Exploratory Factor Analysis

Statistik-Kolloquium

André Meichtry

Departement Gesundheit
Berner Fachhochschule

7. Mai 2025

Exploratory Factor Analysis EFA

- Exploratory factor analysis is based on a formal model predicting observed variables from unobserved theoretical latent factors.
- Our data $\boldsymbol{Y} = (Y_1, \ldots, Y_j, \ldots, Y_k)^T$ consist of a
  - k-dimensional centered vector[1] of
  - observables (variables, items, indicators).
  - The data has covariance matrix Σ.

[1] In the following, vectors are in bold type. ^{T} stands for "transposed".

Model

m-factor model ($m < k$) for the k-dimensional observation vector $\boldsymbol{Y}$

$$
\begin{aligned}
Y_1 &= \lambda_{11} F_1 + \lambda_{12} F_2 + \cdots + \lambda_{1m} F_m + \epsilon_1 \\
Y_2 &= \lambda_{21} F_1 + \lambda_{22} F_2 + \cdots + \lambda_{2m} F_m + \epsilon_2 \\
&\vdots \\
Y_k &= \lambda_{k1} F_1 + \lambda_{k2} F_2 + \cdots + \lambda_{km} F_m + \epsilon_k
\end{aligned} \tag{1}
$$

- The factor scores $\boldsymbol{F} = (F_1, \ldots, F_l, \ldots, F_m)^T$ are the scores on the common factors,
- the $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_j, \ldots, \epsilon_k)^T$ are the specific factors with $\text{Var}(\epsilon_j) = \sigma_j^2$ (the uniquenesses).
- The factor scores $F_1, ..., F_m$ are random and unknown. The constraints on the factor scores are that they are uncorrelated with expectation 0 and unit variance, $\text{Cov}(F_i, F_j) = 0, \mathrm{E}(F_i) = 0, \text{Var}(F_i) = 1$.
- The λ_{jl} are the factor loadings of the j-th variable/item on the l-th factor.
- The $k \times m$ matrix Λ with elements λ_{jl} represent the loadings matrix.

Model

(1) can be written compactly in vector/matrix notation

$$\boldsymbol{Y} = \Lambda \boldsymbol{F} + \boldsymbol{\epsilon} \tag{2}$$

$$= F_1 \boldsymbol{\lambda}_{.1} + \cdots + F_m \boldsymbol{\lambda}_{.m} + \boldsymbol{\epsilon}, \tag{3}$$

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_l \\ \vdots \\ Y_k \end{pmatrix} = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1m} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2m} \\ \vdots & & & \\ \lambda_{j1} & \lambda_{j2} & \dots & \lambda_{jm} \\ \vdots & & & \\ \lambda_{k1} & x_{k2} & \dots & \lambda_{km} \end{bmatrix} \begin{pmatrix} F_1 \\ \vdots \\ F_m \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_l \\ \vdots \\ \epsilon_k \end{pmatrix}. \tag{4}$$

- Λ is the $k \times m$ loadings matrix with factor loadings λ_{jl} of the j-th variable/item on the l-th factor and $\boldsymbol{\lambda}_{.l}$ is the lth column of Λ representing the lth factor.
- Take care to not mess up factor versus factor scores.
- Factor scores are random numbers, factors are vectors in space.

Model

- The model can be directly written for the $k \times k$ covariance matrix Σ:

Model for covariance matrix

$$\Sigma = \Lambda\Lambda^T + \Psi, \tag{5}$$

with

- $\Psi = diag(\sigma_1^2, \ldots, \sigma_k^2)$ (Diagonal matrix with the variances als elements)
- Λ is the $k \times m$ factor loadings matrix.
- $\Lambda\Lambda^T$, Ψ (and, of course Σ) are $k \times k$ matrices[2].

[2] $\Sigma = \mathrm{Cov}(\boldsymbol{Y}) = \mathrm{E}(\boldsymbol{Y}\boldsymbol{Y}^T) = \Lambda\, \mathrm{Cov}(\boldsymbol{F})\Lambda^T + \mathrm{Cov}(\boldsymbol{\epsilon}) = \Lambda\Lambda^T + \mathrm{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) = \Lambda\Lambda^T + \Psi$

Model

- Variances and the covariances among the observed variables/items can be decomposed into:
  - component attributable to the underlying factors
  - the measurement error variances and covariances
- The statistical problem is to estimate the elements on the right-hand side of the equation using the information in the observed variance-covariance matrix.

Conditional independence

- It is assumed that the responses on the observables

$$Y_1, \dots, Y_k$$

are the result of an individual's position on the latent variable(s)

$$F_1, \dots, F_m,$$

and that the observables have nothing in common after controlling for the latent variable(s).

- This is called local independence or conditional independence; we have seen this principle in other latent variable models:
  - Rasch Model (with unknown θ)
  - Mixed Models (with unknown random effects $\boldsymbol{U}_i$)
  - In Reflective models in CTT (unknown η).

Conditional independence

- This means that the latent variable explains why the observed items are related to another.
- Once we know $\boldsymbol{F}$ (conditioning on $\boldsymbol{F}$), knowledge about Y_1 for example does provide no information about $Y_2, \ldots, Y_k$, see Figure 1.
- Example: Height and vocabulary are not independent; but they are conditionally independent if you know age.

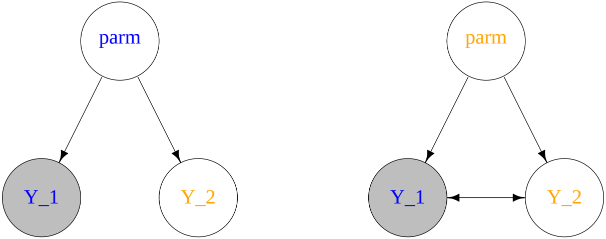


Figure: Conditional independence. blue: known, red: unknown, parm: unknown latent

Eigenvalue of a factor

Eigenvalue of factor l

$$ev_l = \sum_{j=1}^{k} \lambda_{jl}^2 \tag{6}$$

- The eigenvalue of factor l is the sum of the squared loadings of all variables/items on the corresponding factor,
- representing the amount of variance of the data that is explained by factor l.
- The eigenvalue divided by the number of variables represents the percentage of the variance explained by the factor.
- Factors with eigenvalues smaller than one explain less variance than one "average" variable.

Communality of a variable

Communality of variable j

$$com_j = \sum_{l=1}^{m} \lambda_{jl}^2 \tag{7}$$

- The communality of variable/item j is the sum of the squared loadings of the corresponding variable/item on all factors,
- representing the explained variance of variable j by the factors $F_1, \ldots, F_m$.

Factor rotation

- An important property of factor analysis is that the factor loadings (a solution Λ) is identified only up to orthogonal rotations.
- Varimax rotation is a popular rotation for orthogonal rotation:
  - Varimax maximizes the variances of the squared loadings for each factor
  - Moderate loadings will become larger oder smaller and can better be attributed to factors
  - The aim is to clarify the structure of the loadings matrix.
  - The rotation does not change! communalities and the total amount of the variance explained by the factors.

One-Factor Model

- Consider two observables Y_1 and Y_2 and the one-factor model.

$$Y_1 = \lambda_{11} F + \epsilon_1$$
$$Y_2 = \lambda_{21} F + \epsilon_2$$

- That is an extension of CTT with latent η, where $\lambda_{11} = \lambda_{21}$ and $\text{Var}(\epsilon_j) = \sigma^2$.

$$Y_1 = \eta + \epsilon_1$$
$$Y_2 = \eta + \epsilon_2$$

One-Factor Model

- $\text{Var}(Y_1) = \lambda_{11}^2 + \sigma_{\epsilon_1}^2$.
- $\text{Var}(Y_2) = \lambda_{21}^2 + \sigma_{\epsilon_2}^2$.
- The communalities of Y_1 and Y_2 are λ_{11}^2 and λ_{21}^2, respectively.
- The uniquenesses of Y_1 and Y_2 are $\sigma_{\epsilon_1}^2$ and $\sigma_{\epsilon_2}^2$, respectively.
- The eigenvalue of F is $\lambda_{11}^2 + \lambda_{21}^2$.
- $\text{Cov}(Y_1, Y_2) = \lambda_{11}\lambda_{21} \text{Cov}(F, F) = \lambda_{11}\lambda_{21}$.
- When the value of F is known (fixed), then the covariance between Y_1 and Y_2 is 0.
- Thus, we have conditional independence: Y_1 and Y_2 are independent, given F,

$$\text{Cov}(Y_1, Y_2 \mid F) = 0.$$

Implementation in R

- library `psych`
- Factor analysis is implemented with function `fa()`

Example Data

- The Eight Physical Variables problem is taken from Harman (1976)
- It represents the correlations between eight physical variables for $n = 305$ girls.
- The two correlated clusters represent
  - four measures of “lankiness” (“Schlankheit”) and
  - four measures of “stockiness” (“Stämmigkeit”).
- The original data were selected from 17 variables reported in an unpublished dissertation by Mullen (1939).
-
```r
library(psych)
## ?Harman.8
```

Example Data

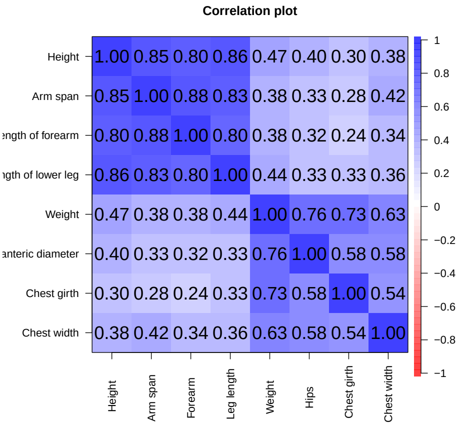
Correlation matrix

```
X <- Harman.8
X

##                         Height Arm span Forearm Leg length Weight  Hips Chest girth Chest width
## Height                   1.000    0.846   0.805      0.859  0.473 0.398       0.301       0.382
## Arm span                 0.846    1.000   0.881      0.826  0.376 0.326       0.277       0.415
## Length of forearm        0.805    0.881   1.000      0.801  0.380 0.319       0.237       0.345
## Length of lower leg      0.859    0.826   0.801      1.000  0.436 0.329       0.327       0.365
## Weight                   0.473    0.376   0.380      0.436  1.000 0.762       0.730       0.629
## Bitrochanteric diameter  0.398    0.326   0.319      0.329  0.762 1.000       0.583       0.577
## Chest girth              0.301    0.277   0.237      0.327  0.730 0.583       1.000       0.539
## Chest width              0.382    0.415   0.345      0.365  0.629 0.577       0.539       1.000
```

Example Data

Correlation matrix

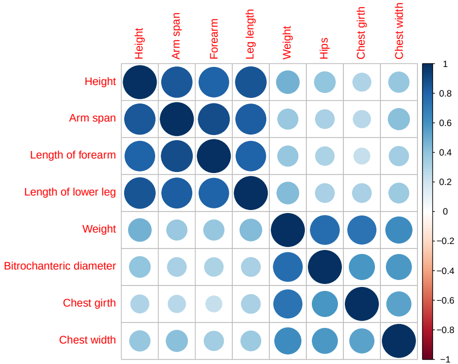
```r
cor.plot(X, las = 2)
```



Example Data

Correlation matrix (Alternative for plot)

```
corrplot::corrplot(X, method = "circle")
```



Sampling adequacy

- Ask Bartlett test whether correlation matrix is identity matrix (elements outside the diagonal are 0, and 1 on the diagonal)
- If correlation matrix is "near" the identity matrix, then a FA would not be adequate.
- Should be significant

```
cortest.bartlett(R = X, n = 305)
```

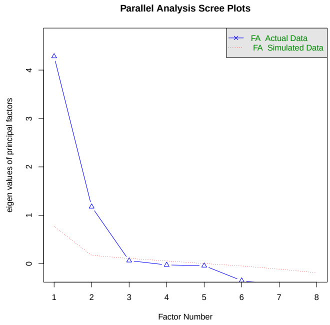
```
## $chisq
## [1] 2086
##
## $p.value
## [1] 0
##
## $df
## [1] 28
```

Estimation methods

- There are different estimation methods
- We do not go into details
- We will use Maximum Likelihood

Number of factors to extract

- Parallel analysis is the gold-standard for determining the number of factors to extract.
- Performed by extracting factors until the eigenvalues of the real data were less than the corresponding eigenvalues of a random data set of the same size.
```
fa.parallel(X, n.obs = 305, fa = "fa", fm = "ML")
```




```
## Parallel analysis suggests that the number of factors =  2  and the number of components =  NA
```

Estimation: fa() from package psych

```
args(fa)

## function (r, nfactors = 1, n.obs = NA, n.iter = 1, rotate = "oblimin",
##     scores = "regression", residuals = FALSE, SMC = TRUE, covar = FALSE,
##     missing = FALSE, impute = "none", min.err = 0.001, max.iter = 50,
##     symmetric = TRUE, warnings = TRUE, fm = "minres", alpha = 0.1,
##     p = 0.05, oblique.scores = FALSE, np.obs = NULL, use = "pairwise",
##     cor = "cor", correct = 0.5, weight = NULL, n.rotations = 1,
##     hyper = 0.15, smooth = TRUE, ...)
## NULL
```

Estimation: Maximum likelihood fit with fa()

- We go on with 2 factors
- We start with an unrotated solution using the maximum likelihood estimation method in fa(fm="ML")
-
```
ml0 <- fa(r = X, nfactors = 2, n.obs = 305, fm = "ML", rotate = "none")
```

Unrotated solution

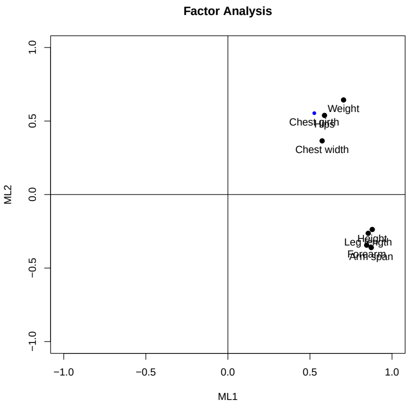
Loadings, communalities and uniquenesses

```
FAresults0 <- data.frame(unclass(ml0$loadings), h2 = ml0$communalities, u2 = ml0$uniqueness)
round(FAresults0, digits = 3)

##               ML1    ML2    h2    u2
## Height      0.880 -0.237 0.830 0.170
## Arm span    0.874 -0.360 0.893 0.107
## Forearm     0.846 -0.344 0.834 0.166
## Leg length  0.855 -0.263 0.801 0.199
## Weight      0.705  0.644 0.911 0.089
## Hips        0.589  0.538 0.636 0.364
## Chest girth 0.527  0.554 0.584 0.416
## Chest width 0.574  0.365 0.463 0.537
```

```
print(ml0$loadings, sort = TRUE, digits = 3)

##
## Loadings:
##             ML1    ML2
## Height       0.880 -0.237
## Arm span     0.874 -0.360
## Forearm      0.846 -0.344
## Leg length   0.855 -0.263
## Weight       0.705  0.644
## Hips         0.589  0.538
## Chest width  0.574  0.365
## Chest girth  0.527  0.554
##
##                  ML1   ML2
## SS loadings    4.434 1.518
## Proportion Var 0.554 0.190
## Cumulative Var 0.554 0.744
```

Unrotated solution

- Unrotated solution is difficult to interpret
- One item on one factor, the others on the other factor
- ```
  plot(ml0, xlim = c(-1, 1), ylim = c(-1, 1), labels = colnames(X))
  ```


  ```

Unrotated solution

- Reproduce the explained variance

- ```
  l <- loadings(ml0)
  ev <- apply(l, 2, function(x) sum(x^2))  #apply a function over all columns
  ev  #eigenvalues

  ##  ML1  ML2
  ## 4.43 1.52

  propVar <- ev/8  #Proportion explained (The sum of all eigenvalues is equal the number of all variables)
  propVar

  ##   ML1   ML2
  ## 0.554 0.190

  cumsum(propVar)  #cumulative relative eigenvalues

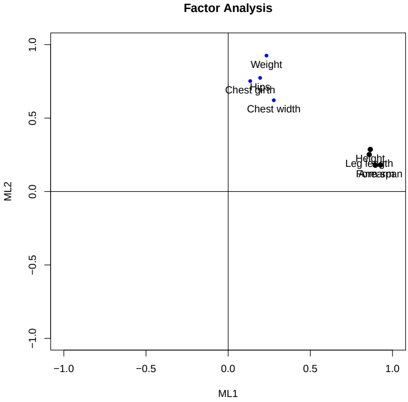
  ##   ML1   ML2
  ## 0.554 0.744
  ```
  ```

Estimation: Rotated solution

- The solution is not unique
- To clarify the structure of the loadings matrix, we
- Rotate the solution with the varimax method.
- ```
  mlRot <- fa(r = X, nfactors = 2, n.obs = 305, rotate = "varimax", fm = "ML")
  ```
  ```

Estimation: Rotated solution

```
FAresults <- data.frame(unclass(mlRot$loadings), h2 = mlRot$communalities, u2 = mlRot$uniqueness)
```

```
round(FAresults, digits = 3)

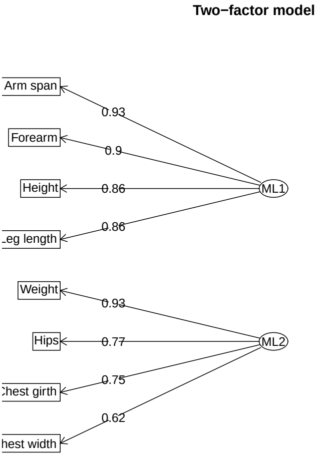
##               ML1   ML2    h2    u2
## Height      0.865 0.287 0.830 0.170
## Arm span    0.927 0.181 0.893 0.107
## Forearm     0.895 0.179 0.834 0.166
## Leg length  0.859 0.252 0.801 0.199
## Weight      0.233 0.925 0.911 0.089
## Hips        0.194 0.774 0.636 0.364
## Chest girth 0.134 0.752 0.584 0.416
## Chest width 0.278 0.621 0.463 0.537

print(mlRot$loadings, sort = TRUE, digits = 3)

##
## Loadings:
##             ML1   ML2
## Height      0.865 0.287
## Arm span    0.927 0.181
## Forearm     0.895 0.179
## Leg length  0.859 0.252
## Weight      0.233 0.925
## Hips        0.194 0.774
## Chest girth 0.134 0.752
## Chest width 0.278 0.621
##
##                  ML1   ML2
## SS loadings    3.335 2.617
## Proportion Var 0.417 0.327
## Cumulative Var 0.417 0.744
```

Estimation: Rotated solution

- Default cutoff for printing loadings is 0.1
- Example: change to 0.3
-

```
print(mlRot$loadings, sort = TRUE, digits = 3, cutoff = 0.3)
##
## Loadings:
##             ML1   ML2
## Height      0.865
## Arm span    0.927
## Forearm     0.895
## Leg length  0.859
## Weight            0.925
## Hips              0.774
## Chest girth       0.752
## Chest width       0.621
##
##                  ML1   ML2
## SS loadings    3.335 2.617
## Proportion Var 0.417 0.327
## Cumulative Var 0.417 0.744
```

Loadings from Rotated solution

- Better interpretation
- ```
  plot(mlRot, xlim = c(-1, 1), ylim = c(-1, 1), labels = colnames(X))
  ```


  ```

“Lankiness” and “Stockiness” as latent factors

- Two factors explain the correlation of the 8 variables
-
```
fa.diagram(mlRot, simple = TRUE, digits = 2, main = "Two-factor model")
```



Software

```
toLatex(sessionInfo(), locale = FALSE)
```

- R version 4.5.0 (2025-04-11), x86_64-pc-linux-gnu
- Running under: Ubuntu 22.04.5 LTS
- Matrix products: default
- BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0
- LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0
- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, utils
- Other packages: bivariate 0.7.0, dplyr 1.1.4, fractional 0.1.3, futile.logger 1.4.3, ggmcmc 1.5.1.1, ggplot2 3.5.2, knitr 1.50, LearnBayes 2.15.1, psych 2.4.12, tidyr 1.3.1, venn 1.12, VennDiagram 1.7.3, xtable 1.8-4
- Loaded via a namespace (and not attached): admisc 0.35, barsurf 0.7.0, cli 3.6.5, colorspace 2.1-0, compiler 4.5.0, corrplot 0.92, dichromat 2.0-0.1, evaluate 1.0.3, farver 2.1.2, formatR 1.14, futile.options 1.0.1, generics 0.1.3, GGally 2.2.1, ggstats 0.6.0, glue 1.8.0, gtable 0.3.6, highr 0.11, KernSmooth 2.23-22, kubik 0.3.0, lambda.r 1.2.4, lattice 0.22-6, lifecycle 1.0.4, magrittr 2.0.3, mnormt 2.1.1, mvtnorm 1.2-4, nlme 3.1-164, parallel 4.5.0, pillar 1.10.2, pkgconfig 2.0.3, plyr 1.8.9, purrr 1.0.4, R6 2.6.1, RColorBrewer 1.1-3, Rcpp 1.0.14, rlang 1.1.6, scales 1.4.0, tibble 3.2.1, tidyselect 1.2.1, tools 4.5.0, vctrs 0.6.5, withr 3.0.2, xfun 0.52

Bibliography

[Alb18] Jim Albert. *LearnBayes: Functions for Learning Bayesian Inference*, 2018. R package version 2.15.1.

[Che22] Hanbo Chen. *VennDiagram: Generate High-Resolution Venn and Euler Plots*, 2022. R package version 1.7.3.

[DSR+19] David B. Dahl, David Scott, Charles Roosen, Arni Magnusson, and Jonathan Swinton. *xtable: Export Tables to LaTeX or HTML*, 2019. R package version 1.8-4.

[Dus24] Adrian Dusa. *venn: Draw Venn Diagrams*, 2024. R package version 1.12.

[Fer21] Xavier Fernández i Marín. *ggmcmc: Tools for Analyzing MCMC Simulations from Bayesian Inference*, 2021. R package version 1.5.1.1.

[iM16] Xavier Fernández i Marín. ggmcmc: Analysis of MCMC samples and Bayesian inference. *Journal of Statistical Software*, 70(9):1–20, 2016.

[Rev24] William Revelle. *psych: Procedures for Psychological, Psychometric, and Personality Research*, 2024. R package version 2.4.12.

[Row16] Brian Lee Yung Rowe. *futile.logger: A Logging Utility for R*, 2016. R package version 1.4.3.

[Spu21] Abby Spurdle. *bivariate: Bivariate Probability Distributions*, 2021. R package version 0.7.0.

[Ven16] Bill Venables. *fractional: Vulgar Fractions in R*, 2016. R package version 0.1.3.

[WCH+25] Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington, and Teun van den Brand. *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*, 2025. R package version 3.5.2.

[WFH+23] Hadley Wickham, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. *dplyr: A Grammar of Data Manipulation*, 2023. R package version 1.1.4.

[Wic16] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

[WVG24] Hadley Wickham, Davis Vaughan, and Maximilian Girlich. *tidyr: Tidy Messy Data*, 2024. R package version 1.3.1.

[Xie14] Yihui Xie. knitr: A comprehensive tool for reproducible research in R. In Victoria Stodden, Friedrich Leisch, and Roger D. Peng, editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC, 2014. ISBN 978-1466561595.

[Xie15] Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. ISBN 978-1498716963.

[Xie25] Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2025. R package version 1.50.