

# Explorative Faktorenanalyse mit R

## Seminar Forschungsevaluation, Universität Wuppertal

Stephan Holtmeier

kibit GmbH, [stephan@holtmeier.de](mailto:stephan@holtmeier.de)

01. Juni 2013

# Ein Anwendungsbeispiel für eine Faktorenanalyse

## Fragebogenkonzeption für eine Mitarbeiterbefragung

Unsere Beispieldaten sind 100 ausgefüllte Fragebögen mit insgesamt 12 Items (`fa_data.csv`). Die Fragen sind konzeptionell vier Dimensionen zugeordnet. Wir wollen jetzt nachträglich evaluieren, ob unsere intenderte Struktur sich in den tatsächlichen Daten wieder findet.

- 1 Im Folgenden geht es ausschließlich um explorative Faktorenanalysen (in Abgrenzung zu konfirmatorischen Faktorenanalysen, die einfache Sonderfälle von Strukturgleichungsmodellen sind).
- 2 Wir rechnen sowohl die Hauptkomponentenanalyse als auch die Hauptachsenanalyse.
- 3 Wir setzen voraus, dass unsere Rohdaten zumindest annähernd intervallskaliert sind.
- 4 Wir verwenden das Paket *psych*.

# Fragebogen

## I Kommunikation:

1. Ich fühle mich über neue Entwicklungen ausreichend informiert
2. Der Informationsfluss zwischen den Mitarbeitern funktioniert gut
3. Führungskräfte leiten Infos schnell weiter

## II Arbeit:

1. Unsere Arbeitszeiten sind flexibel genug
2. Ich werde angemessen bezahlt
3. Leistung lohnt sich generell in unserem Unternehmen

## III Führung:

1. Unsere Führungskräfte sind glaubwürdig
2. Unsere Führungskräfte zeigen Anerkennung, für gute Arbeit
3. Unsere Führungskräfte formulieren klare Ziele und setzen Prioritäten

## IV Innovation:

1. Unser Unternehmen schafft Rahmenbedingungen für kreatives Arbeiten
2. Meine Ideen und Vorschläge werden berücksichtigt
3. Bei uns werden neue Ideen schnell aufgenommen und umgesetzt

# Rohdatensatz

Bevor wir beginnen: Arbeitsverzeichnis festlegen

Session → Set Working Directory → To Source File Location

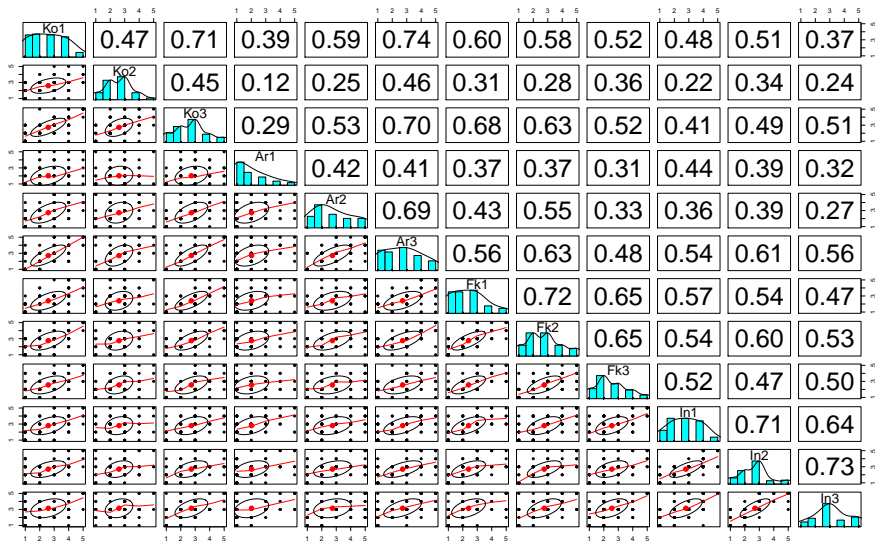
```
library(psych)
fa_data <- read.table("fa_data.csv", sep = ";", header = T)
View(fa_data)  # Dataframe betrachten
head(fa_data, n = 10)  # obere 10 Zeilen betrachten
describe(fa_data)  # Dataframe beschreiben
# Korrelationsmatrix
cor(fa_data, use = "pairwise.complete.obs")
```

Bitte probieren Sie diese Befehle **jetzt** selber aus!

# Rohdatensatz

##		Ko1	Ko2	Ko3	Ar1	Ar2	Ar3	Fk1	Fk2	Fk3	In1	In2	In3
## 1		4	3	3	3	4	4	3	4	4	4	3	4
## 2		2	3	NA	1	2	2	NA	NA	2	2	3	4
## 3		1	3	1	1	3	4	1	3	4	3	3	5
## 4		2	2	2	2	2	1	2	2	2	3	3	3
## 5		1	1	1	3	2	1	1	1	1	3	1	3
## 6		4	3	3	5	4	4	1	3	3	2	3	3
## 7		4	3	4	2	3	2	4	3	5	4	3	NA
## 8		2	4	3	1	2	2	2	2	2	2	3	3
## 9		2	3	1	1	1	1	1	1	2	2	1	1
## 10		3	2	3	2	3	3	3	3	2	3	3	3

# Korrelationsmatrix - pairs.panels(fa\_data)

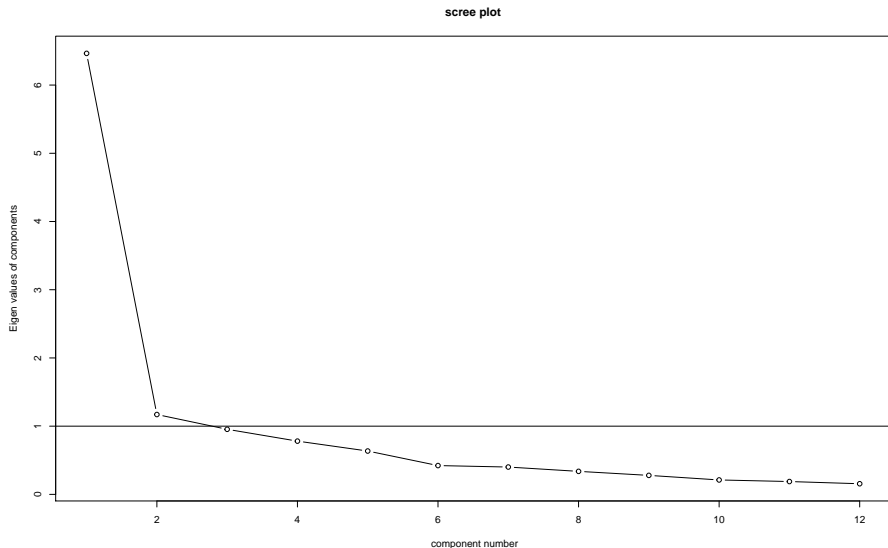


## Eignung der Matrix: Kaiser-Mayer-Olchin-Kriterium

```
KMO(fa_data)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = fa_data)
## Overall MSA = 0.89
## MSA for each item =
## Ko1 Ko2 Ko3 Ar1 Ar2 Ar3 Fk1 Fk2 Fk3 In1 In2
## 0.91 0.85 0.88 0.94 0.87 0.90 0.90 0.92 0.90 0.90 0.88
## In3
## 0.82
```

# Faktorenanzahl ermitteln: Scree-Plot - VSS.scree()





# Faktorenextraktionsmethoden

## 1 Hauptkomponentenanalyse

- a. Principal Components Analysis, PCA
- b. ...versucht die *gesamte* Varianz der manifesten Variablen auf Faktoren zurück zu führen.
- c. Die Diagonale der Korrelationsmatrix enthält Einsen
- d. `principal()`-Funktion

## 2 Hauptachsenanalyse

- a. Principal Axes Factoring, PAF
- b. ...versucht von vornherein nur die gemeinsame Varianz der Variablen durch Faktoren zu erklären. Die Kommunalitäten müssen geschätzt werden. Es bleibt eine Rest-/Fehlervarianz.
- c. In der Diagonalen der Korrelationsmatrix sind die Einsen durch die geschätzten Kommunalitäten ersetzt
- d. `fa()`-Function
- e. Das Argument `fm='pa'` führt zu PAF; es gibt weitere: `minres`, `wls`, `gls`, `ml`, `minchi`

# Hauptkomponentenanalyse

```
fa_data.principal.varimax <- principal(fa_data, nfactors = 4,  
    rotate = "varimax")
```

```
## Loading required package: GPArotation
```

```
print(fa_data.principal.varimax$loadings, cutoff = 0.5)  
print(fa_data.principal.varimax)
```

# Hauptachsenanalyse

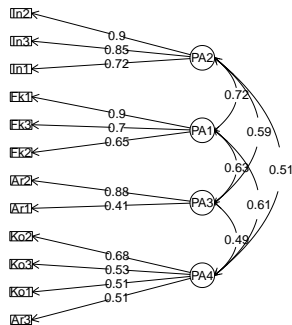
```
# SMC = Squared Multiple Correlation
fa_data.pa.promax <- fa(fa_data, nfactors = 4, SMC = TRUE,
  fm = "pa", rotate = "promax")

print(fa_data.pa.promax)
print(fa_data.pa.promax$loadings, cutoff = 0.5)
```

# Visualisierung

```
fa.diagram(fa_data.pa.promax, simple = TRUE, cut = 0.2,  
           digits = 2)
```

Factor Analysis



# Rotationsmethoden

## 1 orthogonale Rotation

- a. **Varimax**: maximiert die Varianz der quadrierten Faktorladungen pro Faktor, d.h. man möchte pro Faktor möglichst viele beträchtlich hohe oder um Null liegende Ladungen erhalten.
- b. **Quartimax**: versucht, pro Variable möglichst viele hohe oder Null-Ladungen zu erhalten, um die Variablen klarer den Faktoren zuordnen zu können.

## 2 oblique Rotation

- a. **Promax**: ähnlich Varimax, aber mit partiell korrelierten Faktoren.
- b. **Oblimin**: versucht, die Kovarianzen zwischen den quadrierten Faktorladungen aller Paare von Faktoren zu minimieren.

## Bestimmung der Faktorwerte

```
fa.scores <- factor.scores(fa_data, fa_data.pa.promax)
print(fa.scores)
```

```
## $scores
```

##		PA2	PA1	PA3	PA4
##	[1,]	0.793515	0.975927	1.161869	0.548916
##	[2,]	NA	NA	NA	NA
##	[3,]	0.842865	-0.579077	-0.007126	-0.239449
##	[4,]	-0.008371	-0.410038	-0.646651	-1.053185
##	[5,]	-0.826150	-1.239863	-0.425646	-2.201291
##	[6,]	0.095790	-0.459914	1.150128	0.774715
##	[7,]	NA	NA	NA	NA
##	[8,]	-0.298325	-0.558626	-1.049979	0.497016
##	[9,]	-1.645067	-1.296296	-1.440197	-0.720044
##	[10,]	0.154576	0.253817	0.413532	-0.027388
##	[11,]	0.339543	0.527217	1.133252	0.156448