                                |Slack|
      psi2 - (gamma2^2*psi1)          0.000
      psi3 - (gamma2^3*psi1)          0.000
# parTable(smodel7H0) # Start obeyed the constraints.

```

It can be beneficial too look at something this ugly. There are several indications that the numerical search for the MLE went off the rails. The number of iterations was 1336,

which is too many; `smodel7` took 48 iterations, and `smodel8` took 190. Also, many of the supposed parameter estimates are really huge. Then there's the $\hat{\psi}_3$ of -20649480.153. This is an estimated *variance*. The output shows all the signs of a numerical search that accidentally left the parameter space, found a direction that was slightly less bad than where it landed, and then wandered off into nowhere until lavaan (or actually, the underlying `nlm` function) pulled the plug because of too many iterations.

The standard cure for this disease is better starting values. As we saw in the BMI Health Study (Section 0.10.4, starting on page 89), providing a large number of starting values can be a lot of work. There is an alternative that is promising in this case, but which I have not tried. It's to use the `imposeStart` function from the `semTools` package. Using `imposeStart`, you are able to start a numerical search where another similar model successfully finished. Just provide names for the parameters involved. Here, I would start with the most of the estimates from `smodel7`; it would be necessary to provide labels for the parameters whose estimates were to be used as starting values.

I did not do this, because it turned out that I did not need to. Hoping for the best, I imposed the constraints (3.37) and (3.39) in place of (3.37) and (3.38). Even though these two ways of expressing the null hypothesis are mathematically equivalent, numerical software does not do all the math. I was guessing that the numerical details of imposing the constraints would be sufficiently different so that the search would not get lost at the same point as before. Presumably because I have been a good person my entire life, it worked.

```
> # Try again, with different expression of the same constraints
> smodel7H0 = cfa(swine6, data = hs,
+               constraints = 'psi2 == gamma2^2 * psi1
+                             gamma2^2 * psi3 == gamma3^2 * psi2 ')
>
> # summary(smodel7H0) # Commented out
> anova(smodel7H0,smodel7)
Chi-Squared Difference Test
```

	Df	AIC	BIC	Chisq	Chisq diff	Df diff	Pr(>Chisq)
smodel7	12	3494.1	3592.3	9.8461			
smodel7H0	14	3494.1	3586.4	13.8616	4.0155	2	0.1343

The results are exactly the same as for `anova(smodel8H0,smodel8)` on page 361.

Two lessons may be learned from this last excursion. The first lesson is that it's too much work. If you are interested in an identifiable function of the original model parameters, try to use a surrogate model in which that identifiable function is a model parameter. Expressions (3.17) and (3.19) may be helpful, as may the discussion in Section 3.7. If there is no such surrogate model, okay. But don't make things more complicated than they need to be.

The second lesson is narrow and technical, but unexpected. Suppose a restricted model is being tested against an unrestricted model, and that the unrestricted model fits, while the restricted model does not. Re-expressing the constraints in a mathematically equivalent (and not necessarily simpler) way may be helpful.

3.9 The Importance of Planning and Design

In the typical factor analytic study, the investigator hands out a bunch of questionnaires²⁷, or invites people to complete the questionnaires online. Either way, the observable variables in the study are derived from the responses of the participants. Values of the variables are all generated at more or less the same time and under more or less the same circumstances. This unleashes a flood of very predictable common influences. The respondents' mood, recent experiences, view of the investigator, self-presentation strategies, and guesses about the real purpose of the study — all of these latent variables and many more may be assumed to affect the observable variables.

Chances are very good that such variables are not the focus of the study, and are not among the hypothesized factors. That means they are incorporated into the error terms, and because the same extraneous variables will impact more than one observable variable, the result is non-zero covariances between error terms. These common influences are numerous and we don't know exactly what they are, so the most reasonable model will include all possible covariances between error terms.

Such a model may be reasonable, but it is not useable. There are k observable variables, and an error term for each one. The covariance matrix of the observable variables, Σ , is $k \times k$, and the covariance matrix of the errors, Ω , is also $k \times k$. There are already as many unknown parameters as covariance structure equations, so the presence of even a single common factor will violate the [parameter count rule](#). Parameter identifiability is out of reach, and so is consistent estimation. Trying to fit the model by maximum likelihood is guaranteed to fail.

Of course, model with all possible covariances between the errors is not what the factor analyst will try to fit. Instead, it is *very* common to assume a model in which all the error terms are independent. The parameters of such a model may be identifiable, but the model is *mis-specified*. That is, it's not correct. Correlations between observable variables will be taken as evidence for the operation of common factors, when in reality they are due to correlations between errors.

How bad will it be? It's really impossible to say. Certainly, parameter estimates will be at least a little off, even for very large sample sizes. Maybe, the effects of the extraneous unmeasured variables will be small compared to the effects of the common factors, and the picture that emerges will be a fair reflection of reality in all essential respects. Or maybe, the correlations between observed variables will be largely determined by the correlations between error terms, making any conclusions from the analysis scientifically worthless. It is impossible to tell, precisely because, apart from the background noise of sampling error, what identifiable means is *knowable*.

When the model with independent errors is applied to data, it may fit and it may not. If it does not fit, it could be that the correlations between errors have created a sample covariance matrix that is inconsistent with the common factor part of the model. At least there is a clue that something is wrong. If the model does fit, it may be that everything is fairly close to being okay, but not necessarily. This is the case with the Holzinger and

²⁷Or educational tests.

Swineford data of Section 3.8.

In any observational study, there will inevitably be omitted variables that can distort the results. See Section 0.4 in Chapter 0 for a discussion of how this can affect ordinary regression. However, the focus here is on a set of processes that specifically corrupt the measurement process, and can be controlled.

First, the problem is worst when human subjects are involved, and are aware that their behaviour is being assessed. In contrast, imagine a physical anthropology study in which 14 measurements are to be conducted on a sample of 273 fossilized bones and bone fragments. Measurement error is certainly going to occur, but it's easy enough to minimize correlations among the errors. Just randomize the order in which the measurements are taken, over both bones and features. So for example, the person collecting the data will first measure characteristic 6 on bone 47, then characteristic 11 on bone 122, and so on. It's a bit of extra work, but it would make a model with independent errors quite reasonable.

Research design The key to the last example was collecting the data a bit differently. This is an aspect of research design. That's true in more difficult cases as well. This chapter has introduced a good number of rules for establishing parameter identifiability, but there are two big ones — the [double measurement rule](#) and the [reference variable rule](#)²⁸. In both cases, the rules allow for subsets of variables whose error terms might be correlated, but require zero correlation of the error terms for different subsets. For examples, see the [BMI health study](#) of Section 0.10.4 and the [Brand Awareness study](#) of Section 1.6. Section 3.4 on the [reference variable rule](#) also has examples, as well as extended discussion of correlated measurement error and how the design allows for it.

The point here is that there are good alternatives to just handing out a bunch of questionnaires and hoping for the best, but they require advance planning. In other applications of statistics, especially in experiments with random assignment, it is commonplace to think of the research question, the statistical analysis and the details of data collection all at the same time. Factor analysis should be no different. Of course, this principle also holds when a factor analysis model is part of a larger structural equation model.

²⁸Why are these two the “big ones?” Because they grant entry to the system, establishing the parameters of a model or sub-model as identifiable. Then other rules may be used to expand the model or put sub-models together.