

Introduction to Structural Equation Modeling: Confirmatory Factor Analysis

Henrik Kenneth Andersen

2021-10-08

Introduction

In the social sciences, we are often interested with abstract concepts, like xenophobia, environmental attitudes, or need for social approval.

Such abstract concepts cannot be measured directly, unlike other variables, such as income or number of children. We refer to such abstract concepts as *latent variables*. There are many different names for latent variables, such as

- factors,
- latent constructs,
- unobserved variables,
- hypothetical constructs/variables, etc.

Confirmatory factor analysis is a structural equation modeling method to estimate latent variables from the covariances between the observed variables, called *indicators*.

These observed covariances are assumed to reflect a *shared underlying cause*: the latent variable. This means we can decompose the variable into common variance shared with the other variables and unique variance.

We call modeling the relationships between the underlying latent variables and the observed variables the *measurement model*, whereas the path analysis discussed earlier is referred to as the *structural model*.

Factor analysis

The standard *exploratory factor analysis* (EFA) is a widely-used technique *outside of SEM*. For a set of indicators, we are interested in finding the number of *factors*, or latent variables, that adequately describe the covariance of the indicators.

In *confirmatory factor analysis* (CFA), we set the number of latent constructs *a priori*. If we are modeling two or more latent constructs at the same time, we usually assume that a set of indicators of one latent variable are not associated with the other latent variables, though the latent variables themselves can be correlated (something akin to oblique rotation for correlated factors). Contrast this with exploratory factor analysis, where each indicator is allowed to load on each of the factors.

We can think of CFA as decomposing the observed variance into

- the *common variance* due to the latent variable, and
- an *error term*,

which essentially just builds off of classical test theory where it is assumed

$$\text{observed score} = \text{true score} + \text{error}.$$

In other words, we assume the associations between variables are due to some latent construct, and use those observed associations to essentially reverse-engineer the latent cause. Then, we say the true score is the part due to the latent variable and the deviation from that is the error. A more formal way of expressing the classical test theory assumption would be

$$x_j = \lambda_{jk}\xi_k + \delta_j$$

where x_j is the j th indicator, linked to the k th latent variable, ξ_k , by the factor loading λ_{jk} and δ_j is the error term.

Normally, the error in a CFA is called the *measurement error*, because it is assumed that the only source of systematic variance is the latent variable, the rest is random error due to imprecise measuring instruments. However, this is a somewhat restrictive view, and therefore it is sometimes differentiated between *error variance* and *unique variance*, where the unique variance can be due to other unobserved factors influencing the observed score. This differentiation is mostly irrelevant, as long as we can assume that the error, regardless of whether it is systematic or random, is uncorrelated with the latent variable which we will come back to momentarily.

An example

Let us start with an example of a three-indicator measurement model. We will recreate the measurement model for xenophobia (or ‘anti-foreigner attitudes’; what we call the latent variable is basically arbitrary) used in [Reinecke and Pöge \(2010\)](#).

They use three indicators found in the ALLBUS 2006 to measure xenophobia, which are also included in the newest ALLBUS 2018 data:

- x_1 : “There is a dangerous amount of foreigners living in Germany” (px06)
- x_2 : “Foreigners should marry within their own group” (px07)
- x_3 : “Attacks on asylum-seeker housing is understandable” (px10)

(roughly translated from German), which were measured on a Likert scale of 1: “fully disagree,” . . . , 5: “fully agree.” Notice the indicators are not continuous and therefore deviate from the ideal normality assumption (at best, they can be thought of as ‘coarse’ or ‘rough’ measures of the actual normally distributed variable). We will ignore this fact out of convenience.

We see that there are indeed sizable correlations amongst these three variables:

```
# Load haven
library(haven)

# Set working directory
setwd("../04_data")

# Import ALLBUS 2018
df <- read_sav("allbus2018.sav")

# Show correlation matrix
cor(df[, c("px06", "px07", "px10")], use = "complete.obs")
```

Table 1: Notation

Symbol	Pronunciation	Explanation
x		exogenous (independent) observed variable
y		endogenous (dependent) observed variable
ξ	ksi	exogenous latent variable
η	eta	endogenous latent variable
ζ	zeta	error or disturbance of endogenous latent variable
δ	delta	error of exogenous observed variable
ϵ	epsilon	error of endogenous observed latent variable
Λ_x	Lambda x	factor loading matrix for exogenous variables
Λ_y	Lambda y	factor loading matrix for endogenous variables
Φ	Phi	covariance matrix of exogenous latent variables
Ψ	Psi	covariance matrix of endogenous variables (errors/disturbances)
Θ_δ	Theta delta	covariance matrix of the exogenous errors
Θ_ϵ	Theta epsilon	covariance matrix of the endogenous variables (errors)
τ_x	tau x	intercept of observed exogenous variable
τ_y	tau y	intercept of observed endogenous variable
λ_x	lambda x	element of matrix Λ_x
λ_y	lambda y	element of matrix Λ_y
θ_δ	theta delta	element of matrix Θ_δ
θ_ϵ	theta delta	element of matrix Θ_ϵ
ϕ	phi	element of matrix Φ
ψ	psi	element of matrix Ψ

```
##          px06      px07      px10
## px06 1.0000000 0.4642655 0.3196091
## px07 0.4642655 1.0000000 0.3361614
## px10 0.3196091 0.3361614 1.0000000
```

and we assume there is a latent construct, which we could call xenophobia, that is responsible for these correlations.

Notation and illustration

We will need some new notation to describe our CFA. An updated list of notation is found in Table 1. First, we will introduce *factor loadings*, λ , which are essentially regression coefficients, but stemming from unobserved latent variables. Note that the variables x and y are now less the variables of interest, and more so simply *indicators* of the latent variables. The latent variables are now actually where our substantive interest lies. We call the variances of the so-called *measurement errors*, i.e., the unexplained parts of the observed indicators, δ and ϵ , as before, but their variances are referred to with θ .

So, we could again write

$$x_j = \lambda_{jk}\xi_k + \delta_j$$

where λ_{jk} is the factor loading of indicator j on latent variable k and $\xi_k \sim N(0, \phi_k)$ because it is an exogenous latent variable and $\delta_j \sim N(0, \theta_j)$ since we are talking about the measurement error portion of an exogenous indicator.

Since in an isolated CFA there are no other variables influencing the latent variable xenophobia, we would call the three indicators x_1, x_2, x_3 ¹ and the latent variable ξ_1 or just ξ . Let us continue to assume that the variables are mean-centered for now.² We would write the model as

$$\begin{aligned}x_1 &= \lambda_{11}\xi_1 + \delta_1 \\x_2 &= \lambda_{21}\xi_1 + \delta_2 \\x_3 &= \lambda_{31}\xi_1 + \delta_3\end{aligned}$$

or, more compactly in matrix notation as

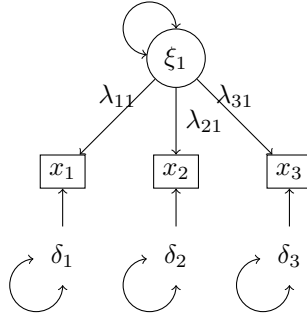
$$\mathbf{x} = \mathbf{\Lambda}_x \boldsymbol{\xi} + \boldsymbol{\delta}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \\ \lambda_{31} \end{bmatrix} \xi_1 + \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix}$$

where, in this case, $\boldsymbol{\xi}$ is a scalar. The notation on the subscripts is a bit more complicated than it has to be, since we have only one latent variable, and no dependent measurement model components, but it will help later to be stringent about this now.

We can illustrate the model even more straightforwardly as a path diagram like the one shown in Figure 1. From it, we can see by the lack of any two-headed arrows between the errors and between the errors and the latent variable that we are assuming they are uncorrelated.

Figure 1: Three-item measurement model



If the observed variables are mean-centered, the model implied-covariance matrix of \mathbf{x} is $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, where

$$\begin{aligned}\boldsymbol{\Sigma}(\boldsymbol{\theta}) &= \mathbb{E}(\mathbf{x}\mathbf{x}^\top) \\ &= \mathbb{E}[(\boldsymbol{\lambda}_x \boldsymbol{\xi} + \boldsymbol{\delta})(\boldsymbol{\lambda}_x \boldsymbol{\xi} + \boldsymbol{\delta})^\top] \\ &= \mathbb{E}[(\boldsymbol{\lambda}_x \boldsymbol{\xi} + \boldsymbol{\delta})(\boldsymbol{\xi}^\top \boldsymbol{\lambda}_x^\top + \boldsymbol{\delta}^\top)] \\ &= \boldsymbol{\lambda}_x \boldsymbol{\Phi} \boldsymbol{\lambda}_x^\top + \boldsymbol{\Theta}_\delta\end{aligned}$$

since $\mathbb{E}(\boldsymbol{\xi}\boldsymbol{\delta}^\top) = \mathbf{0}$ by assumption. $\boldsymbol{\Phi} = \mathbb{E}(\boldsymbol{\xi}\boldsymbol{\xi}^\top)$ is the covariance matrix of the latent variables in $\boldsymbol{\xi}$. In this case, because $\boldsymbol{\xi} = \xi_1$, it contains just $\mathbb{E}(\xi_1^2)$, the variance of the one latent variable. $\boldsymbol{\Theta}_\delta$ is the covariance

¹This can be confusing because indicators are, by definition, dependent or endogenous variables. What counts is whether their latent variable is exogenous or endogenous and we would normally use x for indicators of the former and y for the latter. There are other notations, such as the ‘all- y ’ notation which uses y for all indicators and η for all latent variable, regardless of whether they are exogenous or endogenous.

²We can easily include the intercepts by writing something like $x_j = \tau_j + \lambda_{jk}\xi_k + \epsilon_j$ and $y_m = \tau_m + \lambda_{mn}\eta_n + \epsilon_m$. The intercepts represent the expected value of the observed variable when the latent variable equals zero.

matrix of the errors. $\mathbb{E}(\delta_j \delta_k) = 0$, $j \neq k$, by assumption, so Θ_δ has the variances on the diagonal and zeros everywhere else,

$$\Theta_\delta = \mathbb{E} \begin{bmatrix} \delta_1^2 & & \\ \delta_2 \delta_1 & \delta_2^2 & \\ \delta_3 \delta_1 & \delta_3 \delta_2 & \delta_3^2 \end{bmatrix} = \begin{bmatrix} \theta_1 & & \\ 0 & \theta_2 & \\ 0 & 0 & \theta_3 \end{bmatrix}.$$

Assumptions

We assume the errors, δ_j , are random and simply due to imprecise measurements. This implies that

$$\text{Cov}(\delta_j, \xi_1) = 0, \quad j = 1, 2, 3$$

$$\text{Cov}(\delta_j, \delta_k) = 0, \quad j \neq k.$$

Both of these assumptions are very important. Let us look at this more closely, beginning with the uncorrelatedness of the latent variable with each of the errors.

Note that we can also express these assumptions in terms of expectations as $\mathbb{E}(\boldsymbol{\delta}|\eta) = \mathbb{E}(\boldsymbol{\delta}) = \mathbf{0}$ (where since the variables are centered — if not, we would include an intercept — it is safe to assume the unconditional expectation of the errors is zero), and $\mathbb{E}(\delta_j \delta_k) = 0$, $j \neq k$. The first assumption is analogous to the assumption that the error is uncorrelated with the predictor in a linear regression model. Here, the only thing different is that the independent variable happens to be the latent variable.

Take the second indicator, x_2 : “Foreigners should marry within their own group” (`px07`). We can imagine xenophobic individuals will agree with this statement because they dislike cultural or ethnic mixing. We can also imagine individuals with traditional values will agree with this statement because they believe marriage between partners with similar backgrounds tends to work out better. We can furthermore assume that traditional values are correlated with xenophobic attitudes, i.e., those with traditional values also tend to be more xenophobic. So we call xenophobia ξ_1 , and could represent traditional values as a second latent variable, ξ_2 . In our measurement model, we assumed that xenophobia causes the observed score on the indicator `px07` and $\mathbb{E}(\delta_2, \xi_1) = 0$. But if our alternate theory is correct, then the actual model would be

$$x_2 = \lambda_{11}\xi_1 + \lambda_{12}\xi_2 + \delta_2$$

where the observed variable is caused by both xenophobia and traditional values. Now, we failed to include traditional values above, so we have

$$x_2 = \lambda_{11}\xi_1 + \tilde{\delta}_2$$

where $\tilde{\delta}_2 = \lambda_{12}\xi_2 + \delta_2$, i.e., is the effect of traditional values is relegated to the error. By covariance algebra, the factor loading would be

$$\begin{aligned} \text{Cov}(\xi_1, x_2) &= \text{Cov}(\xi_1, \lambda_{11}\xi_1 + \delta_2) \\ &= \lambda_{11} \text{Var}(\xi_1) \\ \hat{\lambda}_{11} &= \frac{\text{Cov}(\xi_1, x_2)}{\text{Var}(\xi_1)} \end{aligned}$$

but by the Frisch-Waugh theorem ([Frisch and Waugh 1933](#)), we know the estimated factor loading λ_{11} will

be biased as long as $\text{Cov}(\xi_1, \xi_2) \neq 0$:

$$\begin{aligned}
\hat{\lambda} &= \frac{\text{Cov}(\xi_1, x_2)}{\text{Var}(\xi_1)} \\
&= \frac{\text{Cov}(\xi_1, \lambda_{11}\xi_1 + \lambda_{12}\xi_2 + \delta_2)}{\text{Var}(\xi_1)} \\
&= \frac{\lambda_{11} \text{Var}(\xi_1) + \lambda_{12} \text{Cov}(\xi_1, \xi_2)}{\text{Var}(\xi_1)} \\
&= \lambda + \frac{\lambda_{12} \text{Cov}(\xi_1, \xi_2)}{\text{Var}(\xi_1)}.
\end{aligned}$$

If $\text{Cov}(\xi_1, \xi_2) = 0$, then the numerator in the equation above is zero and we are left with $\hat{\lambda}_{11} = \lambda_{11}$, i.e., the estimated factor loading equals the true value and it is thus unbiased. However, in this case, the estimated factor loading would likely be positively biased, since the traditionalism and xenophobia would be positively correlated and the effect of traditionalism on indicator x_2 would be positive, as well.

This is important because *misspecification in the measurement models will generally have an impact on the rest of the model* (Rosseel and Loh 2021). Once we begin looking at full SEMs, the factor loadings are usually no longer of central interest, as long as they are adequate in terms of contributing to a valid and reliable measurement of the latent variable (more on that below). However, bias in the factor loadings will also impact the *structural parameters*, i.e., the relationships between the latent variables, which are of central interest.

Now, consider the assumption of the uncorrelatedness of the errors. This can be described by the following model (focusing just on indicators x_2 and x_3 for the sake of simplicity):

$$\begin{aligned}
x_2 &= \lambda_{21}\xi_1 + \lambda_{22}\xi_2 + \delta_2 \\
x_3 &= \lambda_{31}\xi_1 + \lambda_{32}\xi_2 + \delta_3
\end{aligned}$$

where we again have two latent variables, ξ_1 and ξ_2 , but this time the second latent variable influences both indicators. We could continue with the traditionalism example, though if we are looking at `px07` and `px10`, we would not necessarily have a reason to believe why those with traditional values would think attacks on asylum-seeker housing would be acceptable. Instead, let us assume ξ_2 stood for the individual's need for social approval. Scores on both items would be influenced by xenophobia — what we are interested in — but also social desirability, i.e., individuals would likely downplay their agreement with both items regardless of their true xenophobia.

If we failed to account for need for social approval, we would have again just:

$$\begin{aligned}
x_2 &= \lambda_{21}\xi_1 + \tilde{\delta}_2 \\
x_3 &= \lambda_{31}\xi_1 + \tilde{\delta}_3
\end{aligned}$$

where $\tilde{\delta}_2 = \lambda_{22}\xi_2 + \delta_2$ and $\tilde{\delta}_3 = \lambda_{32}\xi_2 + \delta_3$. Now, by covariance algebra, we would have

$$\begin{aligned}
\text{Cov}(\xi_1, x_2) &= \text{Cov}(\xi_1, \lambda_{21}\xi_1 + \tilde{\delta}_2) \\
&= \lambda_{21} \text{Var}(\xi_1) \\
\hat{\lambda}_{21} &= \frac{\text{Cov}(\xi_1, x_2)}{\text{Var}(\xi_1)}
\end{aligned}$$

since $\text{Cov}(\xi_1, \delta) = \mathbf{0}$, by assumption (we could just as well have looked at $\hat{\lambda}_{31}$). We see, though, that the

factor loading is biased to the extent that the two factors are correlated:

$$\begin{aligned}\hat{\lambda}_{21} &= \frac{\text{Cov}(\xi_1, x_2)}{\text{Var}(\xi_1)} \\ &= \frac{\text{Cov}(\xi_1, \lambda_{11}\xi_1 + \lambda_{12}\xi_2 + \delta_2)}{\text{Var}(\xi_1)} \\ &= \lambda_{21} + \frac{\lambda_{12} \text{Cov}(\xi_1, \xi_2)}{\text{Var}(\xi_1)}.\end{aligned}$$

In this case, it may be plausible that the factors, need for social approval and xenophobia, are essentially uncorrelated, and the factor loadings may be unbiased. But we will have a source of misfit, because we are neglecting to recognize the existence of need for social approval, which would cause the errors to be correlated. In other words, if we erroneously place the second unobserved factor, need for social approval, in the error terms, then the errors will be correlated across indicators

$$\begin{aligned}\text{Cov}(\tilde{\delta}_2, \tilde{\delta}_3) &= \text{Cov}(\lambda_{22}\xi_2 + \delta_2, \lambda_{32}\xi_2 + \delta_3) \\ &= \lambda_{22}\lambda_{32} \text{Var}(\xi_2) \neq 0.\end{aligned}$$

For this reason, it is a *popular simplifying assumption* to state that the rest of the *variance unique to the indicator is random measurement error*. If it is simply random error, there is no reason to believe there would be a correlation between the measurement error and the latent factor, nor between the measurement errors across indicators. Like any other assumption, this can fail, as we have seen.

Again, if this assumption does not hold, then the measurement models are *misspecified* and the parameters of the entire model, including the *structural part can be biased* — even if the structural part is correctly specified (Rosseel and Loh 2021).

We will see some strategies for *diagnosing violated assumptions* when we discuss re-specification and modification indices.

Identification

Once we start bringing in latent variables, the topic of identification starts to play a more prominent role. Remember, we can only estimate as many unknown parameters as we have pieces of information in the observed covariance matrix (and mean vector).

Let us reinspect the covariance matrix of the observed items:

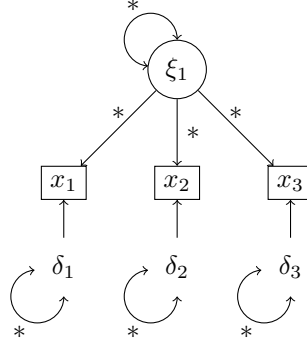
```
cov(df[, c("px06", "px07", "px10")], use = "complete.obs")
```

```
##           px06      px07      px10
## px06 2.0009400 0.7660920 0.4149269
## px07 0.7660920 1.3608021 0.3598991
## px10 0.4149269 0.3598991 0.8423078
```

There are three items and so we have three variances and three nonredundant covariances for *six pieces of observed information*. Notice this follows the $p(p+1)/2$ formula, where p is the number of observed variables, with $3(3+1)/2 = 12/2 = 6$.

And for our model, we have to estimate three factor loadings, $\lambda_{11}, \lambda_{21}, \lambda_{31}$, one exogenous variance, ϕ_1 , the variance of ξ_1 , and three error variances, $\theta_1, \theta_2, \theta_3$. These are shown in Figure 2 marked by the stars. This makes *seven parameters to estimate*, which is one too many! In other words, we have -1 degrees of freedom and we will not be able to properly estimate the model.

Figure 2: Three-item measurement model, parameters to estimate



A factor model with three indicators is not identified unless we put at least one additional restriction on the model. The usual solution is to either fix one factor loading to one, or set the variance of the latent variable to a fixed value, also usually one. *These approaches are equivalent.* The first solution is the most commonly used, and it is called the *marker-item method*. By either fixing a factor loading to one, or setting the variance of the latent variable to one, we solve the problem of the three-indicator model being under-identified (though a just-identified model is also not optimal), and we give the latent variable as scale which, as it is unobserved, it does not have intrinsically.

Generally, there is no intuitive explanation as to the relationship between the variance of the latent variable and the indicators. If we fix the initial factor loading to one, it is tempting to think that the variance of the latent variable is equal to the variance of the observed marker-item, but this is not the case, since we have to account for the error variance. The variance of the latent variable is

$$\begin{aligned}\text{Var}(\mathbf{x}) &= \Sigma(\boldsymbol{\theta}) \\ &= \boldsymbol{\lambda}_x \boldsymbol{\Phi} \boldsymbol{\lambda}_x^\top + \boldsymbol{\Theta}_\delta \\ \boldsymbol{\lambda}_x \boldsymbol{\Phi} \boldsymbol{\lambda}_x^\top &= \text{Var}(\mathbf{x}) - \boldsymbol{\Theta}_\delta \\ \boldsymbol{\Phi} &= (\boldsymbol{\lambda}_x^\top \boldsymbol{\lambda}_x)^{-1} \boldsymbol{\lambda}_x^\top (\text{Var}(\mathbf{x}) - \boldsymbol{\Theta}_\delta) \boldsymbol{\lambda}_x (\boldsymbol{\lambda}_x^\top \boldsymbol{\lambda}_x)^{-1}\end{aligned}$$

which is a rather ugly expression that does not tell us much. But it bears repeating that setting one of the factor loadings to one or setting the variance of the latent variable(s) to one is equivalent, regardless. The following scalar expressions are essentially the same as the matrix expressions above (with the inverse being equivalent to division, etc.)

$$\begin{aligned}x_j &= \lambda_{jk} \xi_k + \delta_j \\ \text{Var}(x_j) &= \lambda_{jk}^2 \text{Var}(\xi_k) + \text{Var}(\delta_j).\end{aligned}$$

From here, we can set $\lambda_{jk}^2 = 1$ and solve for $\text{Var}(\xi_k)$ which will be $\text{Var}(x_j) - \text{Var}(\delta_j)$, or equivalently, we can set $\text{Var}(\xi_k) = 1$ and solve for λ_{jk}^2 which also give $\text{Var}(x_j) - \text{Var}(\delta_j)$. This means whichever approach we choose, the model will compensate and the rest of the results will be identical.

CFA in lavaan

We use the `=~` or ‘measured by’ operator to specify our measurement model. That is, we create a latent variable, which we can call `xi1` that is measured by the three indicators, `px06`, `px07` and `px10`. We will use the marker-item method and fix the factor loading for `px06` to 1.0 (we could, however, fix any one of the three factor loadings to one, it does not have to be the first). We will further use labels for the other two

factor loadings (12, 13 for ‘loading 2,’ ‘loading 3’). These are entirely optional, but it can help us quickly find what we are looking for in the summary output, as well as help us easily extract things from the fitted object.

The indicators are all endogenous variables, so when we specify their variances (also optional), we are referring to the unexplained part, i.e., the measurement error. We use labels here as well, `t1` for ‘theta 1,’ etc. Finally, we can add the variance of the latent variable, which is also optional. This variance is the only exogenous variance in the model.

```
library(lavaan)

# Specify the model
cfa1 <- '
# Measurement model
xi1 =~ 1*px06 + 12*px07 + 13*px10
# Measurement error variances
px06 ~~ t1*px06
px07 ~~ t2*px07
px10 ~~ t3*px10
# Latent variable variances
xi1 ~~ phi1*xi1
'
```

Note again that all we actually need to write is:

```
cfa1 <- '
xi1 =~ px06 + px07 + px10
'
```

since the entire rest is taken care of by default settings. There is a dedicated ‘wrapper’ in `lavaan` for CFA models called `cfa()`. However, at the moment, it seems like the default settings of `cfa()` are identical to those of the `sem()` wrapper we have used up until now.

```
# Fit the model using cfa() - though it does differ from sem()
cfa1.fit <- cfa(model = cfa1, data = df)
```

We will want to use the optional arguments `standardized = TRUE` because the standardized factor loadings are more useful in telling us about the quality of the measurement model.

```
# Request summary with standardized estimates
summary(cfa1.fit, standardized = TRUE)
```

```
## lavaan 0.6-8 ended normally after 19 iterations
##
##      Estimator                      ML
##      Optimization method          NLMINB
##      Number of model parameters          6
##
##                               Used      Total
##      Number of observations          3192      3477
##
## Model Test User Model:
##
```

```

##      Test statistic                0.000
##      Degrees of freedom              0
##
## Parameter Estimates:
##
##      Standard errors                Standard
##      Information                    Expected
##      Information saturated (h1) model Structured
##
## Latent Variables:
##      Estimate  Std.Err  z-value  P(>|z|)  Std.lv  Std.all
##      xi1 =~
##      px06      1.000
##      px07      (12)  0.867    0.049   17.869    0.000    0.815    0.699
##      px10      (13)  0.470    0.025   18.435    0.000    0.441    0.481
##
## Variances:
##      Estimate  Std.Err  z-value  P(>|z|)  Std.lv  Std.all
##      .px06      (t1)  1.117    0.055   20.460    0.000    1.117    0.559
##      .px07      (t2)  0.696    0.039   17.686    0.000    0.696    0.512
##      .px10      (t3)  0.647    0.019   33.663    0.000    0.647    0.769
##      xi1      (phi1)  0.883    0.063   14.093    0.000    1.000    1.000

```

Criteria for evaluating measurement models

In CFA models, we are mainly interested in the relationships between the indicators and the latent cause. These are shown in the *factor loadings*. The formal criteria for evaluating a measurement model involve looking at the *validity* and *reliability* of the measure.

Note that the topics of validity and reliability are not entirely separate from the overall question of whether the hypothetical model is supported by the data, which we will discuss in the next section.

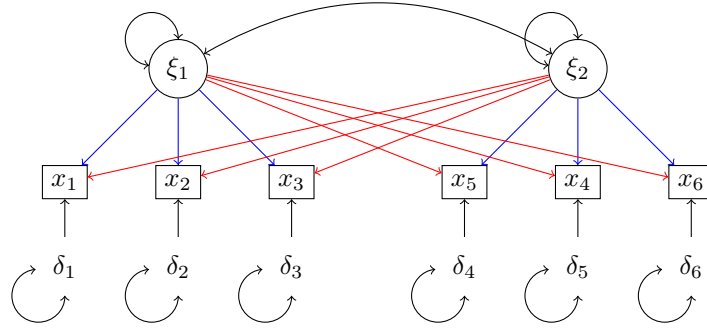
Validity

When we talk about validity, we are referring, roughly speaking, to whether the instrument measures what it is supposed to. There are several types of validity or rather types of tests for assessing validity ([Bollen 1989](#)):

- *Content validity*: a qualitative assessment as to whether all theoretically relevant aspects of the concept are covered (e.g., socioeconomic status encompasses income, education and occupation)
- *Criterion validity*: the use of another measure (the criterion) to ensure a correlation exists where it should (e.g., correlation between self-reported income (measure) and some administrative validation data (criterion))
- *Construct validity*: do the hypothesized associations between a measure and other observed variables exist empirically (e.g., do the correlations between our indicators for xenophobia display the proper sign, magnitude and statistical significance?)

Content validity is dependent on theoretical arguments and there rarely exists a consensus as to what exactly a concept entails, and we often lack an appropriate criterion for assessing criterion validity. But we can always inspect construct validity, which actually entails checking for two different signs of construct validity: *convergent* and *divergent* validity. Convergent construct validity means the indicators load properly on the

Figure 3: Convergent (blue) and divergent (red) construct validity



intended latent variable. Divergent construct validity means the indicators do not load on any unintended latent variables.

With convergent validity, we are interested in answering the questions (see [Bollen 1989, 256](#)):

- Do the factor loadings display the *proper sign*? E.g., if a positive association is hypothesized, the factor loadings should be positive, as well.
- What are the *magnitudes* of the factor loadings? Are they substantial in size and consistent with theory and previous empirical findings?
- Are the factor loadings *statistically significant*? If not, then one or more of the indicators is obviously not caused by the latent factor.

```
...
##
## Latent Variables:
##
##          Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   xi1 =~
##     px06          1.000
##     px07   (12)    0.867    0.049   17.869    0.000    0.815    0.699
...
```

In our case, the signs of the factor loadings are consistent with our hypothesis. Each are positive, which means a positive association between the factor and the items (see column **Estimate**). If xenophobia increases, the agreement with the rather xenophobic statements also increases. This is unsurprising, however, and we would have known this would be the result just by looking at the covariance matrix. The statistical significance is also given: each are significantly different from zero (see column $P(>|z|)$). Note that we do not get a test of significance for the factor loading we fixed to one.

Now, while the magnitudes of the factor loadings are most interesting, we can first look at the unstandardized loadings and remind ourselves of what they mean. Take **px07** with an unstandardized factor loading of 0.867. This means

$$x_2 = 0.867\xi_1 + \delta_2$$

$$\mathbb{E}(x_2|\xi_1) = 0.867\xi_1$$

that is, the expected value of x_2 increases by 0.867 for every unit increase of the latent factor ξ_1 . This is, on its own, not especially informative, but we would expect that if the items were good indicators of the latent variable, that they would have unstandardized factor loadings that were *similar*. Obviously, if one indicator loads with 0.9 on the latent variable and another with 2.5, but both indicators are measured on the same

scale, then there is a large discrepancy and the expected values of the two indicators do not behave in a similar fashion as the latent variable changes.

In fact, the standardized factor loadings are arguably the most important components of the CFA model. These vary between -1 and $+1$, with -1 indicating a perfect negative correlation, $+1$ a perfect positive correlation, and 0 no correlation at all. That is, if we call λ_{jk}^* the standardized loading, then by covariance algebra we see

$$\begin{aligned}\text{Cor}(x_j, \xi_k) &= \text{Cor}(\lambda_{jk}^* \xi_k + \delta_j, \xi_k) \\ &= \lambda_{jk}^* \text{Cor}(\xi_k, \xi_k) \\ \hat{\lambda}_{jk}^* &= \text{Cor}(x_j, \xi_k)\end{aligned}$$

since $\text{Cor}(\xi_k, \xi_k) = 1$. That is, the standardized factor loadings give us the (partial) correlation between the variables, both observed and latent. We usually say a measurement model displays *adequate construct validity* if the standardized factor loadings are all above 0.7 , though some are satisfied with as low as 0.5 .

Why do we choose 0.7 as a rough cutoff for an acceptable standardized factor loading? Because if we look at the standardized variance of an indicator, then the squared standardized factor loading represents the percent of variance attributable to the latent variable. If we use x_j^* and ξ_k^* to represent standardized variables with a mean of zero and variance of one, then

$$\begin{aligned}\text{Var}(x_j^*) &= \text{Var}(\lambda_{jk}^* \xi_k^* + \delta_j^*, \lambda_{jk}^* \xi_k^* + \delta_j^*) \\ 1 &= \lambda_{jk}^{2*} \text{Var}(\xi_k^*) + \text{Var}(\delta_j^*) \\ 1 &= \lambda_{jk}^{2*} + \text{Var}(\delta_j^*)\end{aligned}$$

which means $\lambda_{jk}^{2*} = \text{Var}(x_j^*) - \text{Var}(\delta_j^*) = 1 - \text{Var}(\delta_j^*)$. A standardized factor loading of 0.7 means that $0.7^2 = 0.49$ or roughly 50% of the variance of the indicator is due to the latent variable. It also means that the other 50% is independent of the latent variable! A standardized factor loading of 0.5 means that only $0.5^2 = 0.25$ or 25% of the variance is due to the latent variable, the other 75% is error variance.

In our case, the standardized factor loadings range between 0.506 and 0.707 , which are adequate but not excellent. Because the relationships between the latent variable and the indicators are of the proper sign, arguably adequate magnitude and significant, we could say the model displays *convergent construct validity* or just *convergent validity*.

We cannot empirically test divergent validity in our model because we do not have a second latent construct. From a hypothetical standpoint, however, divergent validity means that any other latent variables should not also cause the indicators of ξ_1 . This is shown in Figure 3. Any hypothesized indicators of a specific latent variable should load on that variable (blue) and *should not* load on any other latent variable (red). We can investigate divergent validity in more detail when we have introduced further latent variables and discuss modification indices.

Reliability

Another aspect to discuss is the *reliability*, though it is sometimes difficult to disentangle reliability from construct validity since they can be seen as essentially just re-expressions of the same thing. Namely, reliability refers to the measurement error, and we say an instrument is reliable if the error (difference between true value and measured value) is low. In this sense, it seems sensible to look at the *proportion of error*, i.e., the percentage of variance of the indicator that is due to error. However, the percentage of error and the standardized factor loadings are linked deterministically.

In other words, by rearranging the last set of equations, we see

$$\begin{aligned}\text{Var}(x_j^*) &= \lambda_{jk}^{2*} \text{Var}(\xi_k^*) + \text{Var}(\delta_j^*) \\ 1 &= \lambda_{jk}^{2*} + \text{Var}(\delta_j^*) \\ \text{Var}(\delta_j^*) &= 1 - \lambda_{jk}^{2*},\end{aligned}$$

in other words, the error variance is given by one minus the standardized factor loading.

For example, the standardized factor loading for indicator x_3 or `px10` is 0.47. This means $1 - 0.47^2 = 1 - 0.221 = 0.779$ of the variance of the variable is *unexplained*. This is the standardized error variance. In order for a measurement model to be reliable, the error variances should not be ‘too large,’ where we can use the rules of thumb noted above (e.g., are we satisfied with 50% error variance or even 75% error variance?) as guidance. Again, the error variances are listed under **Variances**, where the parameters with the . in front of them represent endogenous, i.e., error variances. The standardized variances are listed under **Std.all**.

```
...
##
## Variances:
##
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##   .px06      (t1)   1.117   0.055  20.460   0.000   1.117   0.559
##   .px07      (t2)   0.696   0.039  17.686   0.000   0.696   0.512
##   .px10      (t3)   0.647   0.019  33.663   0.000   0.647   0.769
...
```

More formally, however, we say the reliability of our measurement of the latent variable is given by

$$\text{REL} = \frac{k \cdot r}{1 + (k - 1)r}$$

where k is the number of indicators and r is the average correlation between the three (Urban and Mayerl 2014, 20). By covariance algebra (or tracing), we see that the correlation between any two indicators is given by the product of the standardized factor loadings. So, in this case, we have $\lambda_{11}^* = 0.664$, $\lambda_{21}^* = 0.699$ and $\lambda_{31}^* = 0.481$. So, the model-implied correlation between x_1 and x_2 is $0.664 \cdot 0.699 = 0.464$, between x_1 and x_3 it is $0.664 \cdot 0.481 = 0.319$ and between x_2 and x_3 it is $0.699 \cdot 0.481 = 0.336$. The average correlation is $r = (0.464 + 0.319 + 0.336)/3 = 0.373$. So, we have

$$\begin{aligned} \text{REL} &= \frac{3 \cdot 0.373}{1 + (3 - 1)0.373} \\ &= 0.641. \end{aligned}$$

There is a function in the so-called **semTools** package for calculating the reliability of a measurement model with **semTools::reliability**. The package **semTools** provides functions to extend the functionality of **lavaan** (see <https://cran.r-project.org/web/packages/semTools/index.html>). We use **alpha** to calculate the Cronbach’s Alpha reliability coefficient (with only a slight discrepancy compared to the statistic calculated above by hand):

```
library(semTools)
semTools::reliability(cfa1.fit, what = "alpha")
```

```
##           xi1
## alpha 0.6344807
```

There are no hard rules concerning adequate reliability, but the general rules of thumb suggest a reliability > 0.7 is acceptable, > 0.8 is good, etc. Here, with a reliability measure of only 0.649, coupled with the low standardized factor loadings, we might not be completely satisfied with the measurement model.

Model fit

Remember we defined the residual matrix as $\mathbf{S} - \mathbf{\Sigma}(\hat{\boldsymbol{\theta}})$, i.e., the observed covariance matrix minus the model-implied covariance matrix based on the estimated parameters. We can think of the residual matrix as telling us about the *fit* of the model.

In other words, when the model-implied covariance matrix closely matches the observed covariance matrix, then the magnitude of the discrepancies is small and we say the model ‘fits well.’ When the model-implied matrix is far off from the observed matrix, the magnitude of the discrepancies in the residual matrix is large and we say the model ‘fits poorly.’

Roughly speaking, fit tells whether our hypothesized model is supported by the data, or not. In terms of a CFA model, we might hypothesize that there is a single common cause for all the indicators, and that the rest of the unique variance is simply random measurement error. If this is the case, then the factor loadings should be sufficient to account for essentially all of the covariance between the items. If, however, there is another source of covariance between some of the indicators, then we will not have accounted for all the sources of covariance and our model will not fit optimally. This misfit tells us there may be other sources of covariance between some of the indicators and we may be forced to reject our hypothesized model.

Remember, our hypothesis is

$$\begin{aligned}\mathbf{S} &= \mathbf{\Sigma}(\boldsymbol{\theta}), \text{ or} \\ \mathbf{S} - \mathbf{\Sigma}(\boldsymbol{\theta}) &= \mathbf{0}\end{aligned}$$

and we are interested in 1) whether our model-implied matrix differs from the observed covariance matrix and to what extent, and 2) whether the discrepancy we observe is likely systematic or due to sampling error and can thus be ignored.

Thus stated, we have

$$\begin{aligned}H_0 : \mathbf{S} &= \mathbf{\Sigma}(\boldsymbol{\theta}) \\ H_1 : \mathbf{S} &\neq \mathbf{\Sigma}(\boldsymbol{\theta}).\end{aligned}$$

So, unlike the usual case in hypothesis testing, we *do not want to reject the H_0* . We would prefer it if our hypothesized model is supported by the data. This means that in terms of inference, we prefer a *nonsignificant* result, so that we can say there is insufficient evidence to reject the H_0 and we can tentatively accept that the model is supported by the data.

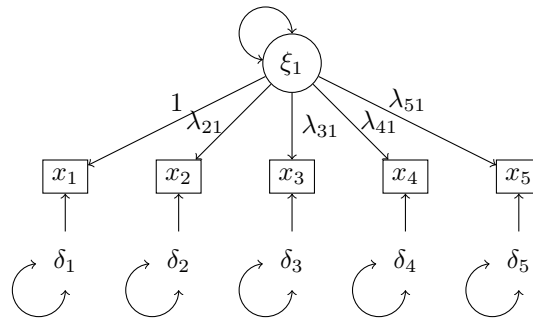
Note, however, that model fit can only be evaluated in *over-identified models*. As discussed previously, just-identified models have 0 degrees of freedom and an exact solution is always available so that $\mathbf{S} - \mathbf{\Sigma}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$. This means that the fit of the example above cannot be evaluated, since with only three indicators, the model is not over-identified without placing further restrictions on the model, e.g., fixing the error variances across variables to be equal, or fixing factor loadings to be equal. And without strong theoretical arguments, it is difficult to justify placing such restrictions.

Fit measures

We will now discuss various measures for model fit. To do so, we will first expand the example above to five indicators, so that our model is over-identified and we can evaluate the fit. To the three indicators listed above, we will add:

- x_4 : “Should immigrants be forced to assimilate? [recoded]” (pa09r)
- x_5 : “Should we prohibit the influx of asylum-seekers? [recoded]” (pa19r).

Figure 4: Five-item measurement model



The new model with five indicators is displayed in Figure 4. Note we will be using the marker-item method, fixing the factor loading of x_1 to one.

We can specify the new model as follows:

```

# Specify the model
cfa2 <- '
# Measurement model
xi1 =~ 1*px06 + l2*px07 + l3*px10 + l4*pa09r + l5*pa19r
# Measurement error variances
px06 ~~ d1*px06
px07 ~~ d2*px07
px10 ~~ d3*px10
pa09r ~~ d4*pa09r
pa19r ~~ d5*pa19r
# Latent variable variances
xi1 ~~ phi1*xi1
'
# Fit the model
cfa2.fit <- cfa(model = cfa2, data = df)

```

When we request a summary of the model results, we will now add `fit.measures = TRUE` to the summary call, where a selection of common fit measures will be listed besides the chi square statistic:

```
summary(cfa2.fit, standardized = TRUE, fit.measures = TRUE)
```

```

...
## Model Test User Model:
##
##   Test statistic                103.133
##   Degrees of freedom              5
##   P-value (Chi-square)           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                3968.850
##   Degrees of freedom            10
##   P-value                       0.000
##
## User Model versus Baseline Model:

```

```

##
##   Comparative Fit Index (CFI)                0.975
##   Tucker-Lewis Index (TLI)                  0.950
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)            -22779.056
##   Loglikelihood unrestricted model (H1)      NA
##
##   Akaike (AIC)                             45578.111
##   Bayesian (BIC)                           45638.679
##   Sample-size adjusted Bayesian (BIC)      45606.905
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                     0.079
##   90 Percent confidence interval - lower    0.066
##   90 Percent confidence interval - upper    0.092
##   P-value RMSEA <= 0.05                    0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                     0.034
...

```

In the next section, we will discuss a number of the fit measures shown in the summary.

Chi square

The most central measure of model fit is the chi square, χ^2 , statistic. It is given by

$$\chi^2 = (n - 1)f$$

where n is the number of observations and f is the value of the fitting function, e.g., F_{ML} evaluated with the estimates $\hat{\theta}$ plugged in. The reason why this expression gives the chi square statistic is not particularly interesting (see [Bollen 1989, 263–69](#)), and we can find the value of chi square from the model output.

The chi square statistic tells us the magnitude of the discrepancy between the observed and model-implied covariance matrices (and mean vectors, if we include the mean structure). So, recalling the main hypotheses:

$$H_0 : \mathbf{S} = \mathbf{\Sigma}(\boldsymbol{\theta})$$

$$H_1 : \mathbf{S} \neq \mathbf{\Sigma}(\boldsymbol{\theta})$$

we would reject the H_0 if chi square is significantly larger than zero.

In this case, we have:

```

...
## Model Test User Model:
##
##   Test statistic                103.133
##   Degrees of freedom            5
##   P-value (Chi-square)          0.000
...

```


In this example, the chi square statistic is 103.13 with 5 degrees of freedom and a p-value less than 0.001. The chi square on its own is not very informative, because it has no upper limit but the fact that it is significantly significant tells us we should reject the H_0 and that our model is not consistent with the data.

Some suggest, however, that the chi square statistic tends to always be significant with large sample sizes (e.g., according to David Kenny, with about 400 or more observations, see <http://www.davidakenny.net/cm/fit.htm>, and Urban and Mayerl 2014, 91f.). This is a point of contention, and some would not agree with this suggestion, see any of the many discussions regarding model fit on SEMNET. Still, most practitioners accept that the chi square test tends to be significant, and this should not be seen as an KO criteria. Rather, the chi square statistic should be interpreted along with other fit measures, which will be discussed in the next section.

Absolute fit measures

Some alternative measures of model fit have been suggested that also work on the logic of comparing the model-implied and the observed covariance matrices.

The *Root Mean Square Error of Approximation* (RMSEA) is based off of the chi square statistic but includes a penalty for complexity by dividing the chi square by the degrees of freedom. It is calculated as

$$RMSEA = \sqrt{\frac{\chi^2 - df}{df(n - 1)}}$$

where df is the degrees of freedom and n is the number of observations. In this case, we have

$$\begin{aligned} RMSEA &= \sqrt{\frac{103.13 - 5}{5(3155 - 1)}} \\ &= 0.079 \end{aligned}$$

We can calculate RMSEA by hand easily to verify:

```
# Extract chisq, df and n from cfa2.fit
chisq <- fitMeasures(cfa2.fit, "chisq")
df     <- fitMeasures(cfa2.fit, "df")
n      <- fitMeasures(cfa2.fit, "ntotal")

# Calculate rmsea by hand
rmsea <- sqrt((chisq - df) / (df * (n - 1))); round(rmsea, 3)
```

```
## [1] 0.079
```

The smaller the RMSEA is, the better the model fit. Take the following fictitious example, where we compare two models with equal sample sizes and chi square statistics, but with differing degrees of freedom. RMSEA favours the model with more degrees of freedom (less complex) with a lower value:

```
n <- 100
chisq <- 100
df1 <- 5
df2 <- 10

rmsea1 <- sqrt(chisq / df1) / sqrt(df1 * (n - 1)); rmsea1
rmsea2 <- sqrt(chisq / df2) / sqrt(df2 * (n - 1)); rmsea2
```

```
## [1] 0.2010076
## [1] 0.1005038
```

Normally, as a rule of thumb, we say models where $RMSEA < 0.05$ fit well, though values up to 0.08 are often seen as adequate. In our case, again $RMSEA = 0.08$ which is above the ‘cutoff’ of 0.08 for an adequate fitting model.

```
...
## Root Mean Square Error of Approximation:
##
##   RMSEA                                0.079
##   90 Percent confidence interval - lower    0.066
##   90 Percent confidence interval - upper    0.092
##   P-value RMSEA <= 0.05                    0.000
...
```

RMSEA also normally comes with a significance test, shown below as **90 Percent confidence interval - lower** and **90 Percent confidence interval - upper**. These tell us where 90% of the RMSEA confidence intervals would be in repeated samples. Here the confidence interval is between 0.07 and 0.09 and the probability that the ‘true’ RMSEA is less than 0.05 is essentially zero — the model fits significantly worse than that.

The *Standardized Root Mean Square Residual* (SRMR) works on a slightly different logic but is similar to the RMSEA at least in terms of its interpretation.

It is given by

$$SRMR = \sqrt{\frac{1}{t} \sum_{i \leq j} \epsilon_{ij}^2}$$

where ϵ_{ij} are the *standardized residuals*, i.e., the elements of the residual matrix based on the difference between the observed and model-implied *correlation* (not covariance!) matrices and $t = p(p + 1)/2$ is the number of nonredundant elements. Note that the observed and model-implied matrices are symmetric, so we are essentially calculating the *square root of the average squared standardized residual*. It is best to just work SRMR out in R, because it is unnecessarily tedious to show the calculation in the equation above:

```
# S is the observed correlation matrix - not covariance matrix!
S  <- round(cor(df[, c("px06", "px07", "px10", "pa09r", "pa19r")], use = "complete.obs"), 3)
# Sig is the model-implied correlation matrix, as well
Sig <- lavInspect(cfa2.fit, "cor.ov")
# R is the residual matrix
R  <- S - Sig

# We say t is the number of nonredundant elements
t  <- 5 * (5 + 1) / 2

# We take the sum of R^2 and then multiply by a boolean matrix with
# TRUE (=1) if the element is on or below the diagonal and FALSE (=0)
# if it is above the diagonal. This gives only the nonredundant elements.
srmr <- sqrt((1 / t) * sum(R^2 * lower.tri(R, diag = TRUE))); round(srmr, 3)
```

```
## [1] 0.034
```

SRMR can be found at the bottom of the fit measures output,

```
...
## Standardized Root Mean Square Residual:
##
##    SRMR                                0.034
...
```

Similar to RMSEA, we normally say that a value < 0.05 indicates a good fit, < 0.08 is satisfactory and anything above that is poor. Here, with a value of 0.034, we would conclude the model fit is indeed good. This contradicts the impressions given by both the chi square and RMSEA, which is not unusual. For exactly this reason, it is good to *report more than one fit measure* because there are advantages and drawbacks to each. It is also sensible to report *different types of fit measures*, where the *comparative fit measures* are the *second main type* besides the absolute ones.

Comparative fit measures

Comparative fit measures work on the logic of comparing the model-implied covariance matrix not to the observed covariance matrix (chi square and absolute measures), but rather to the *worst-fitting model possible*. The worst-fitting model is a model is one where the model variables are all mutually uncorrelated. For example, the ‘baseline’ or ‘null model’ would be:

```
# S is the observed covariance matrix
S <- lavInspect(cfa2.fit, "obs")$cov; S
```

```
##      px06  px07  px10  pa09r pa19r
## px06  2.008
## px07  0.769 1.358
## px10  0.415 0.365 0.843
## pa09r 0.716 0.381 0.163 1.078
## pa19r 1.166 0.629 0.380 0.619 1.764
```

```
# The 'null model' would be S with zeros everywhere except the diagonal
round(diag(diag(S)), 3)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 2.008 0.000 0.000 0.000 0.000
## [2,] 0.000 1.358 0.000 0.000 0.000
## [3,] 0.000 0.000 0.843 0.000 0.000
## [4,] 0.000 0.000 0.000 1.078 0.000
## [5,] 0.000 0.000 0.000 0.000 1.764
```

We can specify a ‘null model’ in lavaan:

```
# Specify the 'null model'
cfa0 <- '
# Just estimate the variances
px06 ~~ px06
px07 ~~ px07
px10 ~~ px10
pa09r ~~ pa09r
pa19r ~~ pa19r
# Set covariances to zero
```

```

px06 ~~ 0*px07 + 0*px10 + 0*pa09r + 0*pa19r
px07 ~~ 0*px10 + 0*pa09r + 0*pa19r
px10 ~~ 0*pa09r + 0*pa19r
pa09r ~~ 0*pa19r
,
cfa0.fit <- cfa(model = cfa0, data = df)
# summary(cfa0.fit) # Uncomment to show summary

```

The *Comparative Fit Index* (CFI) is given by

$$CFI = \left| \frac{(\chi_0^2 - df_0) - (\chi_m^2 - df_m)}{\chi_0^2 - df_0} \right|$$

where χ_0^2 and df_0 are the chi square statistic and degrees of freedom of the null model, respectively and χ_m^2 and df_m are the same from the our actual model, in this case `cfa2.fit`.

We can calculate CFI by hand:

```

# Get chisq and df from both models, cfa0.fit and cfa2.fit
# Note there are attributes attached to the objects, so we use
# as.numeric() to strip those attributes.

# Our model
chisq_m <- as.numeric(fitMeasures(cfa2.fit, "chisq"))
df_m    <- as.numeric(fitMeasures(cfa2.fit, "df"))

# Null model
chisq_0 <- as.numeric(fitMeasures(cfa0.fit, "chisq"))
df_0    <- as.numeric(fitMeasures(cfa0.fit, "df"))

# CFI
cfi <- abs((chisq_0 - df_0 - chisq_m - df_m) / (chisq_0 - df_0)); cfi

```

```
## [1] 0.9726857
```

```

...
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                0.975
##   Tucker-Lewis Index (TLI)                  0.950
...

```

CFI essentially tells us how much better our model fits compared to the worst possible model. It can vary between 0 (no improvement) and 1 (perfect fit). We want our model to represent a substantial improvement compared to the worst possible model, and values for CFI > 0.95 are considered good. Here, we have a CFI of approximately 0.97, which would indicate good fit.

Other comparative measures include the *Tucker-Lewis Index* (TLI) which will tend to be very similar to the CFI. Since they tend to be so similar, only one is usually reported (usually the CFI).

A full list of the most common fit measures, both absolute and comparative, can be accessed using `fitMeasures()`, like so:

```
# Get full list of fit measures
fitMeasures(cfa2.fit)
```

```
##          npar          fmin          chisq          df
##        10.000         0.016        103.133         5.000
##        pvalue    baseline.chisq    baseline.df    baseline.pvalue
##          0.000        3968.850         10.000         0.000
##          cfi          tli          nnfi          rfi
##        0.975         0.950         0.950         0.948
##          nfi          pnfi          ifi          rni
##        0.974         0.487         0.975         0.975
##          logl    unrestricted.logl          aic          bic
##       -22779.056             NA        45578.111        45638.679
##          ntotal          bic2          rmsea    rmsea.ci.lower
##        3155.000        45606.905         0.079         0.066
##    rmsea.ci.upper    rmsea.pvalue          rmr    rmr_nomean
##          0.092         0.000         0.036         0.036
##          srmr    srmr_bentler    srmr_bentler_nomean    crmr
##          0.034         0.034         0.034         0.042
##    crmr_nomean    srmr_mplus    srmr_mplus_nomean    cn_05
##          0.042         0.034         0.034        339.663
##          cn_01          gfi          agfi          pgfi
##        462.512         0.987         0.961         0.329
##          mfi          ecvi
##          0.985         0.039
```

References

- Bollen, Kenneth. 1989. *Structural Equations with Latent Variables*. New York: John Wiley & Sons.
- Frisch, Ragnar, and Frederick V. Waugh. 1933. “Partial Time Regressions as Compared with Individual Trends.” *Econometrica* 1 (4): 387–401. <https://doi.org/10.2307/1907330>.
- Reinecke, Jost, and Andreas Pöge. 2010. “Strukturgleichungsmodelle.” In *Handbuch der sozialwissenschaftlichen Datenanalyse*, edited by Christof Wolf and Henning Best. Wiesbaden: VS Verlag für Sozialwissenschaften.
- Rosseel, Yves, and Wen Wei Loh. 2021. “A Structural After Measurement (SAM) Approach to SEM.” <https://osf.io/xme9g/>.
- Urban, Dieter, and Jochen Mayerl. 2014. *Strukturgleichungsmodellierung: Ein Ratgeber für Die Praxis*. Wiesbaden: Springer VS.