Explorative Faktorenanalyse

Jalynskij et al.

Einstieg

https://openpsychometrics.org/tests/IPIP-BFFM/1.php

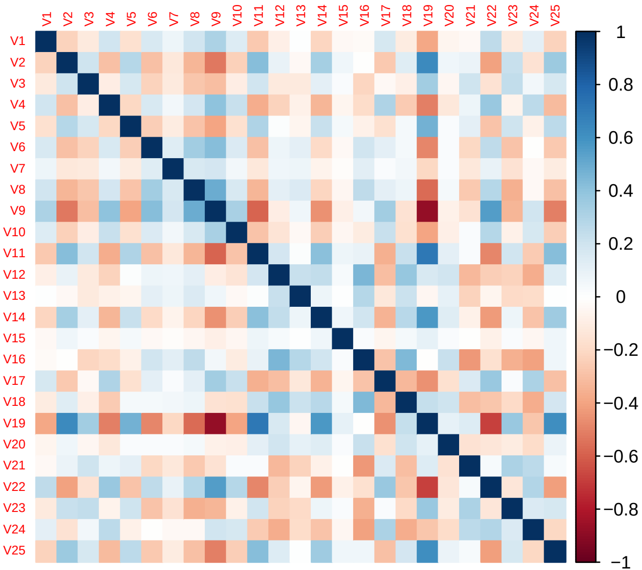
Die “Large Data Set Challenge”

Example

Stellen Sie sich vor, die von Ihnen soeben beantworteten Fragen ergäben die Korrelationsmatrix R auf der nächsten Folie. Die "Large Data Set Challenge" lautet: Erkennen Sie eine Struktur in den Daten? D.h., wenn ja weiter; welche Items könnten man Ihrer Meinung nach zu Itemgruppen zusammenfassen?

Anmerkung: Nein, das sind (wirklich) nicht ihre Antworten; `V1 - V30` sind Zufallsvariablen!

Übungsaufgabe 1: Struktur erkennen & Itemgruppen finden



Die “Large Data Set Challenge”

Large Data Set Challenge

Mit zunehmender Itemzahl nimmt die wird die Anzahl der Korrelationen, die für eine Analyse zu berücksichtigen sind schnell zu. Die "Challenge" ist eine mögliche Struktur zu erkennen!

In a Nutshell:

- Problem: Anzahl der Korrelationen
- z.B.: 25 Items $\widehat{=}$ $2^{25} = 625$ Korrelationen
- Krux: Struktur erkennen
- $\Leftrightarrow$ finde: hoch korrelierende Itemgruppen

Explorative Faktorenanalyse

- (ein) Hilfsmittel: .. (explorative) *Faktorenanalyse*

Faktorenanalyse

"The basic idea is to find latent variables (factors) based on the correlation structure of the manifest input variables (indicators)." (Mair 2018, S. 23)

- andere Helferlein zur *Datenreduktion* (eine Auswahl):
  - Hauptkomponentenanalyse
  - Clusteranalyse
  - Explorative Likertskalierung
  - (Non-) Metric Data Scaling (Voraus.: Distanzmatrizen)

Wichtig: “meaningful compression” (Faktotren) vs. “full compression” (Komponenten)

Stategie & Vorgehen: Simulation & Evaluation

1. Man erschaffe ≥ 1 eine latente Variable (LV)
2. ... lasse die LV Antwortmuster produzieren
3. ... wandel sie in eine Korrelationsmatrix (R) um
4. ... und versucht die Struktur mit der Faktorenanalyse aufzufinden

Vom generativen Prozess zur Korrelationsmatrix

Der generative Prozess, d.h. wie genau ein Konstrukt die Antworten auf den Items erzeugt, bleibt meist verborgen. Wir untersuchen meistens lediglich Verhaltensspuren des Konstruktes, die sich in den Items ausdrückt, d.h. in der Struktur der Korrelationsmatrix niederschlägt. Strukturen zu simulieren ist hilfreich, weil wir dort "die Wahrheit" kennen und das Verfahren damit besser beurteilen können ($\sim$ fake data analysis)

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

"Playing Creator": zwei latente Variable erschaffen

```
# 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$model ; X <- S$observed  # R <- cor(X)
```

Output (grafisch): zwei latente Variablen

Übungsaufgabe 2: Selbstexperiment

Example

Versuchen Sie es selbst! Verändern sie systematisch `F1; F2` und `phi`. Wie verändert sich die Korrelationsmatrix, in Abhängigkeiten Ihrer Veränderungen? Können Sie eine eindeutige Struktur konstruieren? Wenn ja, mit welchen Werten von `F1; F2` und `phi` haben Sie ihr Ziel erreicht?

Zusatz: Haben Sie ein Strukturmodell gefunden, dass ihnen gefällt? Ja, dann überlegen Sie sich jetzt für welche Konstrukte diese Struktur Sinn macht (z.B.: extraversion $\sim$ openness to experience)

Logik latenter Variablen (..reversed)

Von der Korrelationsmatrix zur latenten Variable

Die Faktoranalyse ist ein strukturentdeckendes Verfahren. D.h. den generativen Prozess, d.h. wie ein Konstrukt die Antworten auf den Items verursacht hat anhand der Struktur die sich in der vorgegebenen Korrelationsmatrix zu *modellieren*.

- Modell: *Common Factor Model* (CFM)

Eingangsgleichung

$$x = \Lambda\xi + \epsilon \quad (1)$$

"In other words EFA tries to find p latent variables on the basis of the correlation structure of the m manifest variables." (ebd.)

Das Common Factor Model (CFM)

Anmerkung: Für die Reformulierung von Gleichung (1) zu (2): siehe McCallum (2009)

Fudnamentaltheorem

$$P = \Lambda \Phi \Lambda^t + \Psi \tag{2}$$

- P: Modell-implizierte Korrelationsmatrix
- Λ: Ladungsmatrix
- Φ: Matrix der Zwischenfaktorkorrelationen
- Ψ: Uniqueness

Zusammenhang: Modell & Struktur

Die von ihnen konstruierte Struktur versuchen wir nun mit der Faktorenanalyse unter Einsatz des CFM zu rekonstruieren. Das CFM ist also Ihr Tool im bevorstehenden Rekonstruktionsprozess!

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)*

Konklusion: Versuchen Sie ihr Modell zu verstehen! (siehe: Selbststudium 1-3)

Extraktionsproblem

Extraktionsproblem

Wie extrahieren wir die Faktoren/Komponenten

1. Lösung: Bestimmung der "Principal Components"
   - Verfahren: PCA (Principal Component Analysis/Method)
   - Modellgleichung: $R \leftarrow P = CC^t$
   - Note: Eigenwert-, Singulärwertzerlegung (closed form solution)
2. Lösung: Iterative Bestimmung der "Principal Components"
   - Verfahren: PAFA (Principal Axis Factor Analysis)
   - Modellgleichung: $R^* \leftarrow P = FF^t$
   - Note(s): Reduzierte Matrix, iterativer Prozess (convergence issues)
3. Lösung: Finde die plausibelsten ("most likely") Werte zur Repro
   - Verfahren: MLFA (Maximum Likelihood Factor Analysis)
   - Modellgleichung: $R \leftarrow P = \Lambda\Phi\Lambda^t + \Psi$
   - Note(s): Reduzierte Matrix, itterativer Prozess (convergence issues)

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

Hauptkomponentenanalyse

```
# Old School!
# (dino_pca <- princomp(X, cor=TRUE))
# New School
(pca_fit <- psych::principal(R, nfactors = 2,
                              rotate = "none"))
## Komponentenladungen
pca_fit$loadings
## Kommunalitäten
pca_fit$communality
## Einzigartigkeit
pca_fit$uniquenesses
## Eigenwerte
pca_fit$values
# Quadrierte multiple Korrelation
pca_fit$R2
```

Übungsaufgabe 3: Selbstexperiment

Example

Versuchen Sie es nun selbst! Fitten sie ein PC model. Interpretieren Sie die enstprechenden Kennwerte für ihr Model und präsentieren Sie diese mir oder ihrem Nachbarn. Verändern Sie auch einmal die Anzahl der zu extrahierenden Faktoren (`nfactors`). Wie verändert sich ihre Lösung wenn sie die Zahl vergößern, bzw. verkleinern? Wie wirken sich diese Veränderungen auf die Interpretation Ihrer Ergebnisse aus?

Anmerkung: Denken Sie daran, 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$uniquenesses
# Quadrierte multiple Korrelation
fit_paf$R2
```

Übungsaufgabe 4: Selbstexperiment

Example

Versuchen Sie es nun selbst! Führen Sie eine Hauptachsenanalyse (PAF) durch. Vergleichen Sie die Ergebnisse der PAF mit denen der PCA. Bestehen Unterschiede zwischen den Ergebnissen?

Zusatz: Haben Sie eine Vermutung wie mögliche Unterschiede zustande kommen?

Übungsaufgabe 5: Kritische Reflexion & Diskussion

Example

Eine Folge der Extraktion mittels mit der Hauptkomponentenmethode ist, dass die extrahierten Faktoren unabhängig voneinander sind. Glauben Sie dem Modell uneingeschränkt, nähmen sie damit implizit an, dass auch die zugrundeliegende Konstrukte unabhängig voneinander sein müssten. Denken Sie an ihr Beispiel. Für wie plausibel halten Sie diese Annahme? Diskutieren Sie (heftig)!

Warnhinweis: Sollte ein Inferno entfachen, halten Sie bitte Fluchtwege sowie Zugänge zu den Feuerlöschern frei!

Maximum Likelihood Faktorenanalyse

Kommunalitätenproblem

Das CFM ($\Lambda\Phi\Lambda' + \Psi$) ist unbestimmt. D.h. es hat (deutlich) mehr unbekannte als bekannte Bestanndteile. Die Folge: Das Modell kann deshalb nicht einfach "gelöst" werden, sondern muss geschätzt werden.

- MLFA bietet nun als Lösung die plausibelsten (eng. "most likely") Werte zur Reproduktion der Informationen in der Korrelationsmatrix an. *"Assuming that the residual variance reflects normally distributed random error, the most elegant statistical solution is that of maximum likelihood" (Revelle, in prep. S. 156)*

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

```r
(fit_mlf <- psych::fa(R, nfactors=2, rotate="none", fm="ml"))
# Kommunalitäten
fit_mlf$communality
# Eigenwerte
fit_mlf$e.values
# Einzigartigkeit
fit_mlf$uniquenesses
# Quadrierte multiple Korrelation
fit_mlf$R2
```

Übungsaufgabe 6: Selbstexperiment

Example

Versuchen Sie es nun selbst! Führen Sie eine Maximum Likelihood Faktorenanalyse (MLF) durch. Vergleichen Sie die Ergebnisse mit den anderen Methoden und prüfen Sie die Validität der Ergebnisse in Bezug auf Ihr Beispiel.

Zusatz: Haben Sie eine Vermutung wie mögliche Unterschiede zwischen den Verfahren zustande kommen?

Rotationsproblem

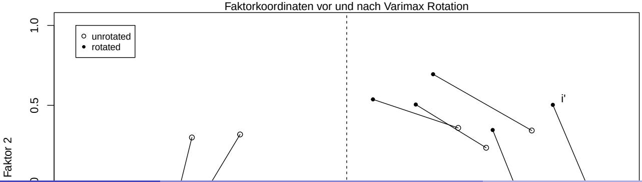
Rotationsproblem

Wie rotiert die erhaltene Ladungsmatrix (Λ) derart, dass eine möglich einfach interpretierbare Lösung daraus hervorgeht?

1. Lösung: Orthogonale Rotation (z.B. Varimax)
    - Ziel: Orthogonalität aufrechterhalten
    - d.h. Faktoren dürfen nicht korrelieren
    - z.B.: Varimax, Promax..
2. Lösung: Oblique Rotation (z.B. Oblimin)
    - Ziel: mögliche Orthogonalität auflösen
    - d.h. Faktoren dürfen korrelieren
    - z.B.: Oblimin, Simplimax..

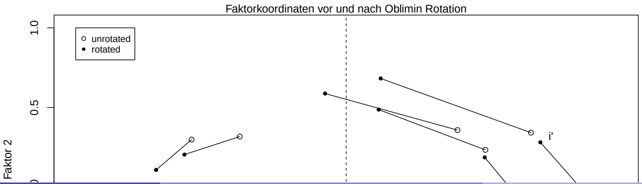
Graphik: Orthogonale Rotation mit Varimax

```
fit_mlf_vmax <- psych::fa(R, nfactors=2, rotate="varimax", fm=
plot(fit_mlf$loadings, ylim = c(-1,1), xlim = c(-1,1),
    xlab = "Ladungen: Faktor 1" , ylab = "Ladungen: Faktor 2")
points(fit_mlf_vmax$loadings, pch=20)
x1 <- fit_mlf$loadings[,1] ; y1 <- fit_mlf$loadings[,2]
x2 <- fit_mlf_vmax$loadings[,1] ; y2 <- fit_mlf_vmax$loadings[
for(i in seq(8)) lines(c(x1[i], x2[i]), c(y1[i], y2[i]))
abline(h = 0, lty=2) ; abline(v = 0, lty=2)
mtext("Faktorkoordinaten vor und nach Varimax Rotation")
i <- 8 ; s <- 0.04
text(c(x1[i]+s, x2[i]+s), c(y1[i]+s, y2[i]+s), c("i", "i'"))
legend(x = -1, 1, legend=c("unrotated", "rotated"), pch = c(1,
```



Graphik: Oblique Rotation mit Oblimin

```
fit_mlf_obl <- psych::fa(R, nfactors=2, rotate="oblimin", fm="
plot(fit_mlf$loadings, ylim = c(-1,1), xlim = c(-1,1),
    xlab = "Ladungen: Faktor 1" , ylab = "Ladungen: Faktor 2"
points(fit_mlf_obl$loadings, pch=20)
x1 <- fit_mlf$loadings[,1] ; y1 <- fit_mlf$loadings[,2]
x2 <- fit_mlf_obl$loadings[,1] ; y2 <- fit_mlf_obl$loadings[,
for(i in seq(8)) lines(c(x1[i], x2[i]), c(y1[i], y2[i]))
abline(h = 0, lty=2) ; abline(v = 0, lty=2)
mtext("Faktorkoordinaten vor und nach Oblimin Rotation")
i <- 8 ; s <- 0.04
text(c(x1[i]+s, x2[i]+s), c(y1[i]+s, y2[i]+s), c("i", "i'"))
legend(x = -1, 1, legend=c("unrotated", "rotated"), pch = c(1,
```



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$uniquenesses
# Quadrierte multiple Korrelation
fit_vmax$R2
```

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$uniquenesses
# Quadrierte multiple Korrelation
fit_obl$R2
```

Übungsaufgabe 7: Selbstexperiment

Zusatz: Sie kennen das “richtige” Ergebnis – Sie haben die Antworten immerhin simuliert! Welche Kombination kommt dem “richtigen” Ergebnis am nähsten? Haben Sie eine Idee warum?

Problem der Anzahl zu extrahierender Faktoren

Rotationsproblem

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

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

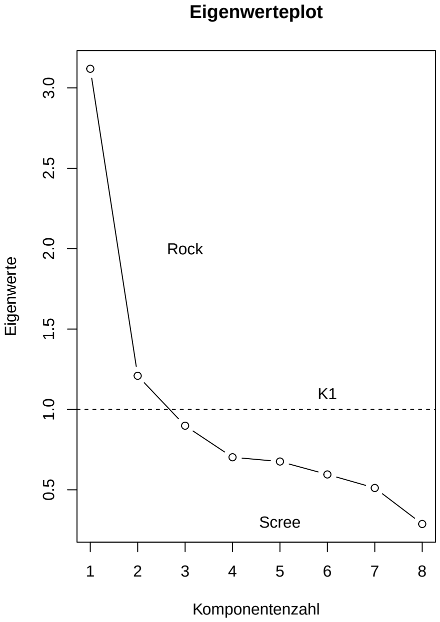
Anmerkung: Es gibt mittlerweile zahlreiche Weiterentwicklungen. Interessieren Sie sich dafür? (siehe: Lorenzo-Seva et al., 2011 & Timmerman et al., 2018)

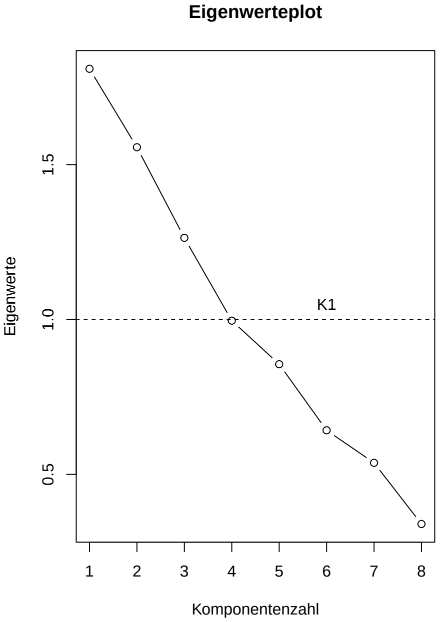
Eigenwerteplots (Kaiser Kriterium & Scree Test)

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 eines Items beende deine Suche ($\widehat{=} EW > 1$). (2) Scree-Test: Finde den "ellbow" der den "rock" vom "scree" unterscheidet.

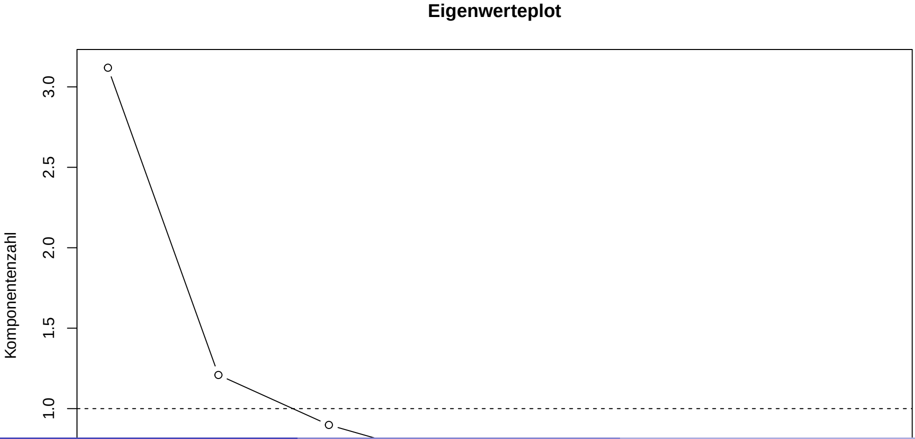
Grafiks: Eigenwertplot





Let's do it (..in R)

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



Parallel Analyse

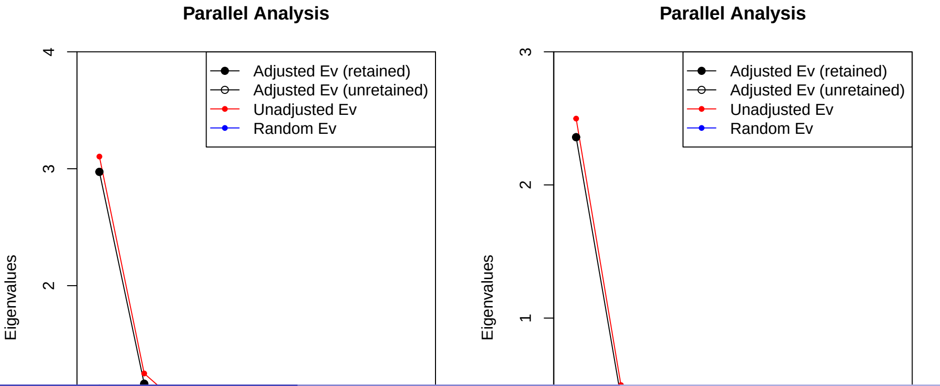
Parallel Analyse

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 95random or resampled data. (Mair 2018, S.31)

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

Grafik: Parallel Analyse

```
##
## Using eigendecomposition of correlation matrix.
## Computing: 10%  20%  30%  40%  50%  60%  70%  80%  90%  100
##
## Using eigendecomposition of correlation matrix.
## Computing: 10%  20%  30%  40%  50%  60%  70%  80%  90%  100
```



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

```
# Run time issue!
# psych::fa.parallel(X, fa = "pc", fm = "ml")

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

Übungsaufgabe 8: Selbstexperiment

Example

Versuchen Sie es nun selbst! Finden Sie die Anzahl zu extrahierender Faktoren in ihrem Beispiel mit den drei genannten Möglichkeiten. Welches der drei Verfahren "errät" ihre Anzahl zu extrahierender Faktoren richtig? Welches nicht? Haben Sie eine Idee warum? Vergleichen Sie auch die Ergebnisse mit ihrem Nachbarn oder den Folien. Warum sind die Ergebnisse anders? Warum vielleicht auch nicht ? (Tipp: vergleichen Sie v.a. Λ, Φ)

Anmerkung: Sie haben den Luxus, das richtige Ergebnis zu kennen. Normalerweise ist das nicht der Fall! Darum sind Simulationen wichtig. So können wir das Verhalten einzelner Kriterien unter verschiedenen Umständen testen. Versuchen Sie es daheim mal mit anderen Werten (`N`, `F1`, `F2` & `phi`)!

(Minimal-)Voraussetzungen: Bartlett-Test

Bartlett-Test

Ist die zu faktorisierende Korrelationsmatrix signifikant von einer Identitätsmatrix verschieden? Eine Minimalvoraussetzung für Faktorisierung.

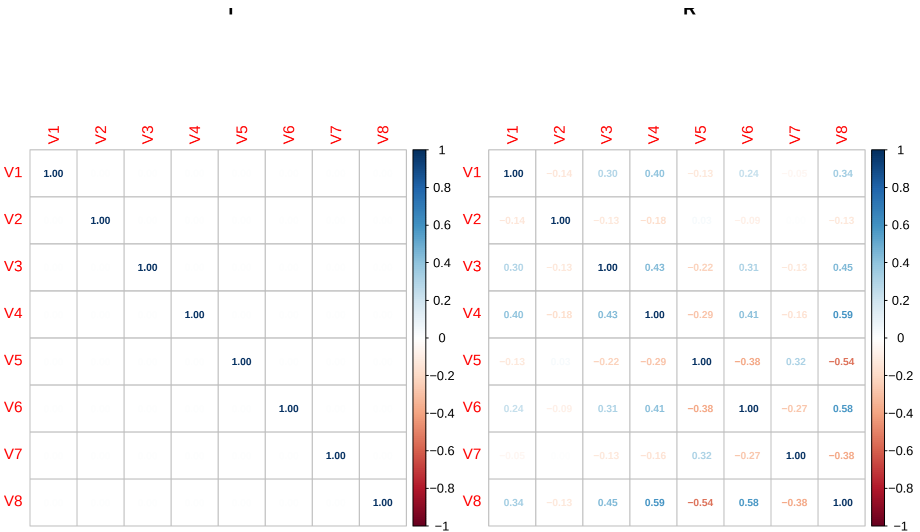
$$H_0 : verwefen \; \Leftrightarrow R = I \tag{3}$$

Output: Identitätsmatrix

```
(I <- diag(10))
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    1    0    0    0    0    0    0    0    0     0
##  [2,]    0    1    0    0    0    0    0    0    0     0
##  [3,]    0    0    1    0    0    0    0    0    0     0
##  [4,]    0    0    0    1    0    0    0    0    0     0
##  [5,]    0    0    0    0    1    0    0    0    0     0
##  [6,]    0    0    0    0    0    1    0    0    0     0
##  [7,]    0    0    0    0    0    0    1    0    0     0
##  [8,]    0    0    0    0    0    0    0    1    0     0
##  [9,]    0    0    0    0    0    0    0    0    1     0
## [10,]    0    0    0    0    0    0    0    0    0     1
```

Grafik: I vs. R


I
R
V1 V2 V3 V4 V5 V6 V7 V8
1.00
1.00
1.00
1.00
1.00
1.00
1.00
1.00
1 0.8 0.6 0.4 0.2 0 −0.2 −0.4 −0.6 −0.8 −1
V1 1.00 −0.14 0.30 0.40 −0.13 0.24 −0.05 0.34
V2 −0.14 1.00 −0.13 −0.18 0.03 −0.09 −0.13
V3 0.30 −0.13 1.00 0.43 −0.22 0.31 −0.13 0.45
V4 0.40 −0.18 0.43 1.00 −0.29 0.41 −0.16 0.59
V5 −0.13 0.03 −0.22 −0.29 1.00 −0.38 0.32 −0.54
V6 0.24 −0.09 0.31 0.41 −0.38 1.00 −0.27 0.58
V7 −0.05 −0.13 −0.16 0.32 −0.27 1.00 −0.38
V8 0.34 −0.13 0.45 0.59 −0.54 0.58 −0.38 1.00

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

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

```
## $chisq
## [1] 1950.464
##
## $p.value
## [1] 0
##
## $df
## [1] 28
```

(Minimal-) Voraussetzungen: Kaiser-Meyer-Olkin Kriterium

\begin{block}{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.

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

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

Übungsaufgabe 9: Selbstexperiment

Example

Versuchen Sie es nun selbst! Führen Sie einen Bartlett-Test durch und wendenen Sie das KMO-Kriterium auf ihre Korrelations- oder Datenmatrix an. Wie beurteilen Sie die Ergebnisse? Sind ihre Matritzen zur analysie geeignet?

Übungsaufgabe 10: Evaluation

Example

Welches Verfahren hat ihre anfängliche Struktur am besten aufgefunden? Haben Sie eine Idee warum?

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.

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.

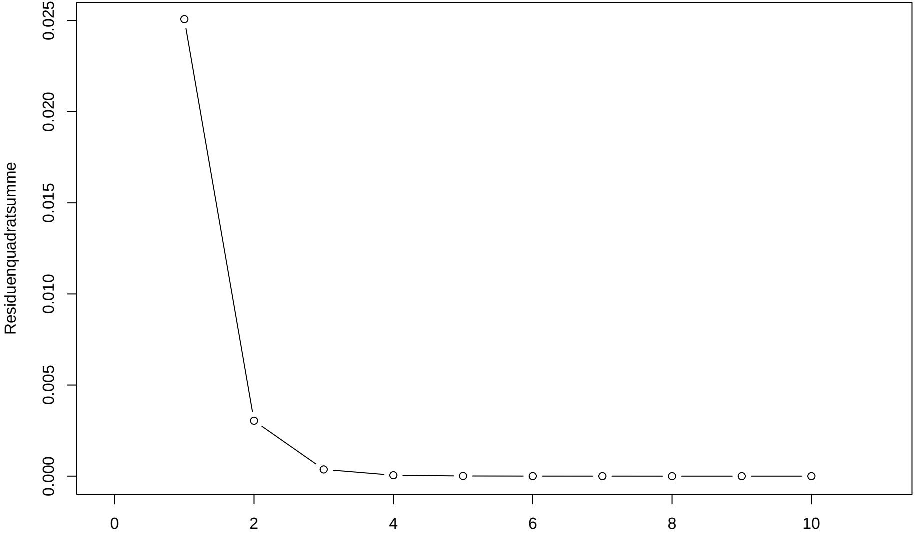
Anmerkung: Gute Klausurübung! Schnappen Sie sich ihre Kolleginnen/Kollegen, simulieren Sie mögliche Welten und üben, üben, üben Sie!

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] 2.508165e-02 3.040011e-03 3.681981e-04 5.153387e-05 9.
##  [6] 1.994291e-06 5.300246e-07 1.605099e-07 1.605099e-07 1.
```



Selbststudium 2: händische PCA – Eigenwertzerlegung

Selbststudium 3: händische EFA – Singulärwertzerlegung

Selbststudium 4: reduzierte Korrelationsmatrix

...Der Output zeigt die erst 8 Iterationen einer PAFA. Vergleichen Sie diese mit ihre am Anfang konstruierte Korrelationsmatrix `$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: PAFA Basis algorithmus

```r
# 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: PAFA – jeweils 1 iteration

```r
EVD_R <- eigen(R) # EVD of R
F <- EVD_R$vectors # Loadings
k <- 7 # k largest PC
# Pre reproduction
Fk <- F[,seq(k)] ; Fk <- cbind(Fk, 0)
# 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)
```

Anmerkung: Da 7 Hauptkomponenten zur Reproduktion genutzt werden, dauert es eine ganze Weile bis etwas passiert!