Fitting the Common Factor Model

PSYn5440 – Introduction to Factor Analysis

Week 6

The Common Factor Model

- **In the past two weeks, we introduced the Common Factor Model in multiple forms.**

- **First, we introduced the data model:**

$$\boldsymbol{x = \mu + \Lambda z + u}$$

The Common Factor Model

- After making some assumptions, we derived a covariance structure model

- For correlated factors:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}' + \boldsymbol{D}_{\psi}$$

- For uncorrelated factors:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{D}_{\psi}$$

The Common Factor Model

- And we have seen how the covariance structure model can be transformed into a correlation structure model.

- For correlated factors:

$$\mathbf{P} = \boldsymbol{\Lambda}^{*}\boldsymbol{\Phi}\boldsymbol{\Lambda}^{*\prime} + \boldsymbol{D}_{\psi}^{*}$$

- For uncorrelated factors:

$$\mathbf{P} = \boldsymbol{\Lambda}^{*}\boldsymbol{\Lambda}^{*\prime} + \boldsymbol{D}_{\psi}^{*}$$

The Common Factor Model

- You learned a LOT already!
- More than a majority of factor analysis practitioners know about the model ☺ (pretty sad, huh?)

Fitting the model

- One important thing to note is that these models are intended for a **population** – they are population models, describing how stuff works in a population.

- Anyway, at the beginning we learned that there are two sides to factor analysis – theory and methodology.
- What we have covered so far was the **theory** (the model itself)
- Now, we will focus on the **methodology** (how to fit the model on data / how to estimate the unknowns in the model)

Fitting the model

- More specifically, we will focus on the **theoretical basis** for fitting the model. Later on in the course, we will cover the actual thing in practice (software and examples).

- A **model** represents some hypothesized structure of data. Different **methods** are available for fitting the model to data and obtaining estimates of model parameters (the elements in model matrices) and providing us with information on how well the model **fits** the data.

Fitting the model

- For the sake of argument, we will consider the hypothetical scenario where the population correlation matrix **P** is known, and the model holds exactly in the population (i.e., the model explains **P** perfectly)

- This will never ever be the case in practice, but it's a better starting point to begin understanding the principles.

- Later, we will drop these assumptions, no worries.

The population correlation matrix

- Last time we talked about the case where the model holds exactly **and** the common factors are uncorrelated (orthogonal). In such a case, the population correlation matrix **P** has the structure:

$$\mathbf{P} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{D}_{\psi}$$

...or, alternatively:

$$\mathbf{P} - \boldsymbol{D}_{\psi} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}'$$

- Strictly speaking, we should use $\boldsymbol{\Lambda}^*$ and $\boldsymbol{D}_{\psi}^*$, but let's omit the stars now for convenience.

Rotational indeterminacy

- We have a bit of a "problem".

- If we can find a single *p* x *m* factor loading matrix $\boldsymbol{\Lambda_1}$ such that $\mathbf{P} = \boldsymbol{\Lambda\Lambda'} + \boldsymbol{D_\psi}$, and if *m* > 2 (i.e., we have two or more factors), then there are infinitely many other *p* x *m* factor loading matrices such that $\mathbf{P} = \boldsymbol{\Lambda_1\Lambda_1'} + \mathbf{D_\psi} = \boldsymbol{\Lambda_2\Lambda_2'} + \mathbf{D_\psi} = \boldsymbol{\Lambda_3\Lambda_3'} + \mathbf{D_\psi} = \cdots$

- Am I kidding? Nope. Let me show you.

Rotational indeterminacy

- Suppose I'm not wrong and it indeed holds that:

$$\mathbf{P} = \boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_1' + \mathbf{D}_{\boldsymbol{\psi}} = \boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_2' + \mathbf{D}_{\boldsymbol{\psi}}$$

(we're just considering two solutions now, but there are infinitely many)

- In that case, one solution ($\boldsymbol{\Lambda}_2$) has to be linked in some way with the other solution ($\boldsymbol{\Lambda}_1$). To be precise, $\boldsymbol{\Lambda}_2 = \boldsymbol{\Lambda}_1\mathbf{T}$ where $\mathbf{T}$ is a *m* x *m* orthogonal matrix ($\mathbf{T}\mathbf{T}' = \mathbf{I}$)

Rotational indeterminacy

- In that case, one solution (Λ_2) has to be linked in some way with the other solution (Λ_1). To be precise, $\Lambda_2 = \Lambda_1 \mathbf{T}$ where $\mathbf{T}$ is a *m* x *m* orthogonal matrix ($\mathbf{T}\mathbf{T}' = \mathbf{I}$)

$$\Lambda_2\Lambda_2' = \Lambda_1\mathbf{T}(\Lambda_1\mathbf{T})'$$

$$\Lambda_2\Lambda_2' = \Lambda_1\mathbf{T}(\mathbf{T}'\Lambda_1')$$

$$\Lambda_2\Lambda_2' = \Lambda_1\mathbf{T}\mathbf{T}'\Lambda_1'$$

$$\Lambda_2\Lambda_2' = \Lambda_1\mathbf{I}\Lambda_1'$$

$$\Lambda_2\Lambda_2' = \Lambda_1\Lambda_1'$$

- See? Λ_1 and Λ_2 are equally fine solutions.

Rotational indeterminacy

- In other words, if we can find one solution, we can find other alternative solutions. We simply choose any matrix **T** such that **TT' = I** and we define $\boldsymbol{\Lambda_2 = \Lambda_1 T}$
- We've just seen that $\boldsymbol{\Lambda_1}$ and $\boldsymbol{\Lambda_2}$ are equally good solutions, since $\boldsymbol{\Lambda_2 \Lambda_2' = \Lambda_1 \Lambda_1'}$
- This is called rotational indeterminacy. Later, when we learn about rotation, we will see that this describes a procedure used to produce alternative (and equally good) solutions to $\boldsymbol{\Lambda}$.

Rotational indeterminacy

- However, we must resolve this problem somehow if we want to find a single, unique solution for $\mathbf{\Lambda}$ every time we perform a factor analysis.
- In other words, we need to find a criterion for defining this unique solution.
- Luckily, we can arrive at a solution with the help of Eigenvalues and Eigenvectors.

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- **Recall what we have learned about eigenvalues and eigenvectors.**

- **The eigenstructure of a symmetric matrix S is the following:**

$$\mathbf{S} = \mathbf{U}\mathbf{D}_l\mathbf{U}'$$

...where the columns of U are eigen*vectors* and the diagonal elements of $\mathbf{D}_l$ are eigen*values* (this is a diagonal matrix).

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- **So, a symmetric matrix S may be expressed as the product of some matrix ($\mathbf{U}$) which has some vectors for columns (these are the eigenvectors) and some diagonal matrix ($\mathbf{D}_{\boldsymbol{l}}$) which has some values on its diagonal (these are the eigenvalues)**

- **Thus, we can basically decompose a symmetric matrix S into two other matrices (one of them is a diagonal matrix) that, when multiplied in some way, give you back your S matrix.**
- **We're not trying to understand *why* that is or *what exactly* do the two matrices ($\mathbf{U}$ and $\mathbf{D}_{\boldsymbol{l}}$) represent. We don't care at the moment. All we need to know is that this is possible and (as you will see) useful for FA.**

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- So, anyway, how do we obtain the $\boldsymbol{\Lambda}$ in the model $\mathbf{P} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{D}_{\psi}$?
- Let's take a look at **Hypothetical Scenario 1:**
  - You know the true $\mathbf{P}$
  - You know the unique factor variances ($\boldsymbol{D}_{\psi}$), and thus you also know the communalities (diagonal of $\mathbf{P} - \boldsymbol{D}_{\psi}$)
  - The model holds perfectly in the population data

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- In this scenario, obtaining $\boldsymbol{\Lambda}$ is actually quite easy
- You take $(\mathbf{P} - \boldsymbol{D}_{\psi})$, which is sometimes called the *reduced correlation matrix*
- Because $\boldsymbol{D}_{\psi}$ is a diagonal matrix containing unique factor variances, the diagonal of $(\mathbf{P} - \boldsymbol{D}_{\psi})$ will contain (1 – unique factor variances), thus, it will contain the true communalities

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- **Perform the eigenvalue-eigenvector decomposition of $(\mathbf{P} - \boldsymbol{D}_{\psi})$, which will yield some eigenvectors $\mathbf{U}$ and some eigenvalues $\mathbf{D}_{\boldsymbol{l}}$**

- **Order the eigenvalues by size from largest to smallest**
- **The first *m* eigenvalues will be non-zero, the rest will be zero**
- **Keep these non-zero eigenvalues and their associated eigenvectors**

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- Keep only the nonzero eigenvalues in $\mathbf{D}_l$, take their square roots and put them back again into a matrix we will call $\mathbf{D}_l^{1/2}$
- Then, calculate $\boldsymbol{\Lambda} = \mathbf{U}\mathbf{D}_l^{1/2}$
- Magic.
- Remember, for the procedure to work, we need to know $\boldsymbol{D}_{\psi}$, or the unique factor variances.

Eigenvalues, eigenvectors, and $\mathbf{\Lambda}$

- Let's look at an example, using the example data we have seen before.

- The matrix **P** is given as follows:

	PC	VO	AR	MPS
PC	1			
VO	.49	1		
AR	.14	.07	1	
MPS	.48	.42	.48	1

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- **Assume the unique variances are known:**

$$\boldsymbol{D_\psi} = \begin{bmatrix} 0.50 & & & \\ & .51 & & \\ & & .50 & \\ & & & .28 \end{bmatrix}$$

- **So the matrix P with communalities in the diagonal is given by:**

$$\left(\boldsymbol{P} - \boldsymbol{D_\psi}\right) = \begin{bmatrix} .50 & & & \\ .49 & .49 & & \\ .14 & .07 & .50 & \\ .48 & .42 & .48 & .72 \end{bmatrix}$$

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- We can obtain the eigenvalues and eigenvectors of $(\boldsymbol{P} - \boldsymbol{D}_{\psi})$
- The non-zero eigenvalues are:

$$\boldsymbol{D}_{l} = \begin{bmatrix} 1.662 & \\ & .548 \end{bmatrix}$$

- And the corresponding eigenvectors:

$$\mathbf{U} = \begin{bmatrix} .502 & -.386 \\ .461 & -.500 \\ .353 & .731 \\ .641 & .259 \end{bmatrix}$$

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- **The factor loading matrix can be obtained: $\boldsymbol{\Lambda} = \mathbf{U}\mathbf{D}_l^{1/2}$**

$$\boldsymbol{\Lambda} = \begin{bmatrix} .647 & -.285 \\ .594 & -.370 \\ .455 & .541 \\ .826 & .192 \end{bmatrix}$$

- **Wait…that's not the loading matrix I have shown you last time for the example data, is it?**

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- **It's a transformation of the matrix I have shown you earlier, in the rotational indeterminacy sense, $\boldsymbol{\Lambda_2 = \Lambda_1 T}$**

$$\boldsymbol{\Lambda} = \begin{bmatrix} .647 & -.285 \\ .594 & -.370 \\ .455 & .541 \\ .826 & .192 \end{bmatrix} = \begin{bmatrix} .70 & .10 \\ .70 & .00 \\ .10 & .70 \\ .60 & .60 \end{bmatrix} \begin{bmatrix} .848 & -.529 \\ .529 & .848 \end{bmatrix}$$

- **Both $\boldsymbol{\Lambda}$ matrices provide an exact solution to the model. The procedure involving eigen-stuff allowed us to identify the unique solution, though.**

Eigenvalues, eigenvectors, and $\boldsymbol{\Lambda}$

- Okay, so, I have just shown you how to obtain the solution ($\boldsymbol{\Lambda}$) if:
  - You know the population correlation matrix, **P**
  - You know the contents of $\boldsymbol{D_\psi}$, so you know the unique variances or (conversely) the communalities
  - The model holds exactly in the population

- Huh. Putting the "model holds exactly" thing aside, you will never know **P** and you will never know $\boldsymbol{D_\psi}$, so this is a theoretical scenario.
- That's true, but this serves as a basis for things to come.

The communality problem

- As I said, the solution obtained by doing the eigen-decomposition of $\left(\boldsymbol{P}-\boldsymbol{D}_{\psi}\right)$ requires that you know either the unique variances or the communalities (once you know one, you know the other one, right?)

- But we don't know these, since finding out what they are is a part of the problem we face.

- When factor analysis was young, this was called the "Communality problem"

The communality problem

- Many solutions were suggested to the communality problem.
- The one that “won” (was and is the most widely used) was suggested by Louis Guttman in 1940.
- Guttman suggested *squared multiple correlations* (SMCs) as the initial approximations to communalities.

The communality problem

- Just what is a *squared multiple correlation* (SMC)?

- Imagine you have *p* manifest variables. You can try to predict the *j*-th manifest variable from the other *(p - 1)* manifest variables, linear regression-style.

- This prediction will be imperfect. You can correlate these predicted values of the *j*-th manifest variable with the actual values of the variable. What you will get is a correlation coefficient, the **multiple correlation** coefficient. Square it and you get the SMC.

The communality problem

- Guttman has shown that if the factor model applies to the population correlation matrix **P**, then the squared multiple correlation of the *j*-th manifest variable on the other $(p-1)$ manifest variables is the *lower bound* for the communality of the *j*-th manifest variable.

- So, not knowing the contents of $\boldsymbol{D}_{\psi}$, one might approximate the manifest variable communalities with manifest variable SMCs, computed from **P**. These approximations can then be substituted into the diagonal of **P** and one can, again, use the eigenvalue-eigenvector approach on this modified **P** matrix to obtain $\boldsymbol{\Lambda}$.

The communality problem

- However, in order to obtain the population SMCs, we need to know **P** in the first place. Most often, we don't.

- In practice, we can apply the same procedure to a sample correlation matrix, **R**, in order to obtain sample SMCs. Since, in reality, we usually work with sample correlation matrices, let's slowly shift the gear towards thinking more about a sample correlation matrix **R** and less about the population correlation matrix, **P**.

Working with a sample correlation matrix

- So far, we have studied factor analysis limiting ourselves to the ideal scenario in which we know the population correlation matrix, **P**. Moreover, we only considered the case where the model holds exactly in the population.

- Now, let's consider the real world in which we do not have access to **P** but we do have access to **R**. In this real world scenario, we are not even sure the sample correlation matrix **R** is drawn from a population with a correlation matrix **P** for which the model holds.

- As before, let's just consider the uncorrelated / orthogonal model for now.

Working with a sample correlation matrix

- First of all, we should tone down the optimism. In our hypothetical scenarios, we could select $\boldsymbol{\Lambda}$ and $\boldsymbol{D_\psi}$ to reconstruct **P** perfectly:

$$\mathbf{P} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}' + \boldsymbol{D_\psi}$$

- In reality, our *estimates* of $\boldsymbol{\Lambda}$ and $\boldsymbol{D_\psi}$, $\widehat{\boldsymbol{\Lambda}}$ and $\widehat{\boldsymbol{D_\psi}}$, will generally not be able to exactly reproduce our sample correlation matrix **R**:

$$\mathbf{R} \approx \widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}' + \widehat{\boldsymbol{D_\psi}}$$

Working with a sample correlation matrix

- So, what we want is a **parsimonious** model ($m << p$) that provides a relatively good approximation to the data we have observed.

- This degree of approximation (how well the model fits the data) is reflected in the **residual matrix**, defined as $\mathbf{R} - \left(\widehat{\mathbf{\Lambda}}\widehat{\mathbf{\Lambda}}' + \widehat{\boldsymbol{D}_{\psi}}\right)$

- The residual matrix tells us how far away the correlation matrix **R** we have observed is from the correlation matrix the model predicts. In other words, how far is the observed correlation matrix from the model-implied correlation matrix (which is simply $\widehat{\mathbf{\Lambda}}\widehat{\mathbf{\Lambda}}' + \widehat{\boldsymbol{D}_{\psi}}$)

Working with a sample correlation matrix

- Every element in the residual matrix tells us how far is the model-implied (predicted) value of this element from its observed value.

- Alright, so – again, we don't have a population correlation matrix **P** which we used for all the computations and methods covered before. What are we going to do?

- Of course, we're going to pretend like the problem isn't there and we'll start by doing things in the exact same way.

Working with a sample correlation matrix

- First, we will obtain some initial communality estimates and plug them into the diagonal of **R**. We can use the SMCs.

- This way, we will arrive at our $(\boldsymbol{R} - \boldsymbol{D}_{\psi})$ matrix just as we did arrive previously at the $(\boldsymbol{P} - \boldsymbol{D}_{\psi})$ matrix. [Oh, by the way, did I ever call this a *reduced* correlation matrix? I didn't? Well, now I do.]

- We get the eigenvalues and eigenvectors for this reduced [see?] sample correlation matrix.

Working with a sample correlation matrix

- Again, we will obtain some eigenvalues and some eigenvectors. However, in this case (**not** having a population correlation matrix, **not** being sure the model holds exactly in the population), we will generally not obtain an eigen-solution where the ($p - m$) smallest eigenvalues are zero.

- Thus, we cannot rely on the number of non-zero eigenvalues to show us the "true" number of factors (m). Thus, we will have to choose m ourselves beforehand, based on our best judgement (more on that later)

Working with a sample correlation matrix

- Thus, having chosen the number m beforehand, we will take the m largest eigenvalues and their corresponding m eigenvectors.

- Just like before, we will take the square root of the eigenvalues, sort them by size and place them in a diagonal matrix $\widehat{\boldsymbol{D}}_{lm}^{1/2}$

- And, just like before, we will create a matrix $\widehat{\boldsymbol{U}}_m$ with the corresponding eigenvectors as columns.

Working with a sample correlation matrix

- Then, we can use the eigenvalues and eigenvector matrices to compute our estimate of factor loadings:

$$\widehat{\boldsymbol{\Lambda}} = \widehat{\boldsymbol{U}}_m \widehat{\boldsymbol{D}}_{lm}^{1/2}$$

- The $\widehat{\boldsymbol{\Lambda}}$ obtained in this way minimizes the residual sum of squares (RSS):

$$RSS = \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p}[\left(\boldsymbol{R} - \widehat{\boldsymbol{D}}_{\psi}\right) - \widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}']_{ij}^2$$

- Let me illustrate the formula for RSS now on the board

Working with a sample correlation matrix

- **This $\hat{\Lambda}$ results in minimum sum of squared residuals, conditional on the given set of prior communality estimates.**

- **This method is known as the *principal factor method using prior communality estimates* (whoa, that's a LONG name)**

- **Let's look at an example.**

Example

- Previously, we used the population correlation matrix for the four-tests-data for examples. Assume we have a sample correlation matrix **R** from the same data, different from the population matrix **P:**

$$\boldsymbol{R} = \begin{bmatrix} 1 & & & \\ .51 & 1 & & \\ .11 & .17 & 1 & \\ .41 & .52 & .39 & 1 \end{bmatrix}$$

- We would like to estimate the model, but we do not know the true communalities. Also, we do not know if the model holds in the population.

Example

- We can, however, compute the SMCs as approximations to the true communalities.

- In this case, we would get the following SMCs: .29, .377, .156 and .39 (respectively). We will replace the diagonal elements in **R** with the communality estimates to get our reduced sample correlation matrix:

$$(\boldsymbol{R} - \widehat{\boldsymbol{D}}_{\psi}) = \begin{bmatrix} .29 & & & \\ .51 & .377 & & \\ .11 & .17 & .156 & \\ .41 & .52 & .39 & .39 \end{bmatrix}$$

Example

- Choosing $m = 2$, we get the following m eigenvalues and m eigenvectors:

$$\widehat{\boldsymbol{D}}_l = \begin{bmatrix} 1.438 & \\ & .184 \end{bmatrix}$$

$$\widehat{\boldsymbol{U}} = \begin{bmatrix} .492 & -.462 \\ .571 & -.331 \\ .296 & .754 \\ .586 & .330 \end{bmatrix}$$

Example

- **...and we compute $\widehat{\boldsymbol{\Lambda}}$:**

$$\widehat{\boldsymbol{\Lambda}} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{D}}_l^{1/2} = \begin{bmatrix} .590 & -.198 \\ .685 & -.142 \\ .355 & .323 \\ .703 & .141 \end{bmatrix}$$

- **Interesting point: the sums of squares of each column correspond to the eigenvalues.**

Example

- Then, we can compute the model-implied **reduced** correlation matrix:

$$\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{\Lambda}}' = \begin{bmatrix} .387 & & & \\ .432 & .489 & & \\ .146 & .198 & .231 & \\ .387 & .461 & .295 & .514 \end{bmatrix}$$

- On the diagonal of this matrix are the communalities for this solution. These values are the sums of squares of factor loading in each row of $\hat{\boldsymbol{\Lambda}}$
- Off the diagonal are the correlations between the MVs as reconstructed by this particular model solution.

Example

- **Compare the model-implied reduced correlation matrix to the observed reduced correlation matrix:**

$$\widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}' = \begin{bmatrix} .387 & & & \\ .432 & .489 & & \\ .146 & .198 & .231 & \\ .387 & .461 & .295 & .514 \end{bmatrix}$$

$$(\boldsymbol{R} - \widehat{\boldsymbol{D}}_{\boldsymbol{\psi}}) = \begin{bmatrix} .29 & & & \\ .51 & .377 & & \\ .11 & .17 & .156 & \\ .41 & .52 & .39 & .39 \end{bmatrix}$$

Example

- **They don't match – the model does not fit the data perfectly.**

$$\widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}' = \begin{bmatrix} .387 & & & \\ .432 & .489 & & \\ .146 & .198 & .231 & \\ .387 & .461 & .295 & .514 \end{bmatrix}$$

$$(\boldsymbol{R} - \widehat{\boldsymbol{D}}_{\boldsymbol{\psi}}) = \begin{bmatrix} .29 & & & \\ .51 & .377 & & \\ .11 & .17 & .156 & \\ .41 & .52 & .39 & .39 \end{bmatrix}$$

Example

- **This lack of fit is represented in the residual matrix:**

$$\left(\boldsymbol{R}-\widehat{\boldsymbol{D}}_{\boldsymbol{\psi}}\right)-\widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}'=\begin{bmatrix}-.097 & & & \\ .078 & -.113 & & \\ -.036 & -.028 & -.075 & \\ .023 & .059 & .095 & -.125\end{bmatrix}$$

Example

- The solution produced a residual matrix with minimum sum of squares, conditional on the prior communality estimates. If the prior communality estimates would be different, a different residual matrix would satisfy the RSS criterion.

Short review

- So, what was the principle behind the *principal factor method using prior communality estimates*? Let's do a short recap:

- 1) First, we obtain some communality estimates (like SMCs) and plug them into the diagonal of **R**. Thus, we get our estimate of $\left(\boldsymbol{R} - \boldsymbol{D}_{\psi}\right)$
- 2) Then, we obtain the eigen-solution of $\left(\boldsymbol{R} - \boldsymbol{D}_{\psi}\right)$
- 3) We use the eigen-solution to obtain $\widehat{\boldsymbol{\Lambda}}$
- 4) What we just got is a solution that minimizes the Residual Sum of Squares (RSS) given our initial $\widehat{\boldsymbol{D}}_{\psi}$

Iterative procedure

- Alternatively, we could try to minimize the RSS criterion by estimating $\boldsymbol{D}_{\psi}$ alongside $\boldsymbol{\Lambda}$ and not sticking only with the initial $\widehat{\boldsymbol{D}}_{\psi}$
- This can be done iteratively, by minimizing RSS with respect to both $\widehat{\boldsymbol{D}}_{\psi}$ and $\widehat{\boldsymbol{\Lambda}}$
- This technique is called *iterative principal factors* or *ordinary least squares* (OLS) or *unweighted least squares* (ULS) or *minres*

Iterative procedure

- We will start by doing things the same way we did previously, using the *principal factors method*:

- 1) First, we obtain some communality estimates (like SMCs) and plug them into the diagonal of **R**. Thus, we get our estimate of $\left(\boldsymbol{R}-\boldsymbol{D}_{\psi}\right)$
- 2) Then, we obtain the eigen-solution of $\left(\boldsymbol{R}-\boldsymbol{D}_{\psi}\right)$
- 3) We use the eigen-solution to obtain $\widehat{\boldsymbol{\Lambda}}$

- ...but we won't end here. We will use the computed $\widehat{\boldsymbol{\Lambda}}$ to obtain new communality estimates by summing the squared elements in each row of $\widehat{\boldsymbol{\Lambda}}$ (diagonal elements of $\widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}'$)

Iterative procedure

- We shall take the new communality estimates and plug them into the diagonal of **R**. Thus, we get a new $\left(\boldsymbol{R}-\boldsymbol{D}_{\psi}\right)$
- Again, we obtain the eigen-solution of this new $\left(\boldsymbol{R}-\boldsymbol{D}_{\psi}\right)$ and use it to compute a new $\widehat{\boldsymbol{\Lambda}}$
- …and repeat (use the newly computed $\widehat{\boldsymbol{\Lambda}}$ to again obtain new communality estimates). We continue this process until the communalities obtained in successive iterations do not significantly differ by some pre-set criterion (convergence criterion).

Iterative procedure

- That’s really all there is (in principle) about OLS.

- By the way, the RSS function (the formula we have seen before) is a *discrepancy function* – it quantifies the distance between the observed and model-implied correlation matrices. In other words, it expresses the degree of lack of model fit.

- Being a discrepancy function, it is always greater than or equal to zero and is zero **only** when the observed and model-implied correlation matrices are the same.

Heywood cases

- One nasty thing can happen when using OLS estimation
- That is, some communalities can, in the course of the iterations, be greater than one. Conversely, the unique variances can become less than zero (because in a standardized solution, the communality and the unique variance of an MV add up to one)
- But there's no such thing as negative variance. Thus, such a solution would be nonsensical and unacceptable. We call these occurrences *Heywood cases*

Heywood cases

- If you're using smart software, you should be notified whenever a *Heywood case* occurs

- If you're using smart software, it can help you circumvent the problem by placing a constraint on the associated unique variance such that it can only be greater than or equal to zero.

Summary

- **We considered multiple scenarios of fitting the model to data. Let's do a quick review.**

- **1) You know $\mathbf{P}$ and you know $\boldsymbol{D_\psi}$. You can obtain the eigen-solution of ($\mathbf{P}$ - $\boldsymbol{D_\psi}$) to compute $\boldsymbol{\Lambda}$.**

….however, this will never be the case in practice.

Summary

- We considered multiple scenarios of fitting the model to data. Let's do a quick review.

- 2) You know **P** but you do not know $\boldsymbol{D}_{\psi}$. You can estimate communalities using SMCs and plug them into the diagonal of **P** to obtain ($\mathbf{P} - \boldsymbol{D}_{\psi}$). Afterwards, you obtain the eigen-solution of ($\mathbf{P} - \boldsymbol{D}_{\psi}$) to obtain $\boldsymbol{\Lambda}$.

....however, this will **also** never be the case in practice.

Summary

- We considered multiple scenarios of fitting the model to data. Let's do a quick review.

- 3) You do not know **P** and you do not know $\boldsymbol{D_\psi}$. All you have is **R**. You can estimate communalities using SMCs and plug them into the diagonal of **R** to obtain $(\mathbf{R} - \widehat{\boldsymbol{D}}_\psi)$. Obtain the eigen-solution of $(\mathbf{R} - \widehat{\boldsymbol{D}}_\psi)$ to get $\widehat{\boldsymbol{\Lambda}}$.

....the solution minimizes RSS given your original $\widehat{\boldsymbol{D}}_\psi$. This can happen very often in practice, although we would normally use a better option coming up next.

Summary

- We considered multiple scenarios of fitting the model to data. Let's do a quick review.

- 4) You do not know **P** and you do not know $\boldsymbol{D_\psi}$. All you have is **R**. You can estimate communalities using SMCs and plug them into the diagonal of **R** to obtain (**R** - $\widehat{\boldsymbol{D}}_{\boldsymbol{\psi}}$). Obtain the eigen-solution of (**R** - $\widehat{\boldsymbol{D}}_{\boldsymbol{\psi}}$) to get $\widehat{\boldsymbol{\Lambda}}$. Use the computed $\widehat{\boldsymbol{\Lambda}}$ to obtain new communality estimates from the diagonal of $\widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}'$. Return to the beginning with fresh new communality estimates, repeat until convergence.