In some respects, one pulls oneself up by one's bootstraps in applying this method. It reminds the author of the joke about the thermodynamic engineer, the physicist, and the statistician who were marooned on a desert island. As food became scarce, one day they observed a wave wash a can of beans ashore. They rushed and took it from the beach ready to eat it. But how were they to open the can to get the beans? The thermodynamic engineer pondered a moment and then he said, "I've got it! Let's build a fire and heat a pile of stones quite hot, cover these with palm leaves, place the can of leaves on top of them and cover it also with palm leaves. The heat of the stones will cause the water in the can to boil and the can will burst, freeing the beans." The physicist did not like that idea because the beans might be spread around from the explosion of the can. "I have a better idea," he said. See that tall palm tree down the beach. It leans over a pile of sharp rocks. I figure it is high enough that if we climb the tree and drop the can onto the rocks, the can will be broken open and we can get the beans." The statistician looked incredulously at the other two. "I don't know why you are going to such elaborate and dangerous steps when the answer is quite simple: first, we assume we have a can opener…."

Vectors of Mind III

Assume you have a subtitle

Contents

- Random variables
- Fundamental Theorem of FA
- Rotational indeterminancy
- Estimation
  - Principal factors
  - Iterative principal factors (i.e., OLS / ULS / minres)
  - Maximum likelihood

Contents

- Random variables
- **FUN**damental Theorem of FA
- Rotational indeterminancy
- Estimation
  - Principal factors
  - Iterative principal factors (i.e., OLS / ULS / minres)
  - Maximum likelihood

- We‘re gonna see many easy, approachable, and friendly-looking equations

$$\boldsymbol{\Sigma = \Lambda\Lambda + \Psi^2 = \Lambda QQ'IQQ'\Lambda' + \Psi^2}$$

- And lots of simplifications that make things even clearer

Consider that

$$\mathbf{R}^2 = \mathbf{RR}$$

which we may rewrite as

$$\mathbf{R}^2 = \mathbf{ADA'ADA'} = \mathbf{ADDA'} = \mathbf{AD}^2\mathbf{A'}$$


GALILEO – GALILEO, GALILEO
ADA'ADA' = ADDA' = AD^2A'

Random variables

Random variables

- A variable whose values are samples from a probability distribution
- It doesn‘t have one true value, the probability distribution ***is*** the variable as it represents the random process behind the variable

- Example:

$$\boldsymbol{X} \sim N(\mu, \sigma^2)$$

X is normally distributed with a mean μ and variance σ^2

Expected value

- A long-run average of a random variable

$$\boldsymbol{X} \sim N(\mu, \sigma^2)$$
$$E(\boldsymbol{X}) = \mu$$

Expected values – sidenote

- By the way!
- You know what is defined as an expected value?

- The CTT true score!

$$X = \tau + \varepsilon$$
$$\tau = E(X)$$

In other words, true score is defined as the long-run average of the raw score

Expected value – constants

- Expectation of a constant is the constant itself (because it‘s not a random variable)

$$E(\boldsymbol{C}) = \boldsymbol{C}$$

Expected values – variance

- Now, consider the (scalar) formula for the **variance** of a random variable:

$$\sigma^2 = \frac{\sum(x-\mu)^2}{N}$$

…which is the “mean squared deviation from the mean”, right?

- As an expected value: $E[(X - \mu)^2]$

Expected values – variance/covariance matrix

$$C_x = E[(\boldsymbol{x} - \boldsymbol{\mu})\,(\boldsymbol{x} - \boldsymbol{\mu})']$$

- Expanding, we get the expectation of:

$$\begin{bmatrix} (x_1 - \mu_1) \\ (x_2 - \mu_2) \\ \vdots \\ (x_p - \mu_p) \end{bmatrix} [(x_1 - \mu_1) \quad (x_2 - \mu_2) \quad \cdots \quad (x_p - \mu_p)]$$

- …which gives us the variance/covariance matrix of the manifest variables

Expected values – centered variables

- For centered variables (means = 0), we can omit the $\boldsymbol{\mu}$s:

$$C_x = E[(\boldsymbol{x}\,\boldsymbol{x}')]$$

Expected values – summary

For normally distributed random variables:

$$\boldsymbol{X} \sim N(\mu, \sigma^2)$$
$$E(\boldsymbol{X}) = \mu$$

For non-random variables:

$$E(\boldsymbol{C}) = \boldsymbol{C}$$

To get a variance-covariance matrix:

$$C_x = E[(\boldsymbol{x} - \boldsymbol{\mu})\,(\boldsymbol{x} - \boldsymbol{\mu})']$$

For centered variables:

$$C_x = E[\boldsymbol{x}\,\boldsymbol{x}']$$

Deriving the fundamental theorem of FA

The data model in factor analysis

- Recall the way we formulated the Common Factor Model earlier – we expressed the MVs as a linear function of the **common factors** and the **unique factors**:

$$x_{ij} = \mu_j + \lambda_{j1}z_{i1} + \lambda_{j2}z_{i2} + \cdots + \lambda_{jm}z_{im} + 1u_{ij}$$

Mean + Common factor part + Unique factor part

$$x_{ij} = \mu_j + \sum_{k=1}^{m} \lambda_{jk}z_{ik} + u_{ij}$$

The data model in factor analysis

$$x_{ij} = \mu_j + \sum_{k=1}^{m} \lambda_{jk} z_{ik} + u_{ij}$$

Where:

x_{ij} is the score of person *i* on manifest variable *j*

μ_j is the mean of manifest variable *j*

z_{ik} is the common factor score of person *i* on factor *k*

λ_{jk} is the factor loading of manifest variable *j* on factor *k*

u_{ij} is the unique factor score of person *i* on unique factor *j;* and $u_{ij} = s_{ij} + e_{ij}$

s_{ij} is the factor score of person *i* on specific factor *j*

e_{ij} is the error term for person *i* on manifest variable *j*

The data model in factor analysis

- We will consider the model as operating in a population, and thus we will consider the data model for a random individual by omitting the subscript *i*:

$$x_j = \mu_j + \lambda_{j1}z_1 + \lambda_{j2}z_2 + \cdots + \lambda_{jm}z_m + 1u_j$$

$$x_j = \mu_j + \sum_{k=1}^{m} \lambda_{jk}z_k + u_j$$

- Here we actually have *p* equations, one for each manifest variable $x_j, \ldots, x_p$, but we can express it all as a single equation using matrix notation:

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

The data model in factor analysis

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

Where:

- $\boldsymbol{x}$ is a p x 1 vector of a random person's scores on the p manifest variables
- $\boldsymbol{\mu}$ is a p x 1 vector of population means of the p manifest variables
- $\boldsymbol{\Lambda}$ is a p x m matrix of factor loadings, where $p > m$ (rectangular matrix)
- $\boldsymbol{z}$ is a m x 1 vector of (unobservable) common factor scores
- $\boldsymbol{u}$ is a p x 1 vector of (unobservable) unique factor scores

The data model in factor analysis

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

- For illustration, let's extract the equation for the third manifest variable. Let's assume that *m* = 3 (there are three common factors):

$$\begin{bmatrix} \\ x_3 \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \mu_3 \\ \\ \end{bmatrix} + \begin{bmatrix} & & \\ \lambda_{31} & \lambda_{32} & \lambda_{33} \\ & & \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} \\ u_3 \\ \\ \end{bmatrix}$$

$$x_3 = \mu_3 + \lambda_{31} z_1 + \lambda_{32} z_2 + \lambda_{33} z_3 + u_3$$

The data model in factor analysis

- The data model represents a random observation in the population. It is intended to explain the structure of the raw data (i.e., the scores on manifest variables)

- However, it contains a LOT of unknowns
- While we observe the manifest variables ***x*** and we can at least estimate the population means $\boldsymbol{\mu}$, the remaining terms in the equation are unknown to us
- We do not know the latent scores ***z*** and ***u***, in fact we **cannot know** them, since latent variables are **unobservable**
- Similarly, we do not know **Λ**, the matrix of factor loadings – we are unaware of how the unobservable latent variables affect the (observable) manifest variables

The data model in factor analysis

- Well, that's kind of a pickle.
- So, do we just, like, go home now?

- Maybe. Or we can help ourselves with some tricks.
- We have already established that the latent variable scores are unobservable, so we might want to give up on trying to solve for them in the data model equation

- Maybe if we turn the problem around, we can get rid of $\boldsymbol{z}$ and $\boldsymbol{u}$ completely and focus on $\boldsymbol{\Lambda}$

The data model in factor analysis

- We could use the data model, along with some assumptions, to derive a **covariance structure** model

- The data model is accompanied by assumptions about the joint distribution of the elements in ***z*** and ***u*** implies a model for the population covariance matrix.
- The model for the covariance matrix is known as the **covariance structure** and is intended to explain the variances and covariances of the manifest variables, **not** the raw data.
- Before we proceed to derive the covariance structure model, we'll talk about the important distributional assumptions and lay down some notational rules.

Assumptions

- We will make the following assumptions about the common factors *z* and unique factors *u*:
  1. The common factors and the unique factors are independently distributed. As such, the common factors are **uncorrelated** with the unique factors. In other words, $\boldsymbol{\Sigma}_{zu} = \mathbf{0} = \boldsymbol{\Sigma}'_{uz}$
  2. The unique factors are mutually independent. As such, the unique factors for different MVs are **uncorrelated** with each other. This implies that the covariance matrix $\boldsymbol{\Sigma}_{uu}$ is diagonal.
  3. The common factors and the unique factors are standardized to have means of zero.
  4. The common factors are also standardized to have unit variances (variances of 1).

Assumptions – recap

Common factors:

$$Since\colon\ E(\boldsymbol{z}) = 0$$

$$Then\colon\ E(\boldsymbol{z}\boldsymbol{z}') = \boldsymbol{\Sigma}_{zz}$$

$$Also\colon\ E(\boldsymbol{z}\boldsymbol{z}') = \boldsymbol{\Sigma}_{zz} = \boldsymbol{R}_{zz}$$

Assumptions – recap

Unique factors:

$$Since\colon\ E(\boldsymbol{u}) = 0$$

$$Then\colon\ E(\boldsymbol{u}\boldsymbol{u}') = \boldsymbol{\Sigma}_{zz}$$

$$Also\colon\ E(\boldsymbol{u}\boldsymbol{u}') = \boldsymbol{\Sigma}_{uu} = \boldsymbol{R}_{uu} = \boldsymbol{I}$$

Assumptions – recap

Relationship between unique and common factors:

$$\boldsymbol{R}_{zu} = \mathbf{0} = \boldsymbol{R}'_{uz}$$

Deriving the mean structure

- The mean and covariance structures are derived from the data model:

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

- Let's derive the mean structure first. We want an equation that represents the mean vector $\boldsymbol{\mu}$ of the manifest variables.

Deriving the mean structure

- If we take the expectation of both sides of the equation above, we get:

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

- Given the assumptions we previously talked about, this follows:

$$\boldsymbol{\mu}_{\mathrm{x}} = \boldsymbol{\mu} + \boldsymbol{\Lambda 0} + \boldsymbol{0}$$
$$\boldsymbol{\mu}_{\mathrm{x}} = \boldsymbol{\mu}$$

Therefore, means do not depend on the model!

Deriving the covariance structure

- Alright. Let's consider the derivation of the covariance structure.
- We know that it scores on the MVs are supposed to be weighted linear combinations of common factors and unique factors:

$$\boldsymbol{X} = \boldsymbol{\Lambda Z} + \boldsymbol{\Psi U}$$

- Right? Regression style.

Deriving the covariance structure

- First, let's standardize the vector of MVs to make our lives easier.
- This way, $\boldsymbol{\Sigma}_{xx}$ becomes $\boldsymbol{R}_{xx}$

$$\boldsymbol{X} = \boldsymbol{\Lambda Z} + \boldsymbol{\Psi U}$$

- We also know that in this case, $\boldsymbol{R}_{xx} = \mathrm{E}(\mathbf{XX}')$
- So:

$$\boldsymbol{R}_{xx} = E[\,(\boldsymbol{\Lambda Z} + \boldsymbol{\Psi U})\,(\boldsymbol{\Lambda Z} + \boldsymbol{\Psi U})'\,]$$

Deriving the covariance structure

$$\boldsymbol{R}_{xx} = E[\,(\boldsymbol{\Lambda Z} + \boldsymbol{\Psi U})\,(\boldsymbol{\Lambda Z} + \boldsymbol{\Psi U})'\,]$$

- Now, let's transpose the second bracket (it simply transposes all the elements while switching the positions of products):

$$\boldsymbol{R}_{xx} = E[\,(\boldsymbol{\Lambda Z} + \boldsymbol{\Psi U})\,(\boldsymbol{Z'\Lambda'} + \boldsymbol{U'\Psi'})\,]$$

- Now, let's multiply the contents of the expectation:

$$\boldsymbol{R}_{xx} = E[\,\boldsymbol{\Lambda Z Z'\Lambda'} + \boldsymbol{\Lambda Z U'\Psi'} + \boldsymbol{\Psi U Z'\Lambda'} + \boldsymbol{\Psi U U'\Psi'}]$$

Deriving the covariance structure

$$\boldsymbol{R}_{xx} = E[\,\boldsymbol{\Lambda ZZ'\Lambda'} + \boldsymbol{\Lambda ZU'\Psi'} + \boldsymbol{\Psi UZ'\Lambda'} + \boldsymbol{\Psi UU'\Psi'}]$$

- Now, we just get rid of the expectation (just like I got rid of expecting to finish my Ph.D. on time):

$$\boldsymbol{R}_{xx} = E(\boldsymbol{\Lambda})E(\boldsymbol{ZZ'})E(\boldsymbol{\Lambda'}) + E(\boldsymbol{\Lambda})E(\boldsymbol{ZU'})E(\boldsymbol{\Psi'}) + E(\boldsymbol{\Psi})E(\boldsymbol{UZ'})E(\boldsymbol{\Lambda'}) + E(\boldsymbol{\Psi})E(\boldsymbol{UU'})E(\boldsymbol{\Psi'})$$

- All the factor weights (loadings) $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ are constants:

$$\boldsymbol{R}_{xx} = \boldsymbol{\Lambda}E(\boldsymbol{ZZ'})\boldsymbol{\Lambda'} + \boldsymbol{\Lambda}\mathrm{E}(\mathbf{ZU'})\boldsymbol{\Psi'} + \boldsymbol{\Psi}E(\boldsymbol{UZ'})\boldsymbol{\Lambda'} + \boldsymbol{\Psi}E(\boldsymbol{UU'})\boldsymbol{\Psi'}$$

Deriving the covariance structure

- $E(\boldsymbol{ZZ}')$ is the variance-covariance matrix of the common factors $\boldsymbol{\Sigma}_{zz}$
- $E(\boldsymbol{UU}')$ is the variance-covariance matrix of the unique factors $\boldsymbol{\Sigma}_{uu}$
- $E(\boldsymbol{ZU}')$ and $E(\boldsymbol{UZ}')$ are the variance-covariance matrices of the common and unique factors among themselves $\boldsymbol{\Sigma}_{uz}$. So, this:

$$\boldsymbol{R}_{xx} = \boldsymbol{\Lambda} E(\boldsymbol{ZZ}')\boldsymbol{\Lambda}' + \boldsymbol{\Lambda}\mathrm{E}(\mathbf{ZU}')\boldsymbol{\Psi}' + \boldsymbol{\Psi} E(\boldsymbol{UZ}')\boldsymbol{\Lambda}' + \boldsymbol{\Psi} E(\boldsymbol{UU}')\boldsymbol{\Psi}'$$

- Becomes this

$$\boldsymbol{R}_{xx} = \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{zz}\boldsymbol{\Lambda}' + \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{uz}\boldsymbol{\Psi}' + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{uz}\boldsymbol{\Lambda}' + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{uu}\boldsymbol{\Psi}'$$

Deriving the covariance structure

$$\boldsymbol{R}_{xx} = \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{zz}\boldsymbol{\Lambda}' + \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{uz}\boldsymbol{\Psi}' + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{uz}\boldsymbol{\Lambda}' + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{uu}\boldsymbol{\Psi}'$$

- Phew! Now what?

Deriving the covariance structure

$$\boldsymbol{R}_{xx} = \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{zz}\boldsymbol{\Lambda}' + \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{uz}\boldsymbol{\Psi}' + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{uz}\boldsymbol{\Lambda}' + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{uu}\boldsymbol{\Psi}'$$

- Phew, now what?
- We get help from our favourite superhero – Mr. **Assumptionman** – of course!

Deriving the covariance structure

$$\boldsymbol{R}_{xx} = \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{zz}\boldsymbol{\Lambda}' + \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{uz}\boldsymbol{\Psi}' + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{uz}\boldsymbol{\Lambda}' + \boldsymbol{\Psi}\boldsymbol{\Sigma}_{uu}\boldsymbol{\Psi}'$$

- Phew, now what?
- We get help from our favourite superhero – Mr. **Assumptionman** – of course! (He's bold because he's a matrix. Wait what, that doesn't make sense)

Deriving the covariance structure

$$\boldsymbol{R_{xx} = \Lambda \Sigma_{zz} \Lambda' + \Lambda \Sigma_{uz} \Psi' + \Psi \Sigma_{uz} \Lambda' + \Psi \Sigma_{uu} \Psi'}$$

- We know some stuff about those sigmas:
  - $\boldsymbol{\Sigma_{uz} = 0 = \Sigma_{uz}}$
  - $\boldsymbol{\Sigma_{uu} = I}$

$$\boldsymbol{R_{xx} = \Lambda \Sigma_{zz} \Lambda' + \Lambda 0 \Psi' + \Psi 0 \Lambda' + \Psi I \Psi'}$$

$$\boldsymbol{R_{xx} = \Lambda \Sigma_{zz} \Lambda' + \Psi^2}$$

Notation

- We will use the following notation from now on:

  The manifest variable covariance matrix: $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{xx}$
  The common factor covariance matrix: $\boldsymbol{\Phi} = \boldsymbol{\Sigma}_{zz}$
  The unique factor covariance matrix: $\boldsymbol{D}_{\psi} = \boldsymbol{\Sigma}_{uu}$

- Note that (because of the assumptions we made), the diagonal elements of $\boldsymbol{\Phi}$ are required to be equal to 1. Thus, $\boldsymbol{\Phi}$ is a factor correlation matrix.

Fundamental Theorem of FA

$$\boldsymbol{R_{xx} = \Lambda \phi \Lambda' + \Psi^2}$$

- For $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma = \Lambda \phi \Lambda' + D_{\psi}}$$

$\boldsymbol{\Psi^2}$ is a diagonal matrix with uniquenesses on the diagonal

$\boldsymbol{D_{\psi}}$ is a diagonal matrix with unique variances on the diagonal (to scale the resulting correlation matrix into a variance-covariance matrix)

Implications of this

1. We don‘t need to know the latent scores (common or unique) to reproduce the variance-covariance matrix of MVs! Whaaaat!

Implications of this

2. The model assumes all correlations between MVs are only due the effect of the common factors:

$$E\left(\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}'\right) = \widehat{\boldsymbol{\Sigma}} = \boldsymbol{\Lambda}\boldsymbol{\phi}\boldsymbol{\Lambda}'$$

Or $E\left(\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}'\right) = \widehat{\boldsymbol{\Sigma}} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}'$ for uncorrelated factors

- How well this assumption corresponds with the reality is what model fit testing is all about. It‘s sometimes called the local independence assumption.

Implications of this

2. The model assumes all correlations between MVs are only due the effect of the common factors:

$$E\left(\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}'\right) = \widehat{\boldsymbol{\Sigma}} = \boldsymbol{\Lambda}\boldsymbol{\phi}\boldsymbol{\Lambda}'$$

This also implies that λ is a regression coefficient. It tells us how the value of x changes with a unit change in the factor score. Which is what we started with so it‘s probably not a surprise.

Implications of this

3. We can use the information from $\boldsymbol{\Psi^2}$ to create the so-called *reduced correlation matrix* that has communalities (1-uniqueness) on the diagonal and MV correlations off the diagonal.

$$\boldsymbol{R_{xx} = \Lambda\phi\Lambda' + \Psi^2}$$
$$\boldsymbol{R_{xx} - \Psi^2 = \Lambda\phi\Lambda'}$$

- This will be useful later on

Implications of this

4. $\boldsymbol{\Lambda}$ has different elements for $\boldsymbol{R}_{xx}$ and $\boldsymbol{\Sigma}$.

- As we already covered, $\boldsymbol{\Psi^2}$ contains uniquenesses while $\boldsymbol{D}_{\psi}$ contains (unscaled) unique variances
- Similarly, we distinguish $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^*$, the second containing standardized factor loadings λ^*

$$\lambda_i^* = \frac{\lambda_i}{SD_i}$$

Or in matrix-speak:

$$\boldsymbol{\Lambda}^* = \mathbf{D}_{\boldsymbol{\sigma}}^{-1/2}\boldsymbol{\Lambda}$$

where $\mathbf{D}_{\boldsymbol{\sigma}}^{-1/2}$ is a diagonal matrix with reciprocal SDs (1/SD) of MVs on the diagonal

An example

- Consider the covariance structure for uncorrelated (orthogonal) factors to better understand the relationship between elements of $\boldsymbol{\Sigma}$ and the elements of $\boldsymbol{\Lambda}$ and$\boldsymbol{D}_{\psi}$. An example:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & & & \\ \sigma_{21} & \sigma_{22} & & \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \\ \lambda_{41} & \lambda_{42} \end{bmatrix} \begin{bmatrix} \lambda_{11} & \lambda_{21} & \lambda_{31} & \lambda_{41} \\ \lambda_{12} & \lambda_{22} & \lambda_{32} & \lambda_{42} \end{bmatrix} + \begin{bmatrix} \psi_{11} & & & \\ & \psi_{22} & & \\ & & \psi_{33} & \\ & & & \psi_{44} \end{bmatrix}$$

- This shows us that $\sigma_{11} = \lambda_{11}^2 + \lambda_{12}^2 + \psi_{11}$
- Also, $\sigma_{21} = \lambda_{21}\lambda_{11} + \lambda_{22}\lambda_{12}$
- The covariance between two MVs is the sum of the products of their loadings on the common factors
- The variance of an MV is the sum of its squared loadings and its unique factor variance

An example

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

An example

- The factor loading matrix is:

	Factor 1	Factor 2
PC	.70	.10
VO	.70	.00
AR	.10	.70
MPS	.60	.60

- The covariance between PC and VO:

$$\sigma_{21} = \lambda_{21}\lambda_{11} + \lambda_{22}\lambda_{12} = 0.7 * 0.7 + 0.0 * 0.1 = 0.49$$

- Let's compute the communality and the unique variance of PC by hand

Communality

- The *j*-th diagonal element ψ_{jj} of $\boldsymbol{D}_{\psi}$ is the *j*-th unique variance. The *j*-th **communality** (proportion of variance of MV *j* due to common factors) can be written as:

$$h_{jj} = \frac{[\boldsymbol{\Lambda\Phi\Lambda}']_{jj}}{\sigma_{jj}} = 1 - \frac{\psi_{jj}}{\sigma_{jj}}$$

- If the factors are uncorrelated, then:

$$h_{jj} = \frac{[\boldsymbol{\Lambda\Lambda}']_{jj}}{\sigma_{jj}} = 1 - \frac{\psi_{jj}}{\sigma_{jj}}$$

...that is, the sum of squares of row *j* of $\boldsymbol{\Lambda}$ divided by the variance of the *j*-th MV.

Recap

- We wanted to predict MVs as a weighted combination of common and unique factor scores:

$$\boldsymbol{X = \Lambda Z + \Psi U}$$

- But we don‘t know the scores, so, instead, we look at their covariance structure
- That‘s how we got here:

$$\boldsymbol{R_{xx} = \Lambda \phi \Lambda' + \Psi^2}$$

Recap

$$\boldsymbol{R_{xx} = \Lambda \phi \Lambda' + \Psi^2}$$

- $\boldsymbol{\Lambda}$ = matrix of factor loadings (k x p)
- $\boldsymbol{\phi}$ = matrix of factor correlations (p x p)
- $\boldsymbol{\Psi^2}$ = diagonal matrix of uniquenesses (k x k)

- You can also scale this into a covariance equation as shown before.

Estimation

Estimation of FA parameters

- Now we need to think about how to find the values to place into the model matrices $\boldsymbol{\Lambda}$,$\boldsymbol{\phi}$, $\boldsymbol{\Psi^2}$

- Let‘s start with an ideal scenario where the factors are uncorrelated (so $\boldsymbol{\phi} = \boldsymbol{I}$) and our observed correlation matrix is the population correlation matrix $\mathbf{P}$ (i.e., there is no sampling error involved)

Rotational indeterminancy

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

- But wait! Even in this scenario, things are weird:

$$\mathbf{P} = \boldsymbol{\Lambda_1\Lambda_1'} + \mathbf{D}_\psi = \boldsymbol{\Lambda_2\Lambda_2'} + \mathbf{D}_\psi = \boldsymbol{\Lambda_3\Lambda_3'} + \mathbf{D}_\psi = \cdots$$

What the hell! Are you telling me there are infinite possible $\boldsymbol{\Lambda}$s?

Rotational indeterminancy

- Hell yeah! If you have more than two factors, there is no unique solution to be found.

- 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 ($\boldsymbol{\Lambda_1}$). To be precise, $\boldsymbol{\Lambda_2} = \boldsymbol{\Lambda_1}\mathbf{T}$ where **T** is a *m* x *m* orthogonal matrix ($\mathbf{T}\mathbf{T}' = \mathbf{I}$)

Rotational indeterminancy

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

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

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

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

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

$$\boldsymbol{\Lambda_2 \Lambda_2'} = \boldsymbol{\Lambda_1 \Lambda_1'}$$

- See? $\boldsymbol{\Lambda_1}$ and $\boldsymbol{\Lambda_2}$ are equally fine solutions.

Rotational indeterminancy

- 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 $\boldsymbol{\Lambda_2} = \boldsymbol{\Lambda_1 T}$
- We've just seen that $\boldsymbol{\Lambda_1}$ and $\boldsymbol{\Lambda_2}$ are equally good solutions, since $\boldsymbol{\Lambda_2 \Lambda_2'} = \boldsymbol{\Lambda_1 \Lambda_1}'$

- This is called **rotational indeterminacy**.

Rotational indeterminancy

- 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 that 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).

I: We know $\mathbf{P}, \mathbf{\Psi^2}$ and the model holds perfectly

- Now, let's take a look at **Hypothetical Scenario 1**:
  - You know the true $\mathbf{P}$
  - You know the unique factor variances ($\mathbf{\Psi^2}$), and thus you also know the communalities (diagonal of $\mathbf{P - \Psi^2}$)
  - The model holds perfectly in the population data

I: We know $\mathbf{P}, \mathbf{\Psi^2}$ and the model holds perfectly

- In this scenario, obtaining $\mathbf{\Lambda}$ is actually quite easy
- You take the *reduced correlation matrix* $(\mathbf{P} - \mathbf{\Psi^2})$
- Because $\mathbf{\Psi^2}$ is a diagonal matrix containing uniquenesses, the diagonal of $(\mathbf{P} - \mathbf{\Psi^2})$ will contain (1 – uniquenesses), thus, it will contain the true communalities

I: We know $\mathbf{P}, \mathbf{\Psi^2}$ and the model holds perfectly

- Perform the eigenvalue-eigenvector decomposition of $(\mathbf{P} - \mathbf{\Psi^2})$, which will yield some eigenvectors $\mathbf{U}$ and some eigenvalues $\mathbf{D}_l$

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

- Note: This is the same as PCA, only done on $\mathbf{P} - \mathbf{\Psi^2}$ instead of $\mathbf{P}$

I: We know $\mathbf{P}, \mathbf{\Psi^2}$ and the model holds perfectly

- 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 $\mathbf{\Lambda} = \mathbf{U}\mathbf{D}_l^{1/2}$
- Magic.

I: We know $\mathbf{P}, \mathbf{\Psi^2}$ and the model holds perfectly

- 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

I: We know $\mathbf{P}, \mathbf{\Psi^2}$ and the model holds perfectly

- Assume the unique variances are known:

$$\mathbf{\Psi^2} = \begin{bmatrix} 0.50 & & & \\ & .51 & & \\ & & .50 & \\ & & & .28 \end{bmatrix}$$

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

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

I: We know $\mathbf{P}, \boldsymbol{\Psi}^{\mathbf{2}}$ and the model holds perfectly

- We can obtain the eigenvalues and eigenvectors of $\left(\boldsymbol{P} - \boldsymbol{\Psi}^{\mathbf{2}}\right)$
- The non-zero eigenvalues are:

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

- And the corresponding eigenvectors:

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

I: We know $\mathbf{P}, \mathbf{\Psi^2}$ and the model holds perfectly

- The factor loading matrix can be obtained: $\mathbf{\Lambda = UD_l^{1/2}}$

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

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

I: We know $\mathbf{P}, \mathbf{\Psi^2}$ and the model holds perfectly

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

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

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

I: We know $\mathbf{P}$, $\mathbf{\Psi^2}$ and the model holds perfectly

- Okay, so, I have just shown you how to obtain the solution ($\mathbf{\Lambda}$) if:
  - You know the population correlation matrix, $\mathbf{P}$
  - You know the contents of $\mathbf{\Psi^2}$, 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 $\mathbf{P}$ and you will never know $\mathbf{\Psi^2}$, so this is a theoretical scenario.

II: We know $\mathbf{P}$ and the model holds perfectly. We don‘t know $\mathbf{\Psi^2}$

- 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, 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”

II: We know $\mathbf{P}$ and the model holds perfectly. We don‘t know $\mathbf{\Psi^2}$

- 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.

II: We know $\mathbf{P}$ and the model holds perfectly. We don‘t know $\mathbf{\Psi^2}$

- 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.

II: We know $\mathbf{P}$ and the model holds perfectly. We don‘t know $\boldsymbol{\Psi}^{\mathbf{2}}$

- 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 $\boldsymbol{\Lambda}$.

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- 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.

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- 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 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.

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- First of all, we should tone down the optimism. In our hypothetical scenarios, we could select $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi^2}$ to reconstruct **P** perfectly:

$$\mathbf{P} = \boldsymbol{\Lambda\Lambda'} + \boldsymbol{\Psi^2}$$

- In reality, our *estimates* of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi^2}$, $\widehat{\boldsymbol{\Lambda}}$ and $\widehat{\boldsymbol{\Psi^2}}$, will generally not be able to exactly reproduce our sample correlation matrix **R**:

$$\mathbf{R} \approx \widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}' + \widehat{\boldsymbol{\Psi^2}}$$

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- 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 $\mathbf{R} - \left(\widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}' + \widehat{\boldsymbol{\Psi}^2}\right)$

- 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 $\widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}' + \widehat{\boldsymbol{\Psi}^2}$)

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- 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.

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- 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

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- 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.

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- 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):

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

III: The model does not hold perfectly. We don‘t know $\boldsymbol{\Psi^2}$ or $\mathbf{P}$

- This $\widehat{\boldsymbol{\Lambda}}$ 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*

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 $\left(\boldsymbol{R}-\boldsymbol{\Psi^2}\right)$

2) Then, we obtain the eigen-solution of $\left(\boldsymbol{R}-\boldsymbol{\Psi^2}\right)$

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 $\widehat{\boldsymbol{\Psi^2}}$

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 $(\boldsymbol{R} - \boldsymbol{\Psi^2})$
- 2) Then, we obtain the eigen-solution of $(\boldsymbol{R} - \boldsymbol{\Psi^2})$
- 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 $\left(\boldsymbol{R}-\boldsymbol{\Psi}^{\boldsymbol{2}}\right)$
- Again, we obtain the eigen-solution of this new $\left(\boldsymbol{R}-\boldsymbol{\Psi}^{\boldsymbol{2}}\right)$ 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 model misfit.

- 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*

Summary

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

- 1) You know **P** and you know $\boldsymbol{\Psi^2}$. You can obtain the eigen-solution of (**P** -$\boldsymbol{\Psi^2}$) 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 $\boldsymbol{\Psi^2}$. You can estimate communalities using SMCs and plug them into the diagonal of **P** to obtain (**P** - $\boldsymbol{\Psi^2}$). Afterwards, you obtain the eigen-solution of (**P** - $\boldsymbol{\Psi^2}$) 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 $\boldsymbol{\Psi^2}$. All you have is **R**. You can estimate communalities using SMCs and plug them into the diagonal of **R** to obtain (**R** - $\widehat{\boldsymbol{\Psi^2}}$). Obtain the eigen-solution of (**R** -$\widehat{\boldsymbol{\Psi^2}}$) to get $\widehat{\boldsymbol{\Lambda}}$.

….the solution minimizes RSS given your original $\widehat{\boldsymbol{\Psi^2}}$. 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 $\boldsymbol{\Psi^2}$. All you have is **R**. You can estimate communalities using SMCs and plug them into the diagonal of **R** to obtain (**R** -$\widehat{\boldsymbol{\Psi^2}}$). Obtain the eigen-solution of (**R** -$\widehat{\boldsymbol{\Psi^2}}$) 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.

Intermezzo

- Phew! We‘ve covered a lot of ground, right?
- And that was still just the foundation of the unrestricted FA
- Now, let‘s look at stuff directly applicable to the restricted FA that is the main variation of FA one should learn / use

?

Pokémon

Hints

1/ He was 23 when the photo was taken

Hints

1/ He was 23 when the photo was taken

2/ Year before that, he created the most potent estimation tool in statistics

Hints

1/ He was 23 when the photo was taken

2/ Year before that, he created the most potent estimation tool in statistics

3/ He was a professor of eugenics

Hints

1/ He was 23 when the photo was taken

2/ Year before that, he created the most potent estimation tool in statistics

3/ He was a professor of eugenics

4/ Opposed Bayesian stats

Hints

1/ He was 23 when the photo was taken

2/ Year before that, he created the most potent estimation tool in statistics

3/ He was a professor of eugenics

4/ Opposed Bayesian stats

5/ Coined the „sexy son hypothesis“

Hints

1/ He was 23 when the photo was taken

2/ Year before that, he created the most potent estimation tool in statistics

3/ He was a professor of eugenics

4/ Opposed Bayesian stats

5/ Coined the „sexy son hypothesis“

6/ Popularized *t*-distribution

Hints

1/ He was 23 when the photo was taken

2/ Year before that, he created the most potent estimation tool in statistics

3/ He was a professor of eugenics

4/ Opposed Bayesian stats

5/ Coined the „sexy son hypothesis“

6/ Popularized *t*-distribution

7/ Created the ANOVA

It‘s Ronald Fisher!

?

Pokémon

Maximum Likelihood upsided

- ML is an ingenious method of estimating parameters
- It has beautiful properties:
  - *Consistency* = with increasing N, ML estimates (MLE) converge to the true values
  - *Efficiency* = MLE is the „best“ way to get these estimates
  - *Asymptotic normality* = with increasing N, MLE are normally distributed

- But this only holds when asssumptions are met

Maximum Likelihood downsides

- Assumptions of ML:
  - Large samples! (e.g., Gorsuch claims that for FA, N = 200 is probably enough if you have few variables, while N = 50 is not)
  - Multivariate normality (i.e., the variables **together** are distributed normally)
  - But you can assume any kind of distribution

- So it‘s a good idea to check N and the distributions but, luckily, MLE seems to be pretty robust against violations of the distributional assumptions at least

- However, MLE is very computationally intensive!

Multivariate normality

- n-dimensional generalization of univariate normality
- Variables are normal **together**
- When they are MVN, they are also normal each on their own
- But all of them being normal on their own does not imply they are MVN
- Weird stuff, right?

Multivariate normality

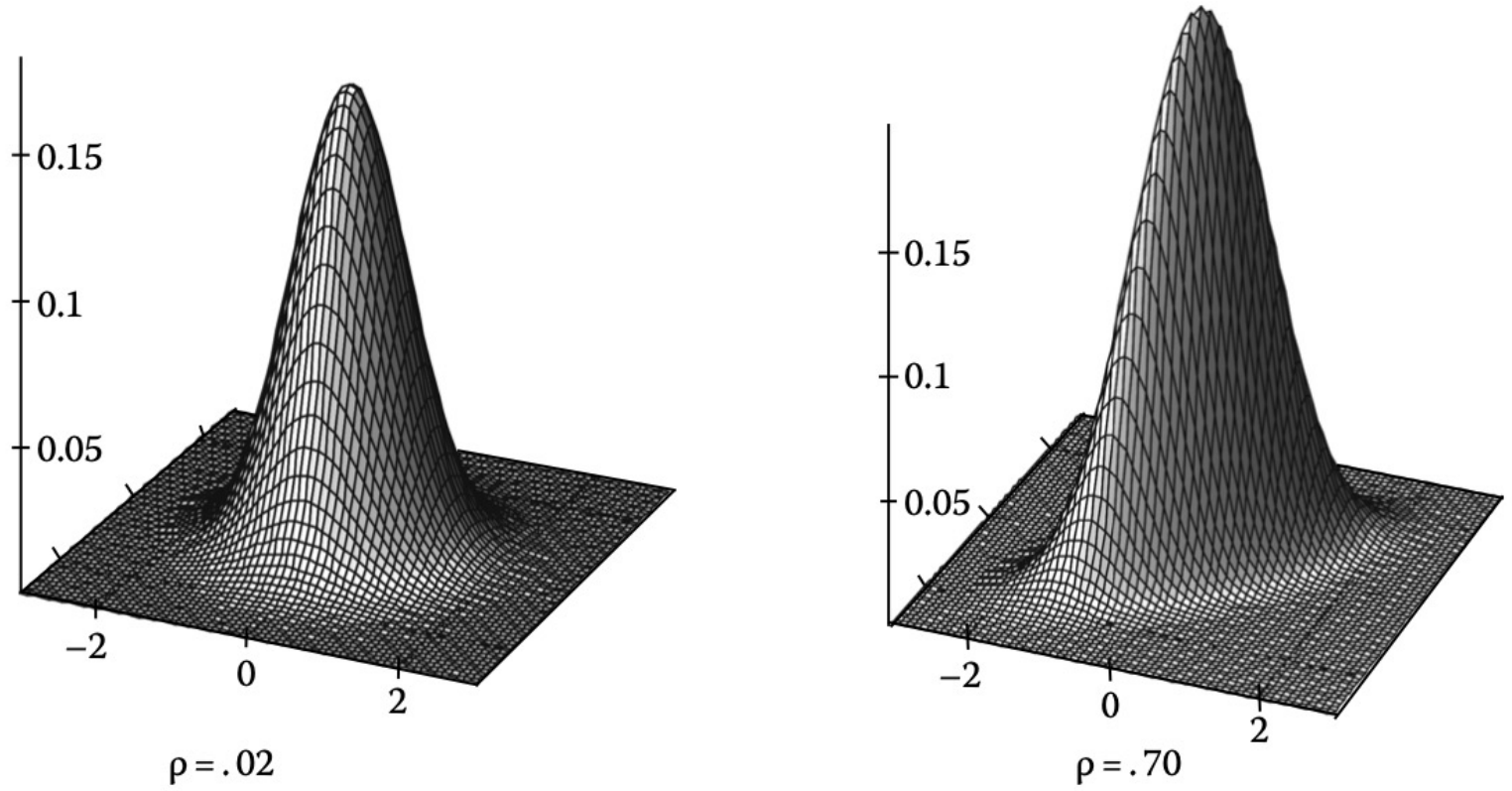


FIGURE 5.2
Graphs of density functions for the bivariate normal distribution for two variables with mean vector $\boldsymbol{\mu}' = [1, 2]$, variances of 1.00, and $\rho_{12} = .02$, and $\rho_{12} = .70$, respectively.

Multivariate normality

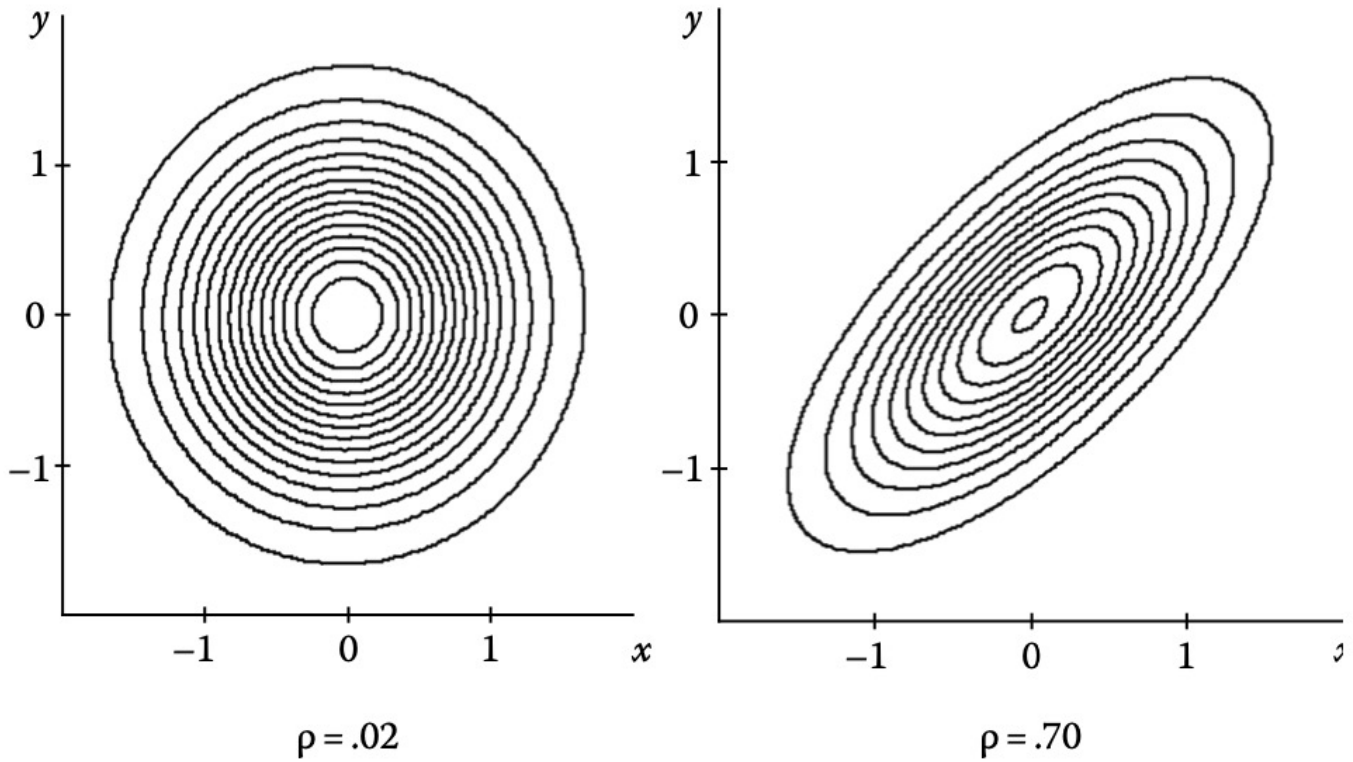


FIGURE 5.3
Contour plots of the bivariate normal distributions shown in Figure 5.2. Elliptical contour lines represent loci of points with equal density.

Probability density

- Coming back to the 1-dimensional world for simplicity, we can describe any normal distribution using its **mean** and **variance**
- After choosing these values, we can calculate the probability of x having a certain value given a normal distribution with the preset mean and variance:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = normalizing\ constant * e^{-\frac{1}{2}z^2}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2 e z^2}}$$

Probability density

- Well, and then we simply plug in the values.
- Let‘s say we are interested in the probability of x = 5 given mean = 3 and variance = 10

$$P(x = 5 \mid \mu = 3, \sigma^2 = 10) = \frac{1}{\sqrt{2\pi 10}} e^{-\frac{(5-3)^2}{2*10}} = .10$$

See? *e* z (pun intended)

The concept of likelihood

- Likelihood is about turning this problem around.
- You usually don‘t know the parameters of the distribution, right? But you know the values. So you can ask:

What set of parameters makes my observations the likeliest (most probable)?

Maximizing the likelihood

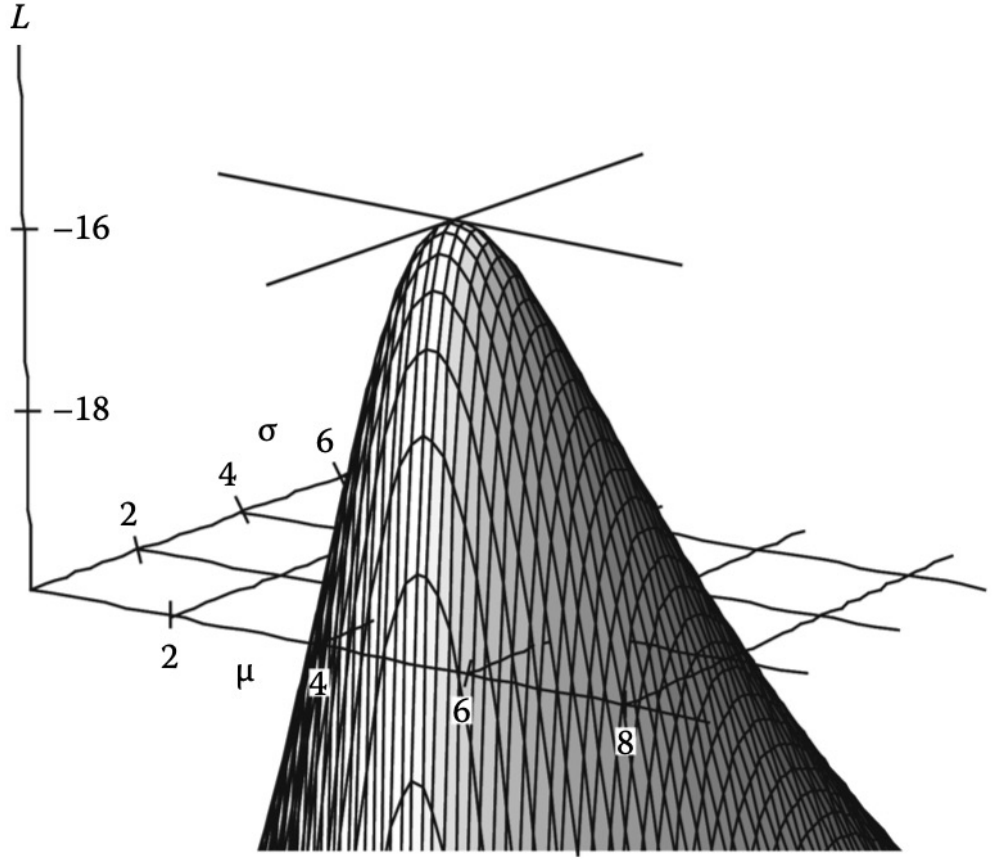
- We have a set of observations **x** and are interested in guessing what normal distribution they came from
- This requires a slight change in the formula:

$$\mathcal{L}(\boldsymbol{x}\,|\mu,\sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \prod_1^N e^{-\frac{(x-\mu)^2}{2\sigma^2}} =$$

$$N\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \sum_1^N \ln(-\frac{(x-\mu)^2}{2\sigma^2})$$

Maximizing the likelihood

- This badass formula creates a function for which we can find a maximum
- X axes are mean and variance
- Y axis is the likelihood value
- **The top of the mountain is the maximum likelihood** = where we can find the likeliest combination of parameters



Computing maximum likelihood

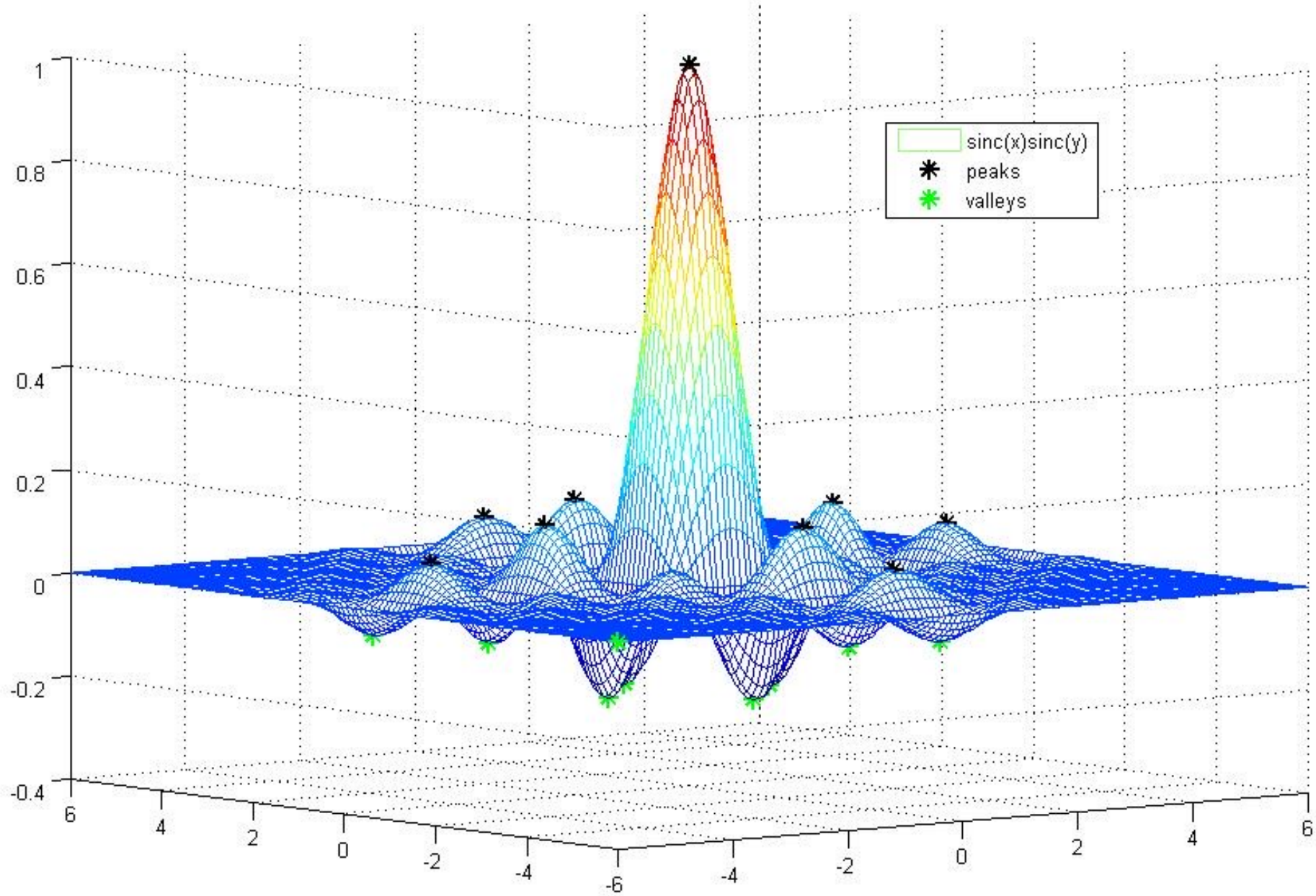
- If you now imagine extending this idea into *n* dimensions, you probably see that this forms an *n*-dimensional likelihood monster onto which we need to climb to find the top
- Computing it directly (using derivatives as shown in Mulaik) is possible only in simple cases
- Otherwise, one needs to use an iterative algorithm

MLE iterations

- These algorithms are mostly concerned with **navigating the parameter space** efficiently
- You see, the mountain is all the possible values our parameters can take.
- And there are rules to this madness – if you stumble upon a steep climb, you are probably on a good track to finding the top!
- So the algorithm computes the steepnes of a climb (2nd derivative) at a given point and then jumps in a certain direction according to the results

MLE iterations

- It‘s basically a game of hide and seek
- The goal is to find the top, and the steepnes of the *n*-dimensional mountain tells you whether you are hotter or colder
- Isn‘t that beautiful?


sinc(x)sinc(y)
peaks
valleys
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
6
4
2
0
-2
-4
-6
-4
-2
0
2
4
6

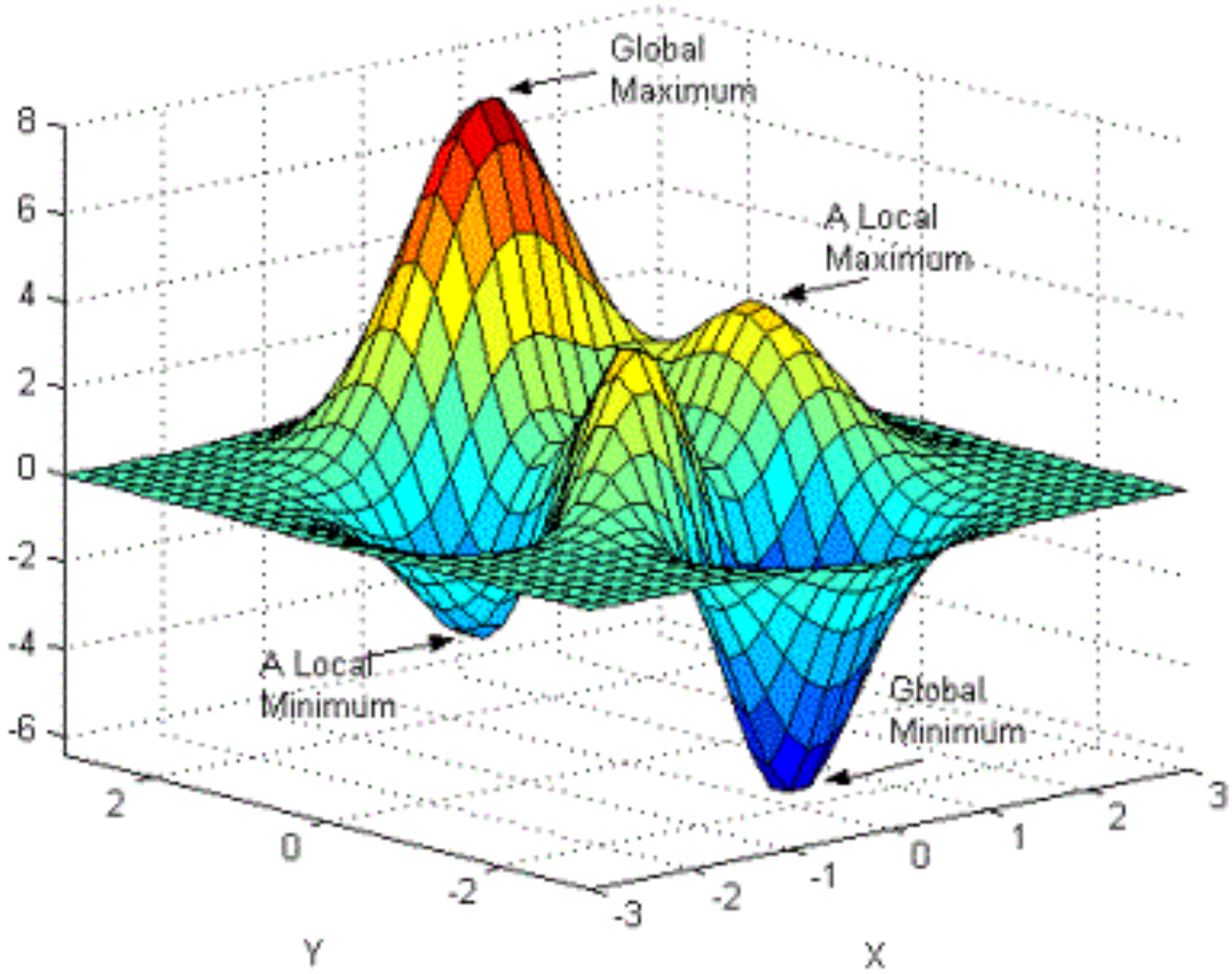
MLE algorithms

- As a sidenote, this means that you should not confuse MLE as an estimation technique with the algorithms of computing the result.
- It‘s always the same ML whether you choose:
  - Newton-Raphson
  - Expectation-Maximization
  - Metropolis-Hastings Robbins-Monro (MCMC)

- They only differ in how they navigate the *n*-dimensional mountain

Local maxima / minima

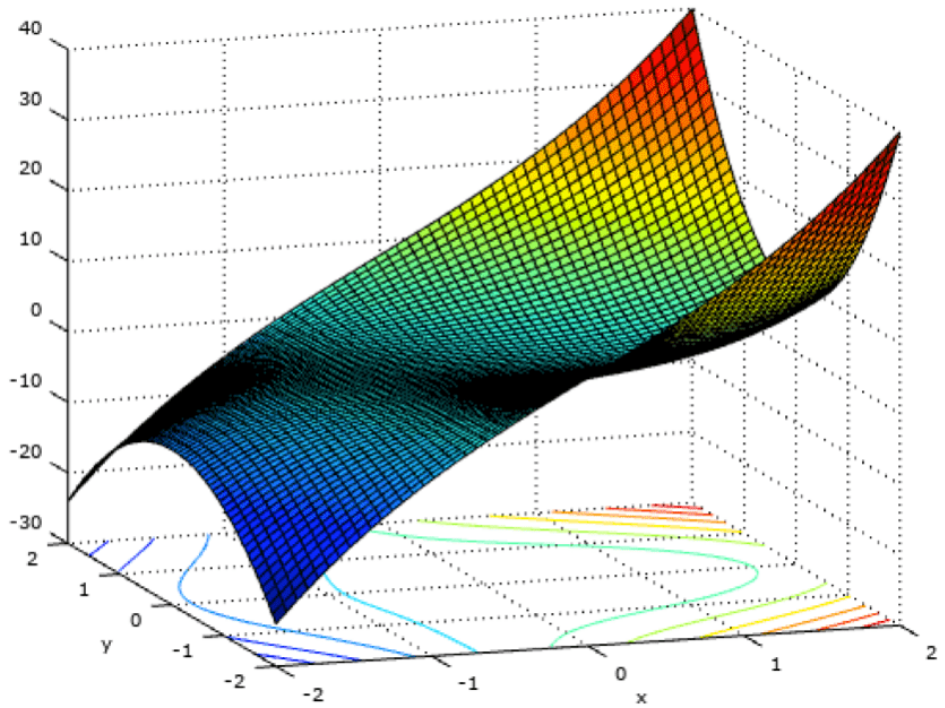
- The technical term for the top of the mountain (MLE) is the **global maximum**
- However, in some cases, the shape of the mountain can be strange, having multiple smaller peaks along with the highest peak
- These smaller peaks are called **local maxima**
- When the alogrithm misidentifies this peak for the top (thinking that you can only descend, not ascend further from that point), you get a wrong result


Global
Maximum
A Local
Maximum
A Local
Minimum
Global
Minimum
8
6
4
2
0
-2
-4
-6
2
0
-2
-3
-2
-1
0
1
2
3
Y
X

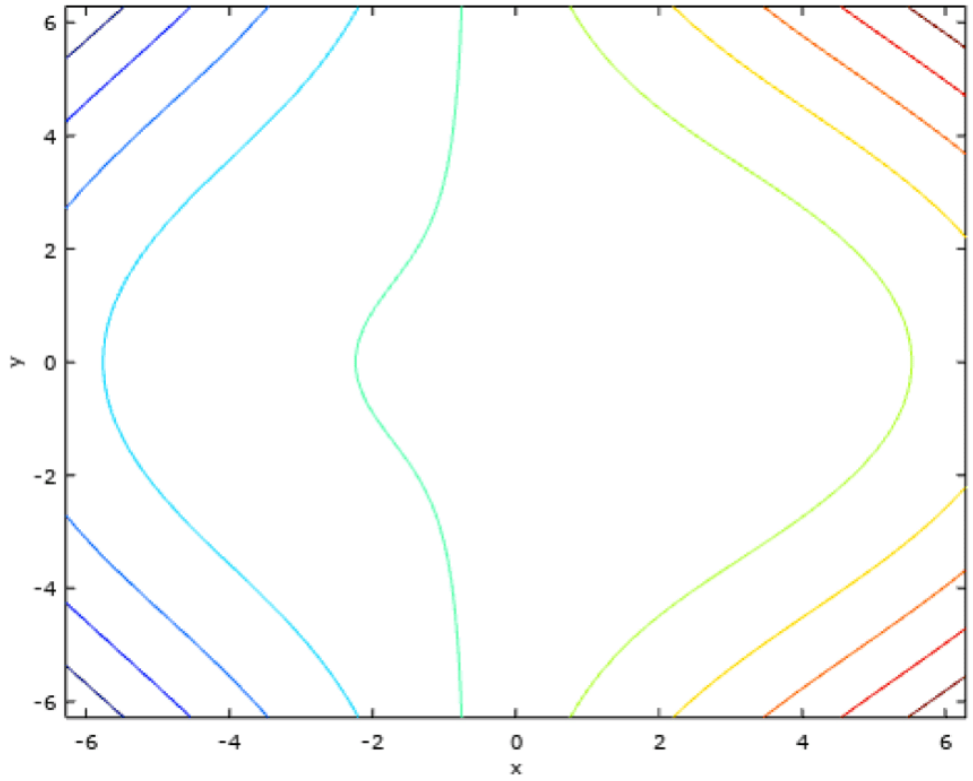
Algorithms – sidenote

- By the way, if you ever heard of the term **Hessian matrix**, then that‘s sort of a map to this n-dimensional monster we are exploring
- It is a matrix containing information about how steeply the monster we are finding the top of rises / descends at various points
- Mathematically, it is a matrix of second partial derivatives at critical points

Algorithms – sidenote



Graph of $g(x, y) = x^3 + 2y^2 + 3xy^2$



Contours of $g(x, y) = x^3 + 2y^2 + 3xy^2$

MLE in FA

- So, the goal of ML estimation is to find such $\widehat{\boldsymbol{\Lambda}}$ and $\widehat{\boldsymbol{D}_{\psi}}$ so that the value of the -2 x log-likelihood function is minimized (to ease computations)

- This function is again a discrepancy function – it is always larger than or equal to 0 and is zero if and only if the model-implied correlation matrix equals the sample correlation matrix

Maximum likelihood estimation

- The logic of ML estimation is very similar to that of (iterative) OLS:

1) Initial estimates of $\widehat{\boldsymbol{D}}_{\psi}$ are obtained (by SMCs or other means)

2) A maximum likelihood estimate of $\boldsymbol{\Lambda}$ is obtained, conditional on the estimated $\widehat{\boldsymbol{D}}_{\psi}$

3) A model-implied reduced correlation matrix is obtained, which completes the first iteration

4) New iteration is initiated with most recent estimates of $\widehat{\boldsymbol{D}}_{\psi}$

5) Iterations continue until convergence is achieved

Summary

- We have described three different methods for fitting the common factor model to sample data:

- Principal factors with prior communality estimates (noniterative)
- Ordinary least squares (iterative principal factors)
- Maximum likelihood

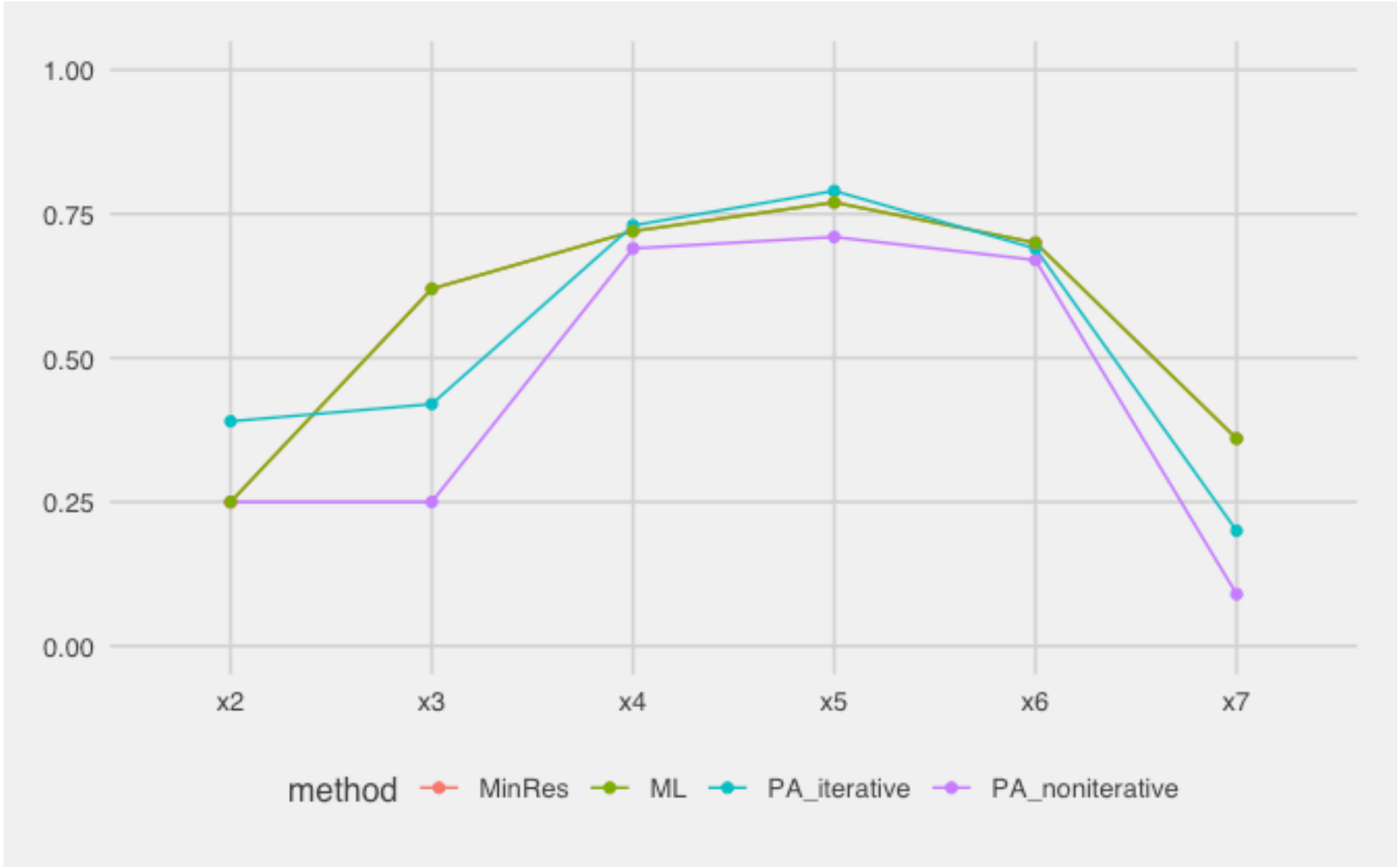
- These are *methods* for fitting the *model* to data. Many more methods exist, but OLS and ML are commonly available.

Summary

- But it’s all about some type of **discrepancy function** that represents the distance between the observed and the expected
- These functions take different forms as they all define the best solution differently
- OLS wants to minimize the residual sum of squares
- MLE wants to maximize the likelihood of the parameters given the observed data

Which is best?

- Minres / PA work when distributional assumptions are broken
- ML works well with large samples and allows model comparison
- However, there are simulation studies that really hype minres stating it performs the best under most conditions
- I'd recommend trying both minres and ML and checking whether the results meaningfully differ (i.e., you would write a different Discussion section if you were to use the other one)
- Usually, you might not see any real difference, so no need to worry
- This approach is called a **sensitivity analysis**


1.00
0.75
0.50
0.25
0.00
x2
x3
x4
x5
x6
x7
method
MinRes
ML
PA_iterative
PA_noniterative

What comes next?

- You‘ve learned a LOT (and I have as well alongside you)
- In 2 weeks, we meet for our last Vectors of Mind session (sniff)
- We will talk about applied topics like:
  - Model fit assessment
  - CFA vs. EFA
  - Extremely (and I mean **extremely**) cool approaches to hypothesis testing in FA
  - Good research practices concerning FA
  - Reporting standards for CFA
  - Bifactor models / ordinal FA