Explorative Faktorenanalyse

Bißantz, Jalynskij, Kupffer & Prestele

BF3 Testtheorie

Section 1

Einstieg

Einstieg

Example

Hinter unterstehendem Link verbirgt sich ein Fragebogen der Platform `openpsychometrics` (eine gute Quelle um kostenlos Datensätze für Analysen zu ergattern). Schauen Sie sich den Big-Five-Fragebogen einmal an. Wenn Sie möchten, beantworten Sie Ihn kurz. Im Anschluss daran stellen wir uns auf dieser Grundlage der Large Dataset Challenge!

https://openpsychometrics.org/tests/IPIP-BFFM/[1]

- Zeit: 2-3 Minuten

[1] Für das Projekt, siehe: https://openpsychometrics.org/

Section 2

Die “Large Data Set Challenge”

Die “Large Data Set Challenge”

Example

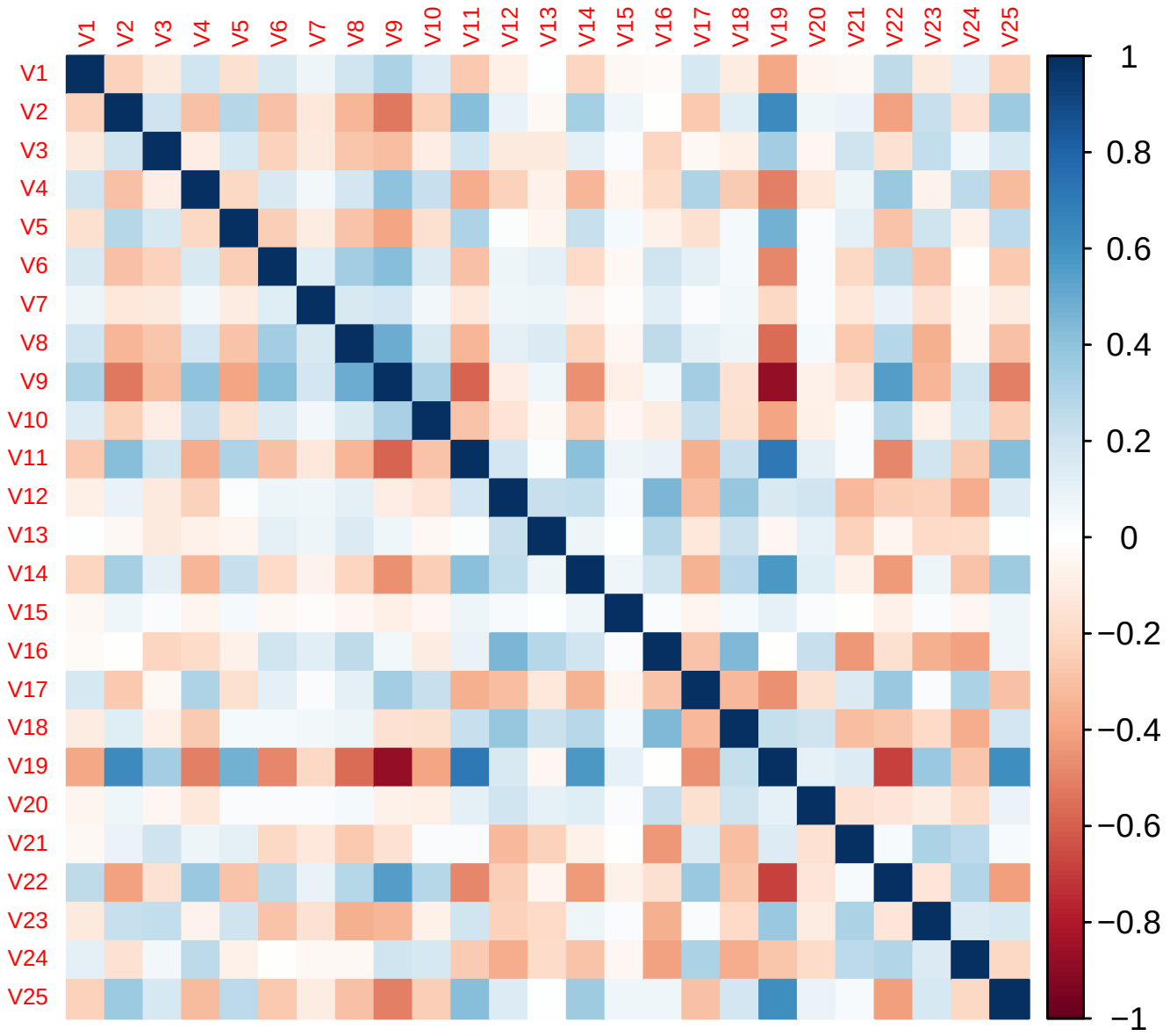
Stellen Sie sich vor, die von Ihnen soeben gesehenen/beantworteten Fragen ergäben die Korrelationsmatrix R auf der nächsten Folie. Die "Large Data Set Challenge" (LDC) lautet: Erkennen Sie eine Struktur in den Daten? Das heißt konkreter, finden Sie homogene Itemgruppen? Besprechen Sie sich mit Ihrem Nachbarn.

Welche Items könnte man Ihrer Meinung nach zu Itemgruppen zusammenfassen?

- Zeit: 2-3 Minuten

Anmerkung: Nein, das sind nicht Ihre Antworten; `V1 - V25` sind Zufallsvariablen!

Übungsaufgabe 1: Struktur erkennen & Itemgruppen finden



Die “Large Data Set Challenge” (..in theory)

Large Data Set Challenge

Mit zunehmender Itemzahl nimmt die Anzahl der Korrelationen, die für eine Analyse zu berücksichtigen sind, schnell zu. Die "Challenge" ist eine mögliche Struktur zu entdecken! (..Wie sie eben in ihrer visuellen Zuordnungen selbst gemerkt haben)

In a Nutshell:

- Problem: Anzahl der Korrelationen
- z.B.: 25 Items $\widehat{=}$ 300 Korrelationen
- Krux: Struktur erkennen
- $\Leftrightarrow$ finde: hoch korrelierende Itemgruppen

Explorative Faktorenanalyse & Large Data Set Challenge

- (ein) Hilfsmittel für die LDC: (explorative) *Faktorenanalyse*
- andere Helferlein zur *Datenreduktion* (eine Auswahl):
  - Hauptkomponentenanalyse
  - Clusteranalyse
  - Explorative Likertskalierung
  - (Non-) Metric Data Scaling (Distanzmatrizen!)

Wichtige Unterscheidung: "meaningful" (Faktoren) vs. "full compression" (Komponenten)

Fokus (heute): Hauptkomponentenanalyse (PCA), Hauptachsenanalyse (PAF), Maximum Likelihood Faktorenanalyse (MLF)

Section 3

Fakotrenanalyse: “A hurdle race”

Fakotrenanalyse: “A hurdle race”

1. Hürde: Extraktionsproblem
2. Hürde: Rotationsproblem
3. Hürde: Problem der Anzahl zu extrahierender Faktoren

“Unfortunately, factor analysis is frequently misunderstood and often misused. Some researchers appear to use factor analysis as a kind of divining rod, hoping to find gold hidden underneath tons of dirt. But there is nothing magical about the technique. [...] Factor analysis will yield meaningful results only when the research was meaningful to begin with.” Gregory (2014, S. 165)

Vorbereitung: "Playing Creator" – 2 Latente Variablen erschaffen

Bevor wir loslegen, wollen wir eine Datenmatrix X und eine Korrelationsmatrix R simulieren.[2]

```
# Faktorladungen; 8 items
load_F1 <- c(0.6, -0.3, 0.5, 0.7, 0.1, 0.2, 0.2, 0.3)
load_F2 <- c(-0.1, 0.1, 0.1, 0.1, -0.7, 0.5, -0.6, 0.7)
fx <- cbind(load_F1, load_F2)
# Zwischenfaktorkorrelation
phi <- diag(rep(1, 2)) ; phi[1, 2] <- phi[2, 1] <- 0.6
# Struktur
S <- psych::sim.structure(fx, phi, n=1000)
# Korrelations- und Datenmatrix
R <- S$r ; X <- S$observed  # R <- cor(X)
```

[2] Nur in simulierten Welten (in denen wir den richtigen Ausgang kennen), können wir prüfen, ob unsere Methoden das "korrekte" Ergebnis reproduzieren. Sie können sich aber auch vorstellen, dass das Ihre Antworten auf den Fragebogen seien. Dann wären Sie der generative Prozess, den wir hier simulieren.

Section 4

Extraktionsproblem

Extraktionsproblem

Extraktionsproblem

Wie extrahieren wir die zur Reproduktion nötigen Faktoren/Komponenten?

1. Lösung: Bestimmung der "Principal Components" (PCA)[3]
2. Lösung: Iterative Bestimmung der "Principal Components" (PAF)
3. Lösung: Finde die plausibelsten ("most likely") Werte (MLF)

[3] Faktoren- und Hauptkomponentenanylse sind Verfahren mit unterschiedlichen Zielen! Beide Klassen nutzen unterschiedliche Modelle und machen unterschiedliche Basisannahmen. Dennoch werden sie häufig in der Literatur unter dem Deckmantel "explorative Faktorenanalyse" vereinheitlicht.

Let's do it! (..in R): PCA

```r
# Old School!
# (dino_pca <- princomp(X, cor=TRUE))
# New School: Datenmatrix? R -> X
(fit_pca <- psych::principal(R, nfactors = 2,
                             rotate = "none"))

## Komponentenladungen
fit_pca$loadings
## Kommunalitäten
fit_pca$communality
## Einzigartigkeit
fit_pca$uniquenesses
## Eigenwerte
fit_pca$values
```

Übungsaufgabe 2: Selbstexperiment

Example

Versuchen Sie es nun selbst! Nutzen Sie den Code zur Übungsaufgabe 2 `06-EFA.R` und führen eine PCA durch. Interpretieren Sie die entsprechenden Kennwerte für Ihr Modell und präsentieren Sie diese Ihrem Nachbarn.

- Zeit: 10 Minuten
- Replikation: `set.seed(123)`
- Anmerkung: Konzepte & Output verstehen $\gg$ Codes verstehen!

Anmerkung: Normalerweise kennen Sie die Anzahl der zu extrahierenden Komponenten/Faktoren nicht. Wie man dieses Problem angeht, dazu gleich mehr!

Let's do it! (..in R): Hauptachsenanalyse (PAF)

```
(fit_paf <- psych::fa(R, nfactors=2,
                      rotate="none",
                      fm="pa"))
# Kommunalitäten
fit_paf$communality
# Eigenwerte
fit_paf$e.values
# Einzigartigkeit
fit_paf$uniquenesses
```

Übungsaufgabe 3: Selbstexperiment

Example

Versuchen Sie es nun selbst! Nutzen Sie den Code zur Übungsaufgabe 3 `06-EFA.R` und führen Sie eine Hauptachsenanalyse (PAF) durch. Vergleichen Sie die Ergebnisse der PAF mit denen der PCA. Bestehen Unterschiede zwischen den Ergebnissen?

- Zeit: 10 Minuten
- Replikation: `set.seed(123)`
- Anmerkung: Konzepte & Output verstehen $\gg$ Codes verstehen!

Primer: Maximum Likelihood Faktorenanalyse (MLF)

MLF bietet als Lösung die plausibelsten (eng. "most likely") (Parameter-) Werte zur Reproduktion der Informationen in der Korrelationsmatrix an.

$$X_i = \lambda_j * \xi_i + \epsilon_i \tag{1}$$

$$R^* = \Lambda \Phi \Lambda' + \Psi \tag{2}$$

- R: Korrelationsmatrix
- X_i: Item i
- λ, Λ: Ladung, Ladungsmatrix
- ξ: Faktor
- Φ: Zwischenfaktorkorrelationsmatrix
- Ψ: Einzigartigkeit

Let's do it (..in R): Maximum Likelihood Faktorenanalyse (MLF)

```
(fit_mlf <- psych::fa(R, nfactors=2,
                      rotate="none",
                      fm="ml"))
# Kommunalitäten
fit_mlf$communality
# Eigenwerte
fit_mlf$e.values
# Einzigartigkeit
fit_mlf$uniquenesses
```

Übungsaufgabe 4: Selbstexperiment

Example

Versuchen Sie es nun selbst! Nutzen Sie den Code zur Übungsaufgabe 4 in `06-EFA.R` und führen Sie eine Maximum Likelihood Faktorenanalyse (MLF) durch. Vergleichen Sie die Ergebnisse auch mit den anderen Methoden.

- Zeit: 10 Minuten
- Replikation: `set.seed(123)`
- Anmerkung: Konzepte & Output verstehen $\gg$ Codes verstehen!

Section 5

Rotationsproblem

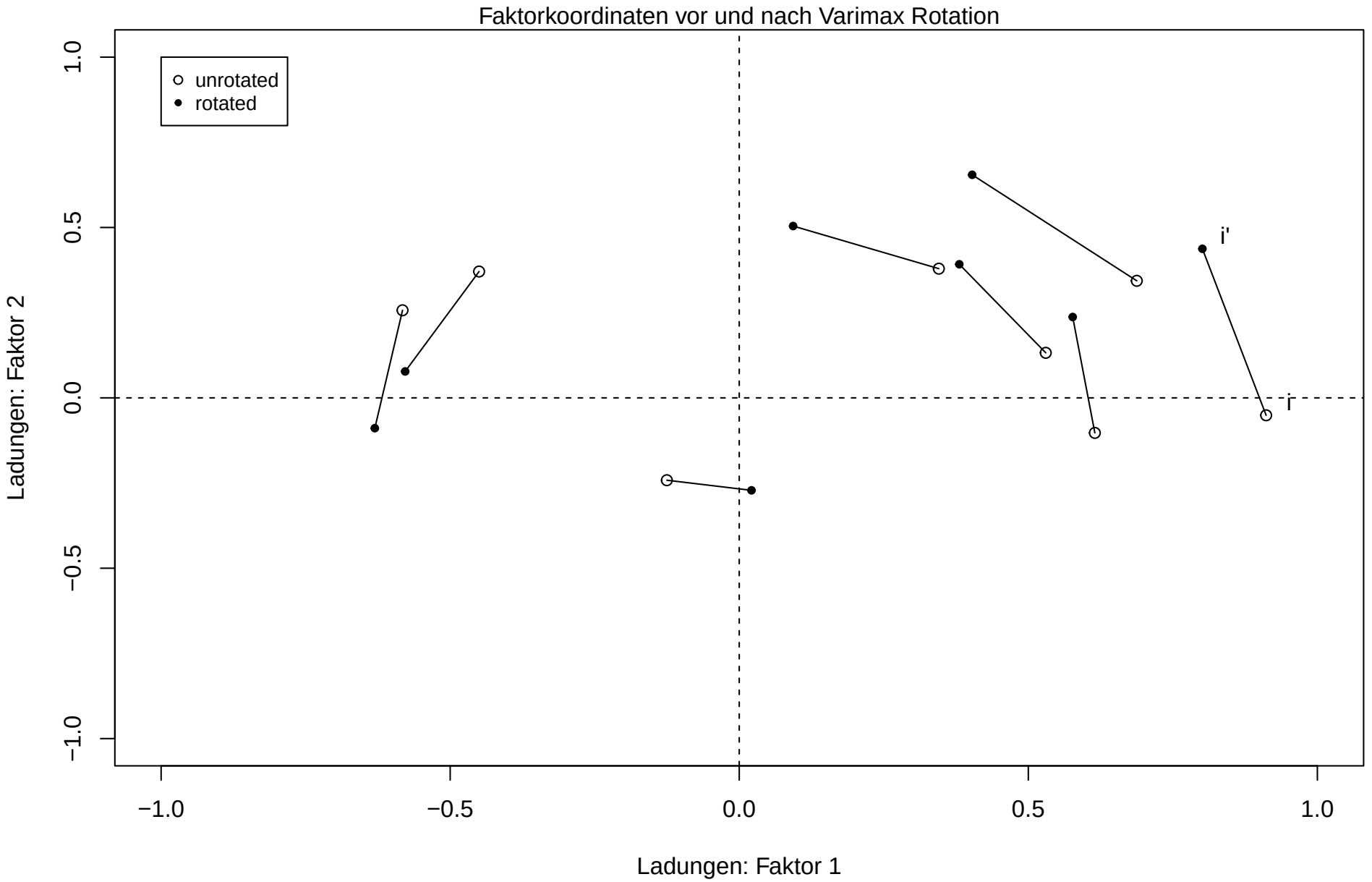
Rotationsproblem

Rotationsproblem

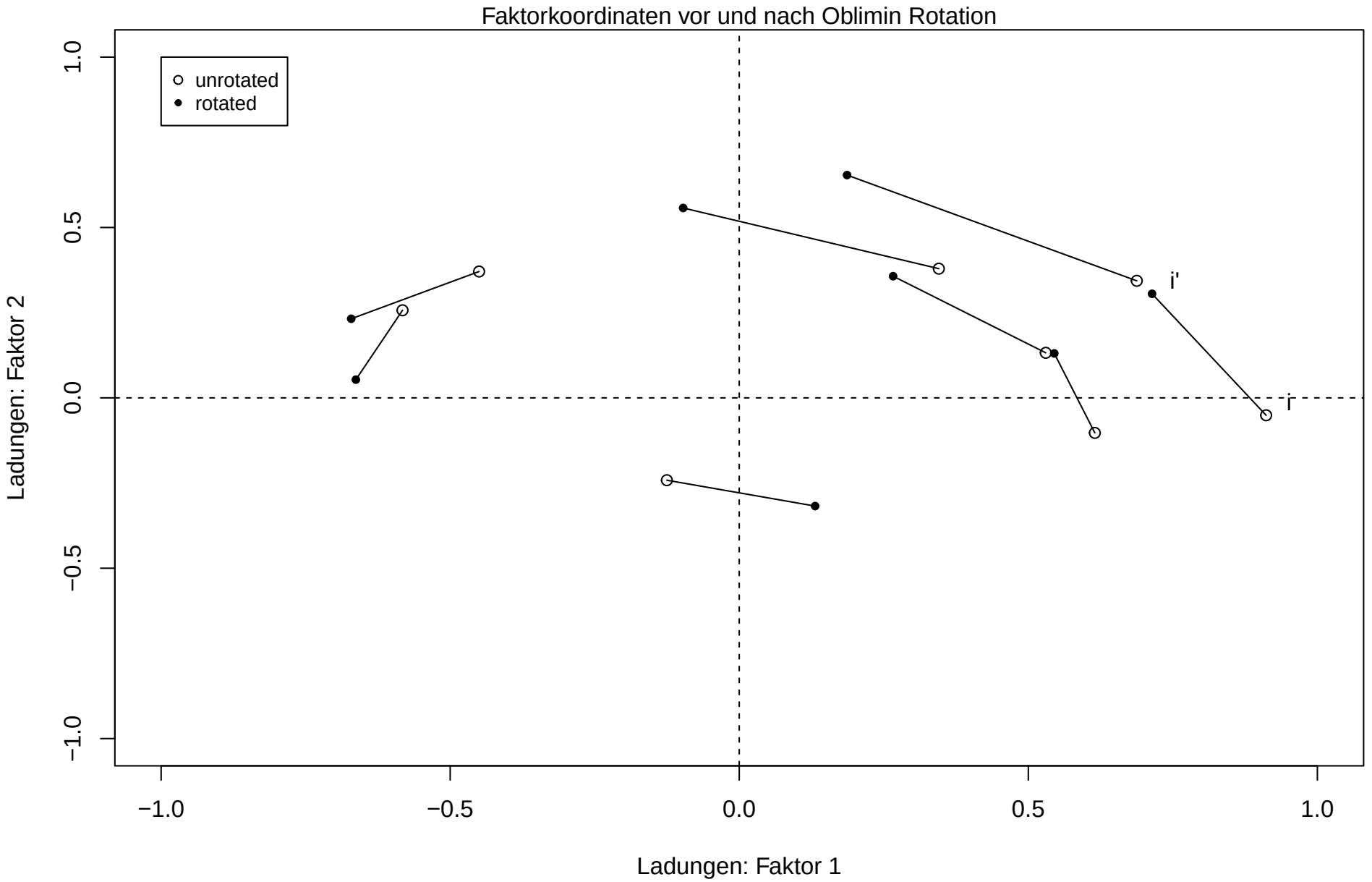
Wie rotiert die erhaltene Ladungsmatrix ($\hat{\Lambda}$) derart, dass eine möglich einfach interpretierbare Lösung daraus hervorgeht? (Einfachstruktur)

1. Lösung: orthogonale Rotation
2. Lösung: oblique Rotation

Graphik: Orthogonale Rotation mit Varimax



Graphik: Oblique Rotation mit Oblimin



Let's do it (..in R): orthogonale Rotation (Varimax)

```
(fit_vmax <- psych::fa(R, nfactors=2,
                       rotate="varimax",
                       fm="ml"))
# Kommunalitäten
fit_vmax$communality
# Eigenwerte
fit_vmax$e.values
# Einzigartigkeit
fit_vmax$uniquenesses
```

Let's do it (..in R): oblique Rotation (Oblimin)

```
(fit_obl <- psych::fa(R, nfactors=2,
                      rotate="oblimin",
                      fm="ml"))
# Kommunalitäten
fit_obl$communality
# Eigenwerte
fit_obl$e.values
# Einzigartigkeit
fit_obl$uniquenesses
# Wichtig: Phi
fit_obl$Phi
```

Übungsaufgabe 5: Selbstexperiment

Example

Versuchen Sie es nun selbst! Nutzen Sie den Code zur Übungsaufgabe 5 in `06-EFA.R` und führen Sie eine MLF durch. Rotieren Sie anschließend die Ergebnisse mit der Varimax und Oblimin Rotation. Wie verändert die Rotation Ihre Interpretation der Ergebnisse? Erleichtert sie sie? Vergleichen Sie die Lösung dazu (1) mit der unrotierten Lösung und vergleichen Sie (2) die Ergebnisse zwischend den Rotationsmethoden.

- Zeit: 10 Minuten
- Replikation: `set.seed(123)`
- Anmerkung: Konzepte & Output verstehen $\gg$ Codes verstehen!

Übungsaufgabe 6: Kritische Reflexion & Diskussion

Example

Eine Folge der orthogonalen Rotation ist, dass die extrahierten Faktoren unabhängig voneinander sind. Glauben Sie dem Modell uneingeschränkt, nähmen Sie damit implizit an, dass auch die zugrundeliegenden Konstrukten unabhängig voneinander sein müssten. Für wie plausibel halten Sie diese Annahme in der Psychologie? Diskutieren Sie miteinander!

- Zeit: 2-3 Minuten

Section 6

Problem der Anzahl zu extrahierender Faktoren

Problem der Anzahl zu extrahierender Faktoren

Rotationsproblem

Wie ermittle ich die Anzahl der zu extrahierenden Faktoren ohne sie zu kennen?

1. Lösung: Kaiser Kriterium
2. Lösung: Scree-Test
3. Lösung. Horns Parallel Analyse[4]

[4] Es gibt mittlerweile zahlreiche Weiterentwicklungen. Interessieren Sie sich dafür? Dann schauen Sie in Lorenzo-Seva et al. (2011) & Timmerman et al. (2018).

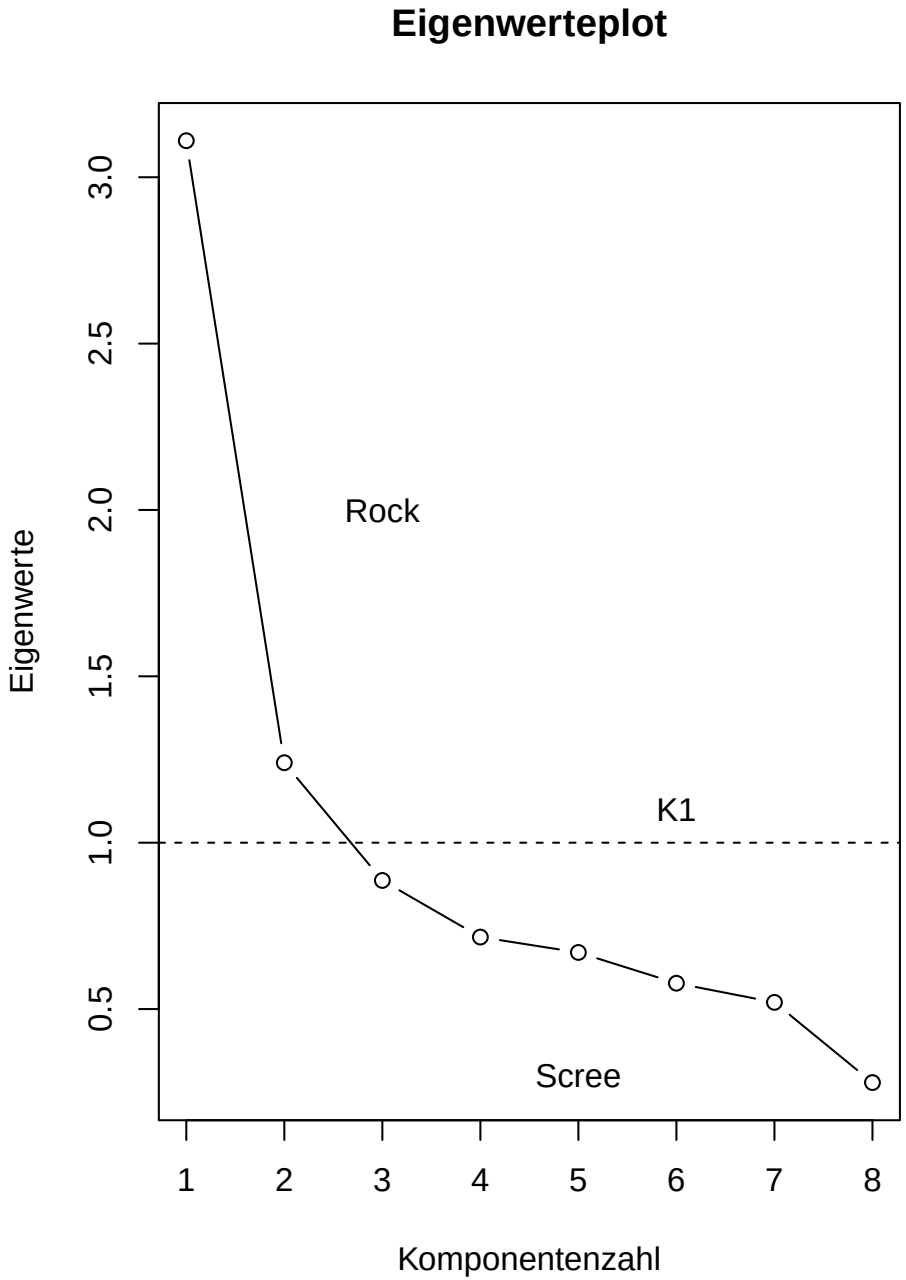
Eigenwerteplots (Kaiser Kriterium & Scree Test)

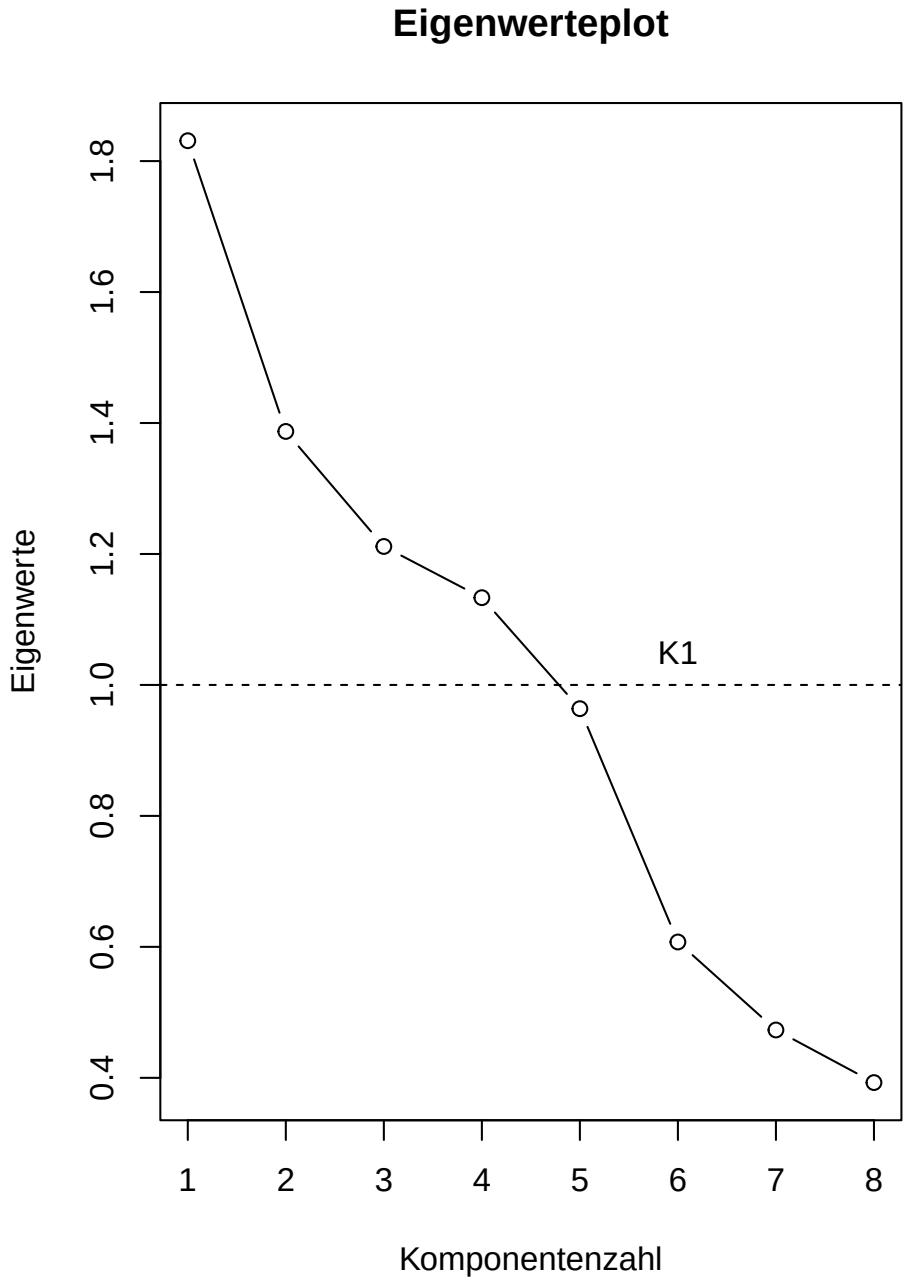
Kaiserkriterium & Scree Test

Zerlege die Korrelationsmatrix in ihre Eigenwerte (SVD/EVD) und plotte diese gegen einen Index.

1. Kaiserkriterium: Ist das Varianzaufklärungspotential eines Faktors kleiner als das eines Items, dann beende deine Suche ($\widehat{=} EW > 1$).
2. Scree-Test: Finde den "ellbow" der den "rock" vom "scree" unterscheidet.

Grafik: Eigenwertplot





Let's do it (..in R): Eigenwerteplots

```r
# Barfuß
 ev <- eigen(R)$values
 plot(ev, main="Eigenwerteplot", xlab="Eigenwerte",
      ylab = "Komponentenzahl", type="b")
 abline(h=1, lty=2)

# psych package
psych::scree(R, factors = FALSE)
# psych::scree(R, factors = TRUE)
```

Parallel Analyse

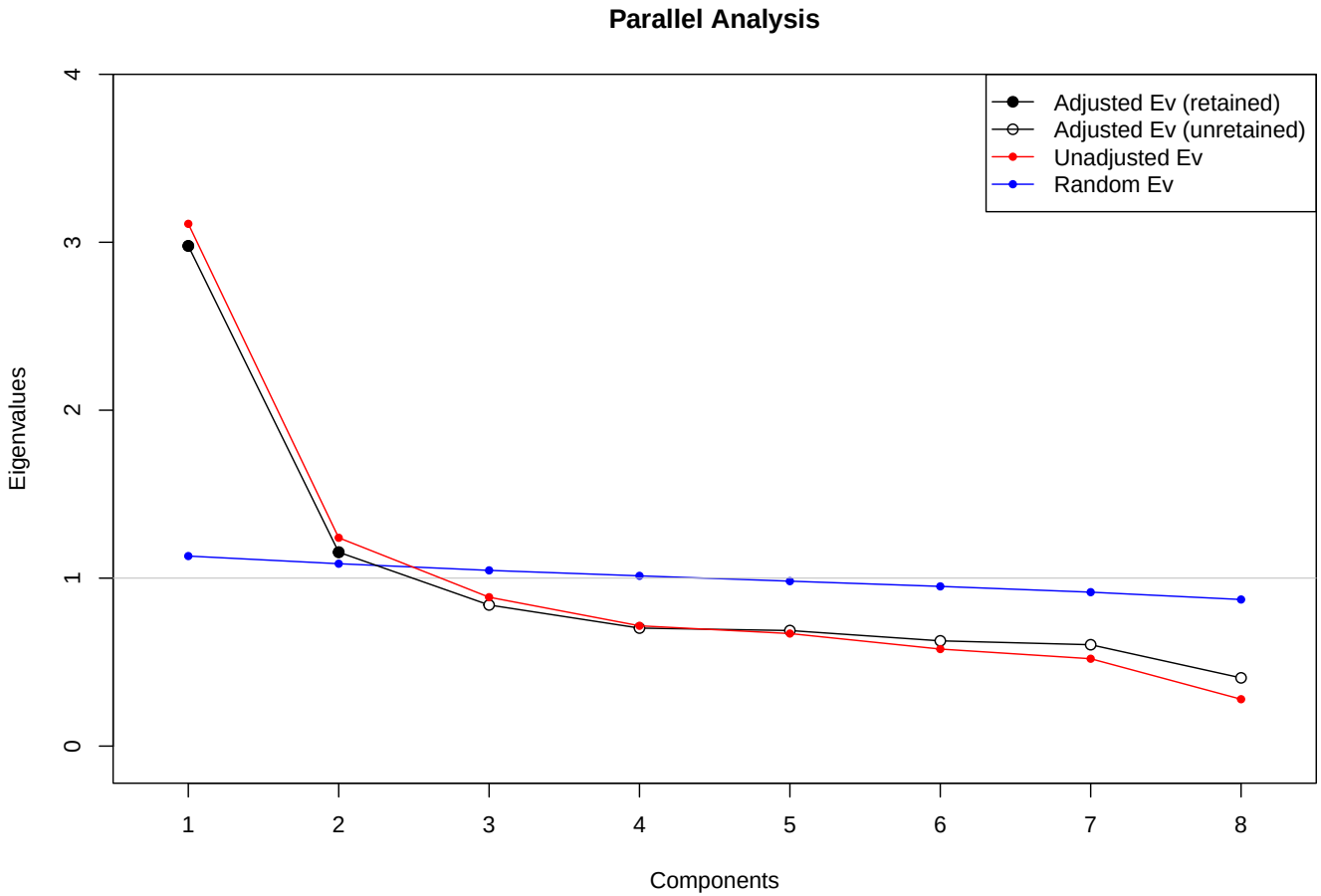
Parallel Analyse

"In parallel analysis, the criterion to be used in order to determine the number of factors is the following. A factor is considered as "significant" if its eigenvalue is larger than the 95% quantile [...] of those obtained from random or resampled data." (Mair 2018, S.31)

Anmerkung: Im Orginal wurde die PA für Komponenten konstruiert. Mittlerweile gibt es sie auch für Faktoren.

Grafik: Parallel Analyse (PCA)

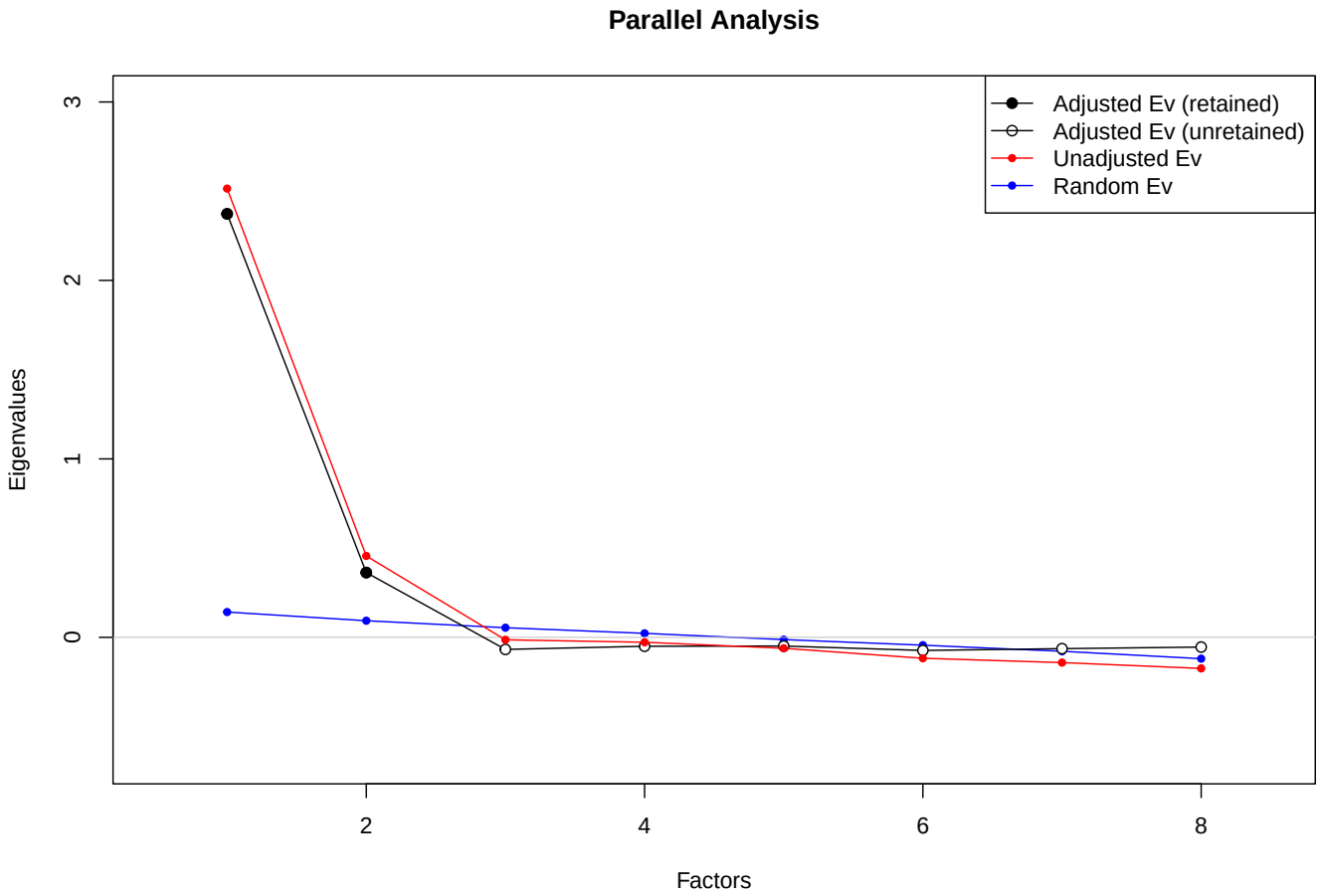
```
##
## Using eigendecomposition of correlation matrix.
```



Anmerkung: Die Grafik wurde mit `paran` nicht `psych` erstellt.

Grafik: Parallel Analyse (FA)

```
##
## Using eigendecomposition of correlation matrix.
```



Anmerkung: Die Grafik wurde mit `paran` nicht `psych` erstellt.

Let's do it (..in R): Parallel Analyse

```r
# Parallelanalyse (Komponenten-basiert)
psych::fa.parallel(X, fa = "pc", fm = "ml")

# Alternative
#
# Für Komponenten
# paran::paran(X)
# Für Faktoren
# paran::paran(X, cfa=TRUE)
```

Übungsaufgabe 7: Selbstexperiment

Example

Versuchen Sie es nun selbst! Nutzen Sie den Code zur Übungsaufgabe 7 in `06-EFA.R` und finden Sie die Anzahl zu extrahierender Faktoren in Ihrem Beispiel mit den drei genannten Möglichkeiten. Welches der drei Verfahren "errät" die Anzahl zu extrahierender Faktoren aus der Simulation (2 Faktoren)?

- Zeit: 10 Minuten
- Replikation: `set.seed(123)`
- Anmerkung: Konzepte & Output verstehen $\gg$ Codes verstehen!

Section 7

Bonusmaterial: Bartlett-Test & KMO

(Minimal-)Voraussetzungen: Faktorenanalyse

Minimalvoraussetzungen

Sind meine Daten, bzw. ist meien Korrelationsmatrix zur Analyse überhaupt geeignet?

1. Lösung: Bartlett-Test
2. Lösung: Kaiser-Meyer-Olkin Kriterium (KMO)

(Minimal-)Voraussetzungen: Bartlett-Test

Bartlett-Test

H_0: Die zu faktorisierende Korrelationsmatrix ist signifikant von einer Identitätsmatrix verschieden.

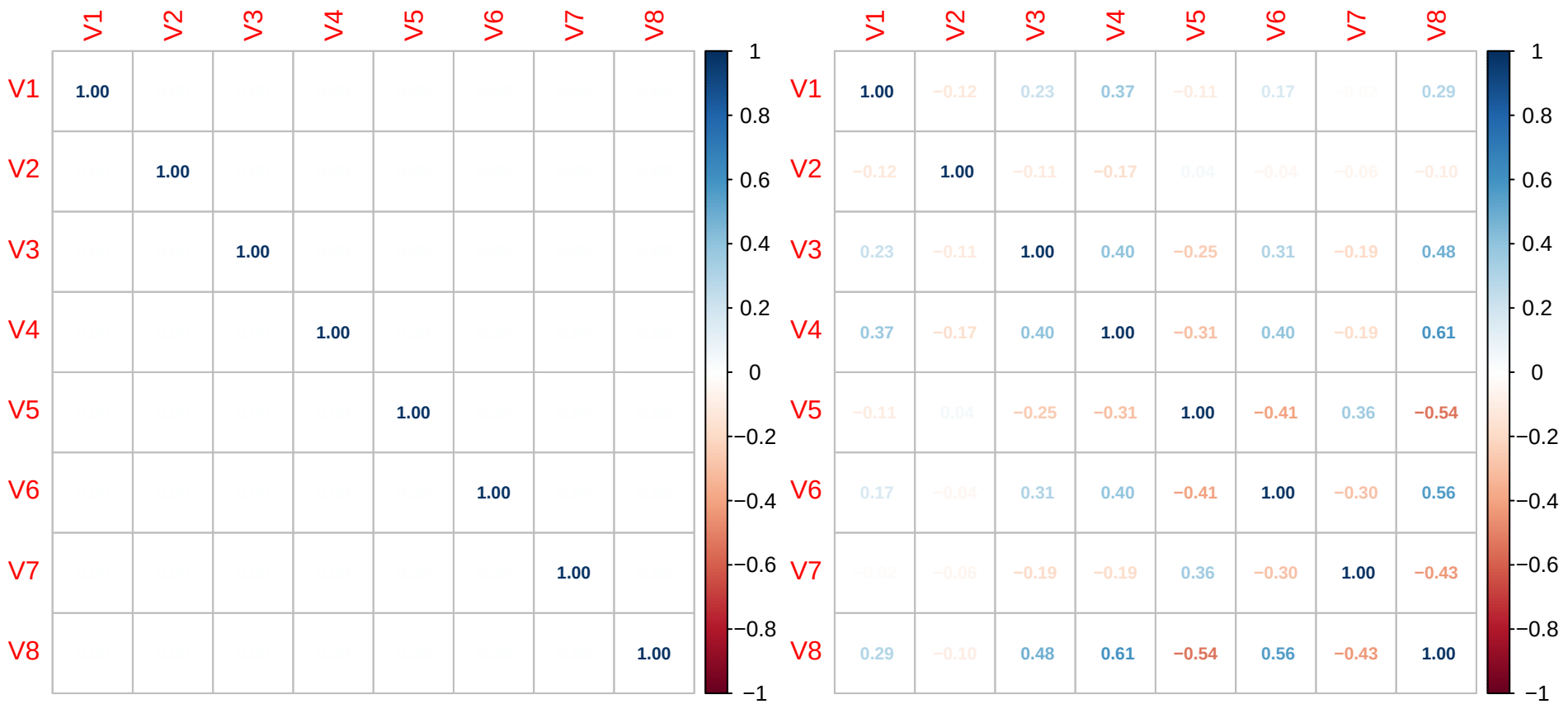
$$H_0 : \textit{verwefen} \Leftrightarrow R = I \tag{3}$$

Output: Identitätsmatrix

```
(I <- diag(8))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    1    0    0    0    0    0    0    0
## [2,]    0    1    0    0    0    0    0    0
## [3,]    0    0    1    0    0    0    0    0
## [4,]    0    0    0    1    0    0    0    0
## [5,]    0    0    0    0    1    0    0    0
## [6,]    0    0    0    0    0    1    0    0
## [7,]    0    0    0    0    0    0    1    0
## [8,]    0    0    0    0    0    0    0    1
```

Grafik: I vs. R


V1 V2 V3 V4 V5 V6 V7 V8
V1 1.00
V2 1.00
V3 1.00
V4 1.00
V5 1.00
V6 1.00
V7 1.00
V8 1.00
1 0.8 0.6 0.4 0.2 0 −0.2 −0.4 −0.6 −0.8 −1
V1 V2 V3 V4 V5 V6 V7 V8
V1 1.00 −0.12 0.23 0.37 −0.11 0.17 0.29
V2 −0.12 1.00 −0.11 −0.17 0.04 −0.04 −0.06 −0.10
V3 0.23 −0.11 1.00 0.40 −0.25 0.31 −0.19 0.48
V4 0.37 −0.17 0.40 1.00 −0.31 0.40 −0.19 0.61
V5 −0.11 0.04 −0.25 −0.31 1.00 −0.41 0.36 −0.54
V6 0.17 −0.04 0.31 0.40 −0.41 1.00 −0.30 0.56
V7 −0.06 −0.19 −0.19 0.36 −0.30 1.00 −0.43
V8 0.29 −0.10 0.48 0.61 −0.54 0.56 −0.43 1.00
1 0.8 0.6 0.4 0.2 0 −0.2 −0.4 −0.6 −0.8 −1

Let's do it (..in R): Bartlett-Test

```
# Datenmatrix (X)
# psych::cortest.bartlett(X)
# Korrelationsmatrix (R)
psych::cortest.bartlett(R, n=nrow(X))
```

```
## $chisq
## [1] 1974.157
##
## $p.value
## [1] 0
##
## $df
## [1] 28
```

(Minimal-) Voraussetzungen: KMO

KMO

Welchen Anteil an der Gesamtvarianz der Items ist auf gemeinsame Varianz rückführbar?

Anmerkung: Der gemeinsame Varianzanteil ist es letztlich, was die Faktoren aufgreifen können.

$$KMO = \frac{\sum\sum_{j \neq k} r_{jk}^2}{\sum\sum_{j \neq k} r_{jk}^2 + \sum\sum_{j \neq k} p_{jk}^2} \in [0, 1] \tag{4}$$

r_{jk} : Korrelation zwischen zwei Items p_{jk} : Partielle Korrelation

Let's do it! (..in R)

```
(kmo <- psych::KMO(X))
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: psych::KMO(r = X)
## Overall MSA =  0.81
## MSA for each item =
##   V1   V2   V3   V4   V5   V6   V7   V8
## 0.81 0.74 0.88 0.80 0.84 0.88 0.80 0.76
```

```
# kmo$MSA
```

Übungsaufgabe 8: Selbstexperiment

Example

Versuchen Sie es nun selbst! Nutzen Sie den Code zur Übungsaufgabe 7 in `06-EFA.R` und führen Sie einen Bartlett-Test durch. Verwenden Sie auch das KMO-Kriterium auf ihre Korrelations- oder Datenmatrix an. Wie beurteilen Sie die Ergebnisse? Sind ihre Matritzen zur Analyse geeignet?

- Zeit: 10 Minuten
- Replikation: `set.seed(123)`
- Anmerkung: Konzepte & Output verstehen $\gg$ Codes verstehen!

Übungsaufgabe 9: Evaluation

Example

Welches Verfahren hat ihre anfängliche Struktur am besten aufgefunden? Haben Sie eine Idee, warum? (Tipp: Denken Sie an Φ)

Anmerkung: Wiederholen Sie die Übung im Selbststudium mit anderen Werten. Verändern Sie diese systematisch und prognostizieren Sie, was geschehen wird und warum. So lernen Sie jedes Verfahren, jeden Algorithmus und jedes Kriterium verstehen.

Section 8

Epilogue: The Power of Simulation!

Epilogue: The Power of Simulation!

. . . Wenn Sie wissen wollen, wie sich ein bestimmtes Kriterium unter spezifischen Voraussetzungen verhält, dann simulieren Sie!

The Power of Simulation

Sie besitzen nun die Power der Simulation! Simulieren Sie sich bestimmte Strukturen (wie in diesem Kurs getan) und testen Sie die Kriterien, Kennwerte, usw. auf Herz und Nieren! Nur so lernen Sie ihre Grenzen kennen.

Anmerkung: Gute Klausurübung! Schnappen Sie sich ihre Mitstudierenden, simulieren Sie mögliche Welten und üben, üben, üben Sie!

Section 9

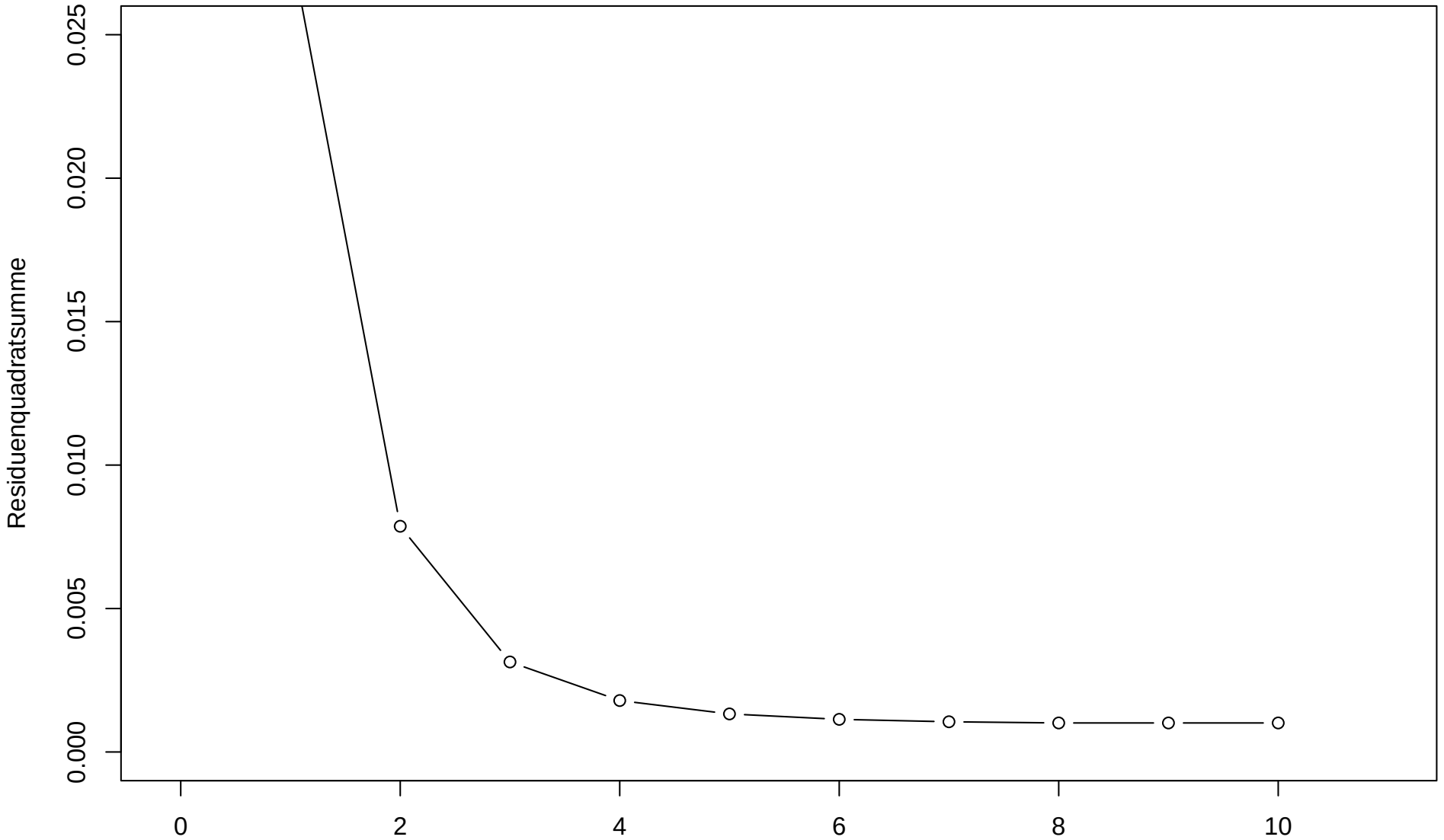
Selbststudium

Selbststudium 1: die Relevanz von Iterationen

It is a useful exercise to run fa using the principal axis factor method (fm= "pa") and specifying the number of iterations (e.g., max.iter=2). Then examine the size of the residuals as the number of iterations increases. When this is done, the solution gets progressively better, in that the size of the residuals in the off diagonal matrix become progressively smaller. (Revelle, in prep., S. 152)

Selbststudium 1: die Relevanz von Iterationen

```
##  [1] 0.028108682 0.007867171 0.003137076 0.001793045 0.0013
##  [7] 0.001052723 0.001011982 0.001011982 0.001011982
```



Selbststudium 2: händische PCA – Eigenwertzerlegung

```
# Black Box
fit_pca <- princomp(X, cor = TRUE)

# Barfuß
R_EVD <- cor(X)
# EVD von R ; Eigenvektoren ; Eigenwerte
EVD_R <- eigen(R_EVD)
# Komponentenladungen
loadings <- EVD_R$vectors
# Gleich?
fit_pca$loadings ; print(loadings, digits=3)
```

Selbststudium 3: händische EFA – Singulärwertzerlegung

```r
# Black Box
fit_pca <- prcomp(X, scale. = TRUE)

# Barfuß
n <- nrow(X)
SVD_X <- svd(scale(X)/sqrt(n-1))
U <- SVD_X$u ; D <- SVD_X$d
loadings <- U %*% diag(D) * sqrt(n-1)
# Gleich?
fit_pca$x ; loadings
```

Selbststudium 4: reduzierte Korrelationsmatrix

. . . Der Output zeigt die erst 8 Iterationen einer PAF. Vergleichen Sie diese mit ihre am Anfang konstruierte Korrelationsmatrix $\mathtt{R}$). Achten Sie auf die Hauptdiagonalen!

Selbststudium 5: PAF

[P]rincipal axes factor analysis. This is similar to principal components, except that it is done with a reduced matrix where the diagonals are the communalities. The communalities can either be (1) specified a priori, (2) estimated by such procedures as multiple linear regression, or (3) found by iteratively doing an eigenvalue decomposition and repeatedly replacing the original 1s on the diagonal with the value of $1 - u^2$ *where U^2 = diag(R-FF'). (Revelle, in prep., S.156)*

Selbststudium 5: händische PAF (Basisalgorithmus, 1 Iteration)

```
# EVD of R
EVD_R <- eigen(R)
# Loadings
F <- EVD_R$vectors
# Residual matrix
R_ast <- R - F %*% t(F)
# Squared uniqness matrix
U2 <- diag(R_ast)
# Replace on diagonals
(diag(R) <- 1 - U2)
R
```

Selbststudium 5: händische PAF (1 Iteration)

```
EVD_R <- eigen(R) # EVD of R
F <- EVD_R$vectors # Loadings
k <- 5 # k largest PC
# Pre reproduction
Fk <- F[,seq(k)]
Z <- matrix(0, ncol = ncol(F) - k,  nrow(F))
Fk <- cbind(Fk, Z)
# Residual matrix
R_ast <- R - Fk %*% t(Fk)
# Squared uniqness matrix
U2 <- diag(R_ast)
# Replace on diagonals
diag(R) <- 1 - U2
#...Repeat
round(R, digits = 3)
```

Selbststudium 5: händische PAF ($\geq$ 1 Iteration)

Anmerkung: Führen Sie das Code Snippet solange immer wieder aus, bis sich die Verändeurngen in der Hauptdiagonale stabilisieren. Jede Wiederholung des Codes stellt eine Iteration in der Maschinerie der PAF dar. Nichts anderes passiert innerhalb des Algorithmuses den Sie in konventionellen Packages finden. ..nur das die Algorithmen deutlich performanter und effizienter implementiert sind.

Literaturverzeichnis I

Dinno, Alexis. 2018. *Paran: Horn's Test of Principal Components/Factors*. http://alexisdinno.com/Software/files/PA_for_PCA_vs_FA.pdf.

Francois, Romain. 2020. *Bibtex: Bibtex Parser*. https://github.com/romainfrancois/bibtex.

Lorenzo-Seva, Urbano, Marieke E. Timmerman, and Henk A. L. Kiers. 2011. "The Hull Method for Selecting the Number of Common Factors" 46 (2): 340–64. https://doi.org/10.1080/00273171.2011.564527.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Revelle, William. 2021. *Psych: Procedures for Psychological, Psychometric, and Personality Research*. https://personality-project.org/r/psych/%0Ahttps://personality-project.org/r/psych-manual.pdf.

Literaturverzeichnis II

Timmerman, Marieke E., Urbano Lorenzo-Seva, and Eva Ceulemans. 2018. "The Number of Factors Problem." In, 305–24. John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118489772.ch11.

Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown-cookbook.