

# Übungszettel Principal Components Analyse und Faktoranalyse

M.Psy.205, Dozent: Dr. Peter Zezula

Johannes Brachem ([johannes.brachem@stud.uni-goettingen.de](mailto:johannes.brachem@stud.uni-goettingen.de))

## Deutsch

### Links

[Übungszettel als PDF-Datei zum Drucken](#)

### Übungszettel mit Lösungen

[Lösungszettel als PDF-Datei zum Drucken](#)

[Der gesamte Übungszettel als .Rmd-Datei](#) (Zum Downloaden: Rechtsklick > Speichern unter...)

### Hinweise zur Bearbeitung

1. Bitte beantworten Sie die Fragen in einer .Rmd Datei. Sie können Sie über `Datei > Neue Datei > R Markdown...` eine neue R Markdown Datei erstellen. Den Text unter dem *Setup Chunk* (ab Zeile 11) können Sie löschen. [Unter diesem Link](#) können Sie auch unsere Vorlage-Datei herunterladen (Rechtsklick > Speichern unter...).
2. Informationen, die Sie für die Bearbeitung benötigen, finden Sie auf der [Website der Veranstaltung](#)
3. Zögern Sie nicht, im Internet nach Lösungen zu suchen. Das effektive Suchen nach Lösungen für R-Probleme im Internet ist tatsächlich eine sehr nützliche Fähigkeit, auch Profis arbeiten auf diese Weise. Die beste Anlaufstelle dafür ist der [R-Bereich der Programmiererplattform Stackoverflow](#)
4. Auf der Website von R Studio finden Sie sehr [hilfreiche Übersichtszettel](#) zu vielen verschiedenen R-bezogenen Themen. Ein guter Anfang ist der [Base R Cheat Sheet](#)

### Ressourcen

Da es sich um eine praktische Übung handelt, können wir Ihnen nicht alle neuen Befehle einzeln vorstellen. Stattdessen finden Sie hier Verweise auf sinnvolle Ressourcen, in denen Sie für die Bearbeitung unserer Aufgaben nachschlagen können.

---

Ressource	Beschreibung
Field, Kapitel 17	Buchkapitel, das Schritt für Schritt erklärt, worum es geht, und wie man Principal Components Analysen in R durchführt. <b>Große Empfehlung!</b>

---

### Tipp der Woche

Es gibt ein neues Paket namens `conflicted`, das Ihnen automatisch eine Fehlermeldung anzeigt, wenn Sie eine Funktion benutzen, die von zwei oder mehr Paketen verwendet wird. So werden Sie schnell

auf potentiell nervtötende Flüchtigkeitsfehler aufmerksam gemacht. Alles, was Sie dafür tun müssen, ist das Paket zu installieren (`install.packages("conflicted")`) und zu Beginn Ihrer R-Skripte zu laden (`library(conflicted)`).

**Achtung** Wir haben Indizien dafür, dass ein laufendes `conflicted` die Stabilität von R unter Umständen stören kann. Wenn Sie den Verdacht haben, schalten Sie das Paket bitte vorsichtshalber aus. `detach(conflicted, unload=TRUE)`. Es gibt Alternativen, z. B. `conflicts(detail=TRUE)`, vgl. [https://md.psych.bio.uni-goettingen.de/mv/unit/block\\_intro/block\\_intro\\_virt.html#command\\_masking](https://md.psych.bio.uni-goettingen.de/mv/unit/block_intro/block_intro_virt.html#command_masking)

### Beispiel

```
library(conflicted)
library(dplyr)

filter(mtcars, am & cyl == 8)
```

```
Fehler: filter found in 2 packages. You must indicate which one you want with ::
* dplyr::filter
* stats::filter
```

## 1) Daten einlesen, allgemeines zum Thema

1. Laden Sie die nötigen Pakete (dazu gehört heute auch `psych`) und setzen Sie ein sinnvolles Arbeitsverzeichnis.
2. Laden Sie den Datensatz `raq.dat` über den Link <https://pzezula.pages.gwdg.de/data/raq.dat> herunter.
3. Lesen Sie den Datensatz unter dem Namen `raq_data` in R ein. Jede Zeile enthält die Antworten einer Versuchsperson.
4. Lesen Sie die Abschnitte “Überblick über die Items” und “Worum geht es?”, um einen Überblick über die Thematik für diesen Übungszettel zu bekommen.

### Lösung

```
# Pakete laden
library(tidyverse)
library(psych)
```

```
# Arbeitsverzeichnis setzen
# Ersetzen Sie hier den Dateipfad durch Ihren eigenen Dateipfad
setwd("~/ownCloud/_Arbeit/Hiwi Peter/gitlab_sheets")
```

### Unteraufgabe 1

**Unteraufgabe 2** Bitte folgen Sie den Anweisungen in der Aufgabenstellung.

```
# raq_data <- read_delim("data/raq.dat", delim = "\t")
raq_data <- read_delim("http://md.psych.bio.uni-goettingen.de/mv/data/af/raq.dat", delim = "\t")
```

### Unteraufgabe 3

```
##
## -- Column specification -----
## cols(
##   .default = col_double()
## )
## i Use `spec()` for the full column specifications.
```

**Unteraufgabe 4** Bitte folgen Sie den Anweisungen in der Aufgabenstellung.

### Überblick über die Items

Der Datensatz enthält die Ergebnisse eines fiktiven Fragebogens (R Anxiety Questionnaire, RAQ) zur Angst vor Statistik (Field, S. 873). Hier sehen Sie die einzelnen Fragen.

SD = Strongly Disagree, D = Disagree, N = Neither, A = Agree, SA = Strongly Agree						
		SD	D	N	A	SA
1	Statistics make me cry	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	My friends will think I'm stupid for not being able to cope with R	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Standard deviations excite me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	I dream that Pearson is attacking me with correlation coefficients	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	I don't understand statistics	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	I have little experience of computers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	All computers hate me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	I have never been good at mathematics	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	My friends are better at statistics than me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Computers are useful only for playing games	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	I did badly at mathematics at school	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	People try to tell you that R makes statistics easier to understand but it doesn't	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	I worry that I will cause irreparable damage because of my incompetence with computers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	Computers have minds of their own and deliberately go wrong whenever I use them	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	Computers are out to get me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	I weep openly at the mention of central tendency	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17	I slip into a coma whenever I see an equation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18	R always crashes when I try to use it	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19	Everybody looks at me when I use R	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20	I can't sleep for thoughts of eigenvectors	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21	I wake up under my duvet thinking that I am trapped under a normal distribution	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
22	My friends are better at R than I am	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
23	If I am good at statistics people will think I am a nerd	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Worum geht es?

Dieser Übungszettel behandelt die explorative Faktorenanalyse (EFA) und die Hauptkomponentenanalyse (Principal Components Analysis, PCA).

Beide Verfahren dienen grob gesagt dazu, Variablen zu Gruppen zusammenzufassen. Dadurch kann man z.B. mit Multikollinearität (ein Problem von Regressionen, wenn mehrere Prädiktoren sehr hoch miteinander korrelieren) umgehen (PCA), oder nicht direkt beobachtbare (latente) Konstrukte erschließen (EFA). Mit der konfirmatorischen Faktorenanalyse (Confirmatory Factor Analysis, CFA) können zusätzlich Theorien, die z.B. auf Grundlage von EFA aufgestellt wurden, überprüft werden.

Wir behandeln hier nur die ersten beiden Verfahren, also **Principal Components Analysis (PCA)** und **Exploratory Factor Analysis (EFA)**. Beide sind sich sehr ähnlich, tatsächlich kann die PCA als Sonderfall der EFA aufgefasst werden.

**Tabelle:** Übersicht über Faktoranalyse-Verfahren

Verfahren	Bedeutung	Anwendung
Principal Components Analysis (PCA)	Variablen, die die gleiche Information beinhalten, werden zusammengefasst.	Bspw. zur Vermeidung von Multikollinearität in Regressionen.
Explorative Factor Analysis (EFA)	Variablen, die auf das selbe zugrundeliegende Merkmal, z.B. einer Person, zurückzuführen sind, werden zusammengefasst.	Bspw. Konstruktion von Fragebögen.
Confirmatory Factor Analysis (CFA)	Variablen, die auf das selbe zugrundeliegende Merkmal, z.B. einer Person, zurückzuführen sind, werden zusammengefasst.	Bspw. Test von Theorien über Faktoren

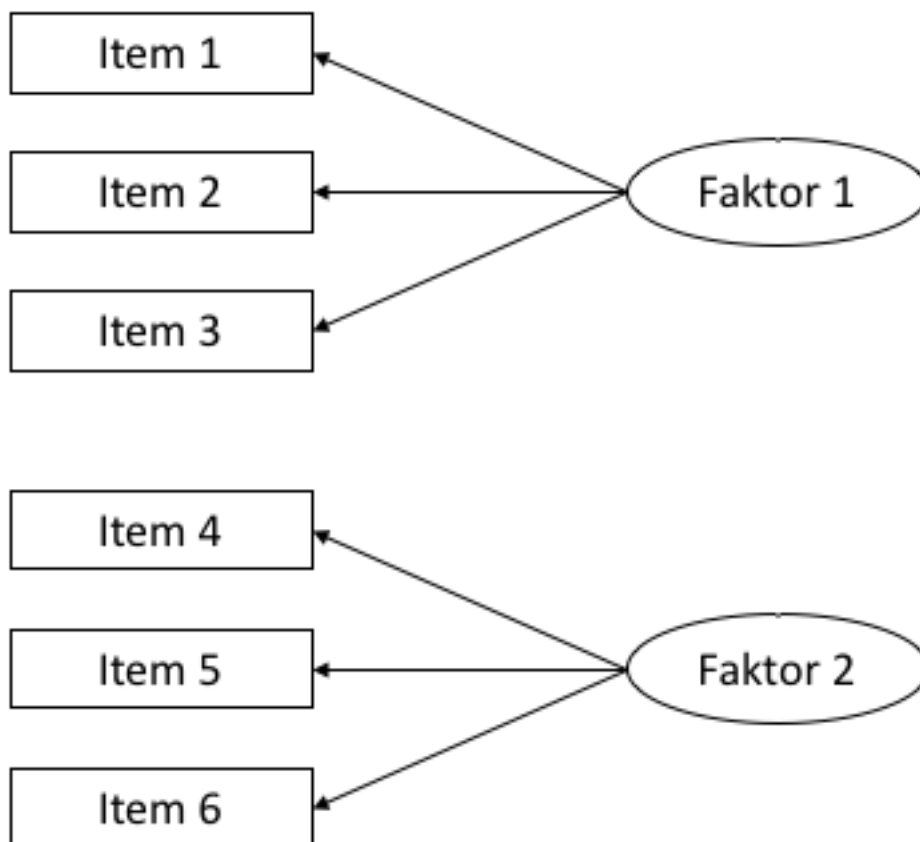


Figure 1: **Abbildung:** Beispiel für ein Pfaddiagramm. Solche Diagramme werden häufig verwendet, um Faktoranalysen zu visualisieren. In diesem Fall werden die *manifesten*, d.h. direkt beobachteten, von VP beantworteten, Items 1-3 auf den zugrundeliegenden *latenten*, d.h. nicht beobachteten, Faktor 1 zurückgeführt. Ebenso verhält es sich mit den Items 4-6 und Faktor 2.

## 2) Erster Blick auf die Daten

Wie bei anderen Analysen auch müssen wir zunächst prüfen, ob die Daten überhaupt geeignet dafür sind, eine Faktoranalyse durchzuführen. Das heißt vor allem, dass wir überprüfen, ob es ausreichend Beziehungen zwischen den Daten gibt, die aber auch nicht zu stark sein dürfen. Wenn zwei Variablen praktisch identisch sind (Korrelation .90 oder höher), dann bereitet das der Analyse Probleme.

Wir inspizieren die Daten zunächst per Augemaß, führen dann einen Test durch und schauen uns zwei Indices an, die uns Informationen über die Angemessenheit einer Faktorenanalyse liefern.

Erklärungen dazu (nähere Information finden Sie in Field (2012)):

Kriterium	Bedeutung	Anwendung
Korrelationsmatrix	Gibt es augenscheinlich angemessene Zusammenhänge zwischen den betrachteten Variablen?	Hinweise auf Probleme gibt es, wenn 1) Kaum Korrelation von .30 oder größer vorliegen, 2) (Viele) Korrelationen über .90 vorliegen, 3) Einzelne oder mehrere Variablen überhaupt nur sehr geringe Korrelationen mit allen anderen Variablen aufweisen.
Bartlett's Test	Handelt es sich bei der Korrelationsmatrix um eine Einheitsmatrix?	Wenn der Test signifikant wird, handelt es sich <b>nicht</b> um eine Einheitsmatrix. Das ist gut für die Faktorenanalyse. Ein nicht-signifikanter Test wäre ein Hinweis auf ein Problem.
Kaiser-Meyer-Olkin Maß (KMO)	Gibt es Muster in der Korrelationsmatrix, die durch Faktorenanalyse aufgedeckt werden können?	Kann Werte von 0-1 annehmen, je näher an 1, desto besser. Es gibt ein Gesamtmaß und Maße für jede einzelne Variable. Beide sollten betrachtet werden.
Determinante der Korrelationsmatrix	Sind die Beziehungen der Variablen in der Matrix stark (0) oder schwach (1)?	Die Determinante sollte klein, aber nicht kleiner als 0.00001 sein, damit eine Faktorenanalyse durchgeführt werden kann.

1. Nutzen Sie den Befehl `cor()`, um eine Korrelationsmatrix auf Grundlage des Datensatzes `raq_data` zu erstellen. *Tipp: Mit der Funktion `round()` können Sie die Ergebnisse abrunden, so dass sie leichter lesbar sind.*
2. Inspizieren Sie die Korrelationsmatrix, indem Sie einen Blick darauf werfen. Gibt es einen Hinweis auf Probleme für die Faktorenanalyse?
3. Wenden Sie den Befehl `cortest.bartlett()` aus dem Paket `psych` auf den Datensatz oder auf die Korrelationsmatrix an (beides liefert identischer Ergebnisse, wenn der Datensatz ausschließlich dieselben Variablen enthält wie die Matrix). Gibt es einen Hinweis auf Probleme für die Faktorenanalyse?  
*Hinweis dazu:*
  - i) Wenn Sie die Funktion auf die Korrelationsmatrix anwenden, müssen Sie mit dem Argument `n =` angeben, wie groß die Stichprobe ist.
4. Wenden Sie den Befehl `KMO()` aus dem Paket `psych` auf den Datensatz oder auf die Korrelationsmatrix an (beides liefert identischer Ergebnisse, wenn der Datensatz ausschließlich dieselben Variablen enthält wie die Matrix). Gibt es einen Hinweis auf Probleme für die Faktorenanalyse?
5. Wenden Sie den Befehl `det()` auf die **Korrelationsmatrix** an. Gibt es einen Hinweis auf Probleme für die Faktorenanalyse?

## Lösung

```
raq_matrix <- raq_data %>% cor() %>% round(2)
```

### Unteraufgabe 1

### Unteraufgabe 2

- a) Es gibt Korrelationen über .30
- b) Es gibt keine Korrelationen über .90
- c) Es gibt keine Variablen, die besorgniserregend wenig mit anderen korrelieren.

```
# Anwendung auf die Matrix  
psych::cortest.bartlett(raq_matrix, n = 2571)
```

### Unteraufgabe 3

```
## $chisq  
## [1] 19323.64  
##  
## $p.value  
## [1] 0  
##  
## $df  
## [1] 253
```

```
# Anwendung auf den Datensatz  
psych::cortest.bartlett(raq_data)
```

```
## R was not square, finding R from data
```

```
## $chisq  
## [1] 19334.49  
##  
## $p.value  
## [1] 0  
##  
## $df  
## [1] 253
```

Der Test ist signifikant: Es gibt Beziehungen zwischen den Daten.

```
# Anwendung der Funktion auf den Datensatz  
kmo_raq <- psych::KMO(raq_data)  
  
# Anzeigen der Ergebnisse  
kmo_raq
```

## Unteraufgabe 4

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: psych::KMO(r = raq_data)
## Overall MSA = 0.93
## MSA for each item =
## Q01 Q02 Q03 Q04 Q05 Q06 Q07 Q08 Q09 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20
## 0.93 0.87 0.95 0.96 0.96 0.89 0.94 0.87 0.83 0.95 0.91 0.95 0.95 0.97 0.94 0.93 0.93 0.95 0.94 0.89
## Q23
## 0.77
```

```
# Anzeigen der Einzel-Werte in aufsteigender Reihenfolge
kmo_raq$MSAi %>% sort()
```

```
##      Q23      Q09      Q08      Q02      Q22      Q20      Q06      Q11      Q21      Q01
## 0.7663994 0.8337295 0.8713055 0.8747754 0.8784508 0.8890514 0.8913314 0.9059338 0.9293369 0.9297610
##      Q16      Q15      Q19      Q07      Q18      Q13      Q10      Q03      Q12      Q04
## 0.9336439 0.9404402 0.9407021 0.9416800 0.9479508 0.9482270 0.9486858 0.9510378 0.9548324 0.9553403
##      Q14
## 0.9671722
```

Der Gesamt-Wert liegt mit 0.93 im hervorragenden Bereich, und der geringste Einzelwert ist mit 0.77 immer noch im guten Bereich.

```
# Determinante der Korrelationsmatrix
det(raq_matrix)
```

## Unteraufgabe 5

```
## [1] 0.0005293425
```

Die Determinante ist klein, was auf starke Beziehungen zwischen den Variablen hindeutet. Gleichzeitig ist sie größer als der Cutoff von 0.00001, d.h. die Beziehungen zwischen den Variablen sind nicht *zu* stark. Wir können hier also eine Faktoranalyse durchführen.

## 3) Faktoren identifizieren

Bei explorativem Vorgehen wissen wir vor der Analyse nicht, wie viele Faktoren vorliegen. Wir nutzen zur Identifikation einen sogenannten *Scree-Plot* und die *Eigenvalues*, Werte aus einer ersten Analyse.

**Wir arbeiten hier zunächst nur mit der PCA. Das Vorgehen für die EFA ist beinahe identisch. Bei Unterschieden weisen wir darauf hin. In einer späteren Aufgabe thematisieren wir Unterschiede in den Ergebnissen.**

Der allgemeine Befehl für eine PCA in R stammt aus dem Paket `psych` und lautet: `principal(r = , nfactors = , rotate = )`. Dabei steht `r` für die Korrelationsmatrix, bzw. die zugrundeliegenden Rohdaten, `nfactors` für die Anzahl von Faktoren, die wir identifizieren möchten, und `rotate` für die Art der "Rotation". Die Rotation wird in einer späteren Aufgabe thematisiert.

Der Befehl für die explorative Faktorenanalyse ist genau gleich aufgebaut und lautet: `fa(r = , nfactors = , rotate = )`.

**Tabelle 3.1:** Wichtige Komponenten des Outputs einer PCA / EFA.



Komponente	Wo im Output?	Bedeutung
Faktorladungen (factor loadings)	Große Tabelle, oben im Output	Korrelation der jeweiligen Variable mit dem jeweiligen Faktor.
h2	Am rechten Rand der Tabelle mit den Faktorladungen	<i>Communality</i> : Maß für den Anteil der Varianz der jeweiligen Variable, der durch die Faktoren des Modells erklärt werden kann.
u2	Neben h2	<i>Uniqueness</i> : Anteil einzigartiger Varianz der jeweiligen Variable. Berechnung als $1 - h^2$
Eigenvalues (SS loadings)	Tabelle unter den Faktorladungen, erste Zeile	Summe der quadrierten Faktorladungen pro Faktor. Werden dargestellt als Anteil erklärter Varianz. Bei 23 Variablen wie in unserem Fall entspricht ein eigenvalue von 7.29 (PC1) der Erklärung von $7.29 / 23 = 0.32$ , also 32% der Varianz in den Daten. Das ist auch in der zweiten Zeile der Tabelle abzulesen.

**Tabelle 3.2:** Kriterien zur Bestimmung der Anzahl sinnvoller Faktoren.

Kriterium	Welche Faktoren erhalten?
Kaiser	Faktoren mit Eigenvalue über 1 Wahrscheinlich ein gutes Kriterium, wenn weniger als 30 Variablen untersucht werden und die communalities <i>nach</i> der Faktor-Extraktion alle über 0.7 liegen. Wenn die Stichprobengröße über 250 liegt, sind auch communalities von 0.6 akzeptabel. Bei größeren Stichproben können wiederum noch kleinere communalities akzeptabel sein.
Jolliffe	Faktoren mit Eigenvalue über 0.7 Jolliffe war der Meinung, dass Kaisers Kriterium zu konservativ sei.
Scree Plot	Zeigt die Wichtigkeit aller Faktoren der ersten Analyse. Als Cut-Off wird der <i>point of inflexion</i> gewählt: Der Punkt, an dem sich die Steigung der Linie stark verändert, in der Regel von fast vertikal auf fast horizontal. Alle Faktoren links von diesem Punkt werden erhalten, die Faktoren rechts nichts.

- Führen Sie eine PCA mit `raq_data` durch und speichern Sie diese unter dem Namen `pc1`. In der ersten Analyse möchten wir für `nfactors` die Gesamtzahl unserer untersuchten Variablen eingeben, in diesem Fall also 23. Für die Rotation wählen Sie zunächst bitte `rotate = "none"`.
  - Achtung, Unterschied zur EFA:** In der EFA geben wir beim ersten Modell nicht die Gesamtzahl der Variablen ein, sondern einen geringeren Wert. In diesem Fall könnte man z.B. 18 eingeben.
- Erstellen Sie einen Scree-Plot mit dem Befehl `plot(pc1$values, type = "b")`. Verstehen Sie den Code?
- Betrachten Sie den Scree-Plot und die Eigenvalues im Output der PCA. Entscheiden Sie auf Grundlage von Tabelle 3.2, wie viele Faktoren Sie extrahieren möchten.
- Führen Sie nun die PCA erneut durch, diesmal mit `nfactors = 4`, da wir vier Faktoren extrahieren möchten. Betrachten Sie die *communalities*. Ist die Extraktion von vier Faktoren nach Kaisers Kriterium gerechtfertigt?
  - Sie können zu diesem Zweck auch den Mittelwert der *communalities* betrachten.
- Eine weitere, **sehr empfehlenswerte Methode** zur Bestimmung der Anzahl von Faktoren oder Hauptkomponenten ist die Parallel-Analyse. Dabei werden Zufallsdaten erzeugt und mit den realen

Daten verglichen. Dieses Verfahren erlaubt eine weniger subjektive Entscheidung. Das Paket `psych` stellt mit dem Befehl `fa.parallel(<data>, fa = <"method">)` eine unkomplizierte Methode zur Durchführung zur Verfügung. Setzen Sie für `<data>` den Datensatz ein, und für `<"method">` die gewünschte Methode. In der PCA ist das `pc`, in den Faktorenanalyse `fa` (Sie können auch `both` eingeben). Führen Sie eine Parallel-Analyse für den vorliegenden Datensatz durch. Was ist Ihre Schlussfolgerung? (Näheres zur Interpretation finden Sie hier: <http://md.psych.bio.uni-goettingen.de/mv/unit/fa/fa.html#parallelanalyse>)

6. Betrachten Sie die letzte Zeile des Outputs. Dort finden Sie ein weiteres Maß für die Passungs des Modells. Dieses Maß, wie so viele, geht von 0 bis 1. Je näher an der 1, desto besser ist die Passung des Modells auf die Daten. Werte über 0.95 werden generell als gute Passung angesehen. Eine genauere Erklärung zu diesem Wert finden Sie in Field (2012), S. 889 - 891.

## Lösung

```
# Analyse
pc1 <- psych::principal(raq_matrix, nfactors = ncol(raq_matrix), rotate = "none")
```

```
# Output
pc1
```

### Unteraufgabe 1

```
## Principal Components Analysis
## Call: psych::principal(r = raq_matrix, nfactors = ncol(raq_matrix),
##      rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9  PC10  PC11  PC12  PC13  PC14  PC15  PC16
## Q01  0.59  0.18 -0.21  0.13 -0.39 -0.13 -0.21 -0.10 -0.02 -0.05  0.12 -0.11  0.27 -0.29  0.18 -0.11
## Q02 -0.30  0.55  0.16  0.02 -0.01 -0.37  0.26 -0.38  0.01 -0.07  0.31  0.26  0.00  0.03 -0.24  0.03
## Q03 -0.63  0.29  0.21 -0.06  0.03  0.02  0.04 -0.06  0.21  0.11  0.12  0.03  0.10  0.08  0.40  0.09
## Q04  0.63  0.15 -0.14  0.15 -0.21 -0.14 -0.06  0.09 -0.07 -0.01 -0.03  0.33 -0.35 -0.14  0.09 -0.31
## Q05  0.56  0.10 -0.07  0.14 -0.41 -0.18 -0.05  0.09  0.26  0.03 -0.29  0.16  0.19  0.46 -0.03  0.10
## Q06  0.56  0.09  0.57 -0.04  0.18  0.02 -0.01  0.04  0.00 -0.01 -0.13  0.20  0.22 -0.07  0.11 -0.20
## Q07  0.68  0.04  0.25  0.11  0.17 -0.07  0.03  0.05 -0.08  0.12 -0.28  0.20 -0.01 -0.21 -0.02  0.25
## Q08  0.55  0.40 -0.32 -0.42  0.13  0.10 -0.06 -0.06  0.01 -0.04 -0.09  0.03 -0.02  0.04 -0.04 -0.03
## Q09 -0.28  0.63 -0.01  0.10  0.18 -0.25  0.00 -0.01  0.16  0.28 -0.26 -0.38 -0.17 -0.05  0.11 -0.09
## Q10  0.44  0.04  0.37 -0.09 -0.35  0.22  0.44  0.02  0.32 -0.26 -0.10 -0.23 -0.14 -0.15 -0.11 -0.03
## Q11  0.65  0.25 -0.20 -0.41  0.12  0.18 -0.02  0.02  0.08 -0.16  0.02  0.03  0.03  0.03 -0.01 -0.09
## Q12  0.67 -0.05  0.04  0.24  0.04 -0.09 -0.13  0.05  0.05 -0.13  0.21 -0.08 -0.43  0.21  0.10  0.09
## Q13  0.67  0.07  0.29  0.00  0.14  0.04 -0.22  0.00  0.08 -0.20  0.26 -0.08  0.01  0.09  0.15  0.13
## Q14  0.66  0.03  0.19  0.14  0.09 -0.04 -0.09 -0.04 -0.14  0.19  0.07 -0.27  0.11  0.19 -0.34 -0.25
## Q15  0.59  0.01  0.12 -0.11 -0.09  0.30  0.33 -0.06 -0.25  0.42  0.10  0.09 -0.10  0.18  0.18 -0.04
## Q16  0.68  0.01 -0.14  0.08 -0.32 -0.01  0.13 -0.11 -0.18  0.19  0.15 -0.18  0.10 -0.10  0.04  0.21
## Q17  0.64  0.33 -0.20 -0.35  0.09  0.06 -0.04  0.03 -0.04  0.01  0.02 -0.03 -0.01  0.00 -0.05  0.19
## Q18  0.70  0.03  0.29  0.13  0.17 -0.09 -0.11  0.05 -0.07 -0.11  0.07 -0.11  0.09 -0.03  0.02  0.00
## Q19 -0.43  0.39  0.09 -0.02 -0.14  0.06 -0.03  0.69  0.06  0.16  0.30  0.07  0.05 -0.07 -0.14  0.01
## Q20  0.44 -0.20 -0.42  0.28  0.34  0.02  0.34  0.05  0.32  0.03  0.18  0.03  0.18  0.03  0.10 -0.20
## Q21  0.66 -0.05 -0.20  0.27  0.25 -0.13  0.18  0.13  0.12  0.06 -0.02  0.05  0.01 -0.16 -0.09  0.24
## Q22 -0.30  0.47 -0.13  0.37  0.06  0.11  0.26  0.16 -0.45 -0.39 -0.14 -0.07  0.10  0.14  0.11  0.00
```

```

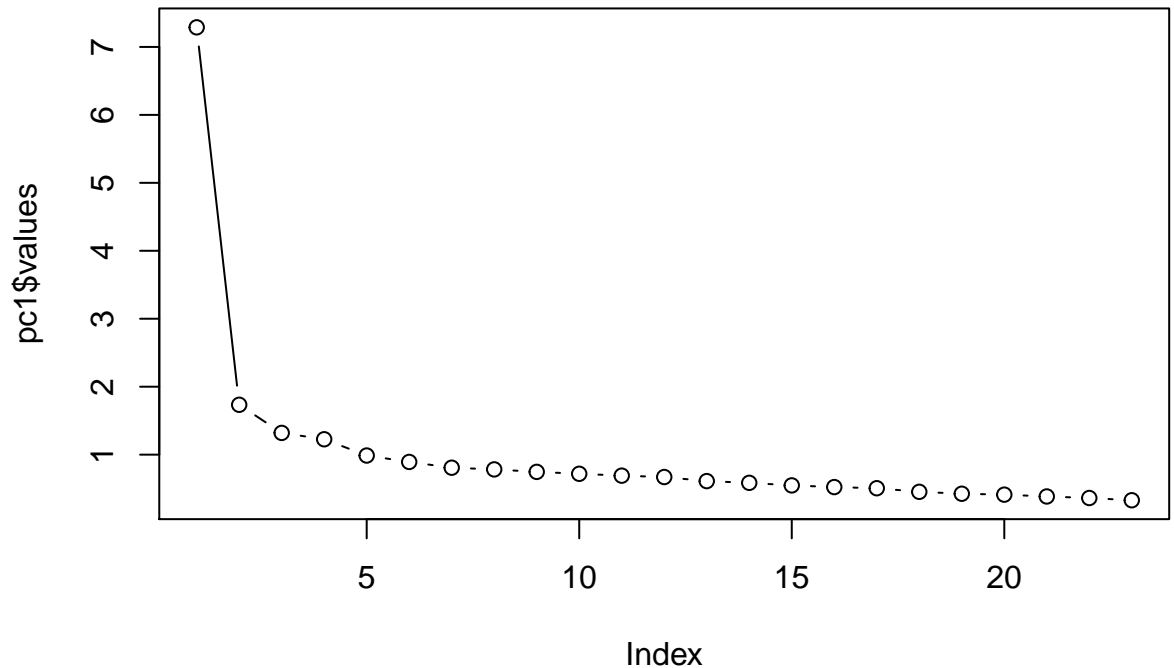
## Q23 -0.14  0.37 -0.03  0.51  0.00  0.62 -0.26 -0.23  0.17  0.08 -0.01  0.12 -0.02 -0.06 -0.14  0.05
##      PC19 PC20  PC21  PC22  PC23 h2      u2 com
## Q01  0.16  0.03 -0.21  0.02  0.00  1 0.0e+00 6.0
## Q02  0.02  0.00 -0.02  0.02  0.02  1 6.7e-16 6.3
## Q03  0.07  0.05  0.00 -0.03  0.04  1 1.2e-15 4.3
## Q04 -0.22  0.01  0.07 -0.01  0.02  1 1.1e-16 4.9
## Q05 -0.04 -0.02 -0.02  0.02  0.02  1 1.0e-15 5.4
## Q06  0.09 -0.08  0.10 -0.28 -0.15  1 1.0e-15 4.5
## Q07  0.16  0.24  0.10  0.22  0.12  1 1.2e-15 4.1
## Q08  0.14 -0.11  0.04  0.23 -0.33  1 1.1e-15 5.7
## Q09 -0.07 -0.05  0.03  0.00  0.04  1 5.6e-16 4.9
## Q10 -0.01  0.04 -0.03  0.02 -0.04  1 3.3e-16 7.8
## Q11  0.13 -0.15  0.09 -0.04  0.40  1 6.7e-16 4.1
## Q12  0.34  0.06 -0.01 -0.11 -0.05  1 1.1e-16 3.9
## Q13 -0.31 -0.09  0.13  0.13 -0.02  1 0.0e+00 4.2
## Q14  0.04  0.06  0.03 -0.02  0.03  1 7.8e-16 4.4
## Q15 -0.02 -0.07 -0.21  0.08  0.03  1 6.7e-16 5.5
## Q16 -0.02 -0.06  0.37 -0.06 -0.03  1 0.0e+00 4.0
## Q17 -0.19  0.35 -0.14 -0.28 -0.04  1 6.7e-16 4.5
## Q18 -0.09  0.01 -0.22  0.19  0.05  1 5.6e-16 3.3
## Q19  0.05  0.00  0.03  0.05 -0.02  1 2.2e-16 3.4
## Q20 -0.02  0.20  0.09  0.07 -0.04  1 6.7e-16 8.5
## Q21 -0.03 -0.35 -0.17 -0.14 -0.02  1 1.0e-15 4.5
## Q22  0.00  0.01  0.01 -0.02  0.00  1 3.3e-16 6.9
## Q23 -0.01  0.00  0.00 -0.02  0.00  1 5.6e-16 4.1
##
##
##      PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9  PC10  PC11  PC12  PC13  PC14  PC15  PC16
## SS loadings      7.29 1.73 1.32 1.23 0.99 0.89 0.81 0.78 0.75 0.72 0.69 0.67 0.61 0.58 0.55 0.53
## Proportion Var   0.32 0.08 0.06 0.05 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.02 0.02
## Cumulative Var   0.32 0.39 0.45 0.50 0.55 0.58 0.62 0.65 0.69 0.72 0.75 0.78 0.80 0.83 0.85 0.88
## Proportion Explained 0.32 0.08 0.06 0.05 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.02 0.02
## Cumulative Proportion 0.32 0.39 0.45 0.50 0.55 0.58 0.62 0.65 0.69 0.72 0.75 0.78 0.80 0.83 0.85 0.88
##
##      PC19 PC20 PC21 PC22 PC23
## SS loadings      0.42 0.41 0.38 0.36 0.33
## Proportion Var   0.02 0.02 0.02 0.02 0.01
## Cumulative Var   0.94 0.95 0.97 0.99 1.00
## Proportion Explained 0.02 0.02 0.02 0.02 0.01
## Cumulative Proportion 0.94 0.95 0.97 0.99 1.00
##
## Mean item complexity = 5
## Test of the hypothesis that 23 components are sufficient.
##
## The root mean square of the residuals (RMSR) is 0
##
## Fit based upon off diagonal values = 1

```

```

# Ohne Pipe
plot(pci$values, type = "b")

```



### Unteraufgabe 2

```
# Mit pipe
# pc1$values %>% plot(type = "b")
```

`pc1$values` greift auf das Objekt `values` im Output der PCA zu. `type = "b"` bedeutet, dass sowohl Punkte als auch Linien (both) geplottet werden sollen.

**Unteraufgabe 3** Der Scree-Plot wird relativ schnell flach, besonders allerdings nach dem 4. Faktor. Außerdem gibt es vier Faktoren mit Eigenvalues über 1, was nach Kaisers Kriterium passen würde. Nach Jollifes Kriterium würden wir 10 Faktoren extrahieren. Wir würden an dieser Stelle zunächst mit vier Faktoren weiterarbeiten, da Scree Plot und Kaisers Kriterium eher übereinstimmen. Es gibt hier kein völlig objektives Entscheidungskriterium, allerdings können wir in der Folge anhand der *communalities* überprüfen, ob die Entscheidung angemessen war.

Man kann anhand des Scree-Plots auch dafür argumentieren, dass nur zwei Faktoren extrahiert werden sollten. Probieren Sie das gerne auch selbst noch einmal aus!

```
# Analyse mit vier Faktoren
pc2 <- psych::principal(raq_data, nfactores = 4, rotate = "none")

# Output
pc2
```

### Unteraufgabe 4

```

## Principal Components Analysis
## Call: psych::principal(r = raq_data, nfactors = 4, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1  PC2  PC3  PC4  h2  u2 com
## Q01  0.59  0.18 -0.22  0.12  0.43  0.57  1.6
## Q02 -0.30  0.55  0.15  0.01  0.41  0.59  1.7
## Q03 -0.63  0.29  0.21 -0.07  0.53  0.47  1.7
## Q04  0.63  0.14 -0.15  0.15  0.47  0.53  1.3
## Q05  0.56  0.10 -0.07  0.14  0.34  0.66  1.2
## Q06  0.56  0.10  0.57 -0.05  0.65  0.35  2.1
## Q07  0.69  0.04  0.25  0.10  0.55  0.45  1.3
## Q08  0.55  0.40 -0.32 -0.42  0.74  0.26  3.5
## Q09 -0.28  0.63 -0.01  0.10  0.48  0.52  1.5
## Q10  0.44  0.03  0.36 -0.10  0.33  0.67  2.1
## Q11  0.65  0.25 -0.21 -0.40  0.69  0.31  2.2
## Q12  0.67 -0.05  0.05  0.25  0.51  0.49  1.3
## Q13  0.67  0.08  0.28 -0.01  0.54  0.46  1.4
## Q14  0.66  0.02  0.20  0.14  0.49  0.51  1.3
## Q15  0.59  0.01  0.12 -0.11  0.38  0.62  1.2
## Q16  0.68  0.01 -0.14  0.08  0.49  0.51  1.1
## Q17  0.64  0.33 -0.21 -0.34  0.68  0.32  2.4
## Q18  0.70  0.03  0.30  0.13  0.60  0.40  1.4
## Q19 -0.43  0.39  0.10 -0.01  0.34  0.66  2.1
## Q20  0.44 -0.21 -0.40  0.30  0.48  0.52  3.2
## Q21  0.66 -0.06 -0.19  0.28  0.55  0.45  1.6
## Q22 -0.30  0.47 -0.12  0.38  0.46  0.54  2.8
## Q23 -0.14  0.37 -0.02  0.51  0.41  0.59  2.0
##
##
##      PC1  PC2  PC3  PC4
## SS loadings      7.29  1.74  1.32  1.23
## Proportion Var   0.32  0.08  0.06  0.05
## Cumulative Var   0.32  0.39  0.45  0.50
## Proportion Explained 0.63  0.15  0.11  0.11
## Cumulative Proportion 0.63  0.78  0.89  1.00
##
## Mean item complexity = 1.8
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is 0.06
## with the empirical chi square 4006.15 with prob < 0
##
## Fit based upon off diagonal values = 0.96

# Mittlere communalität
pc2$communality %>% mean()

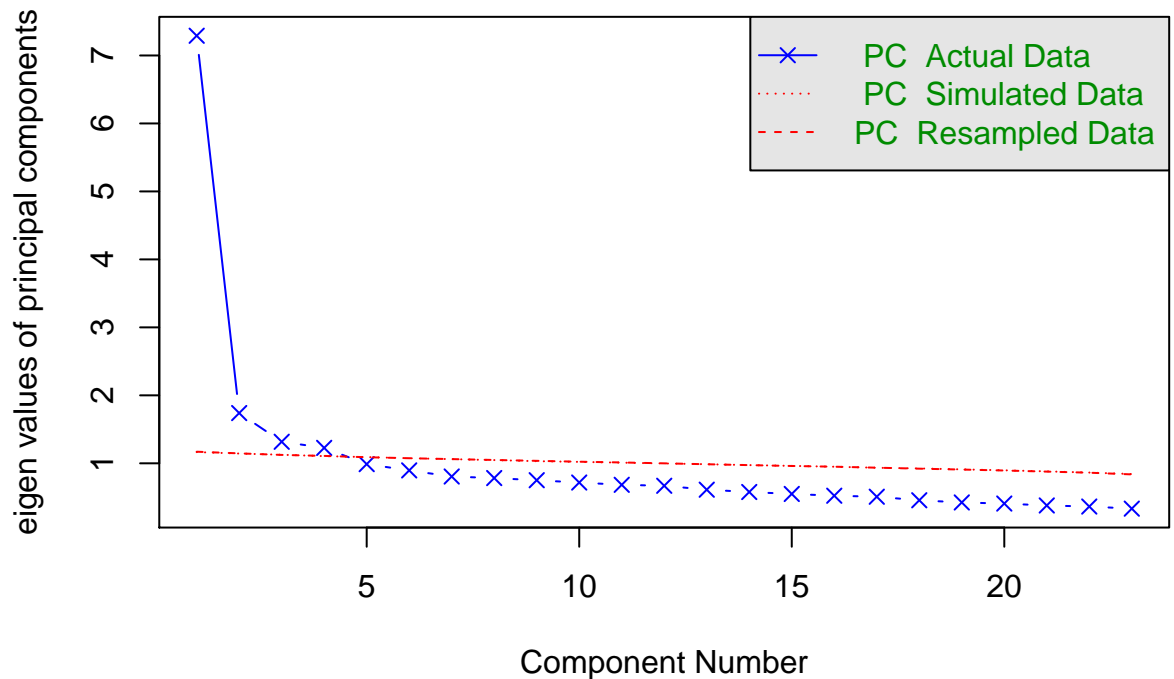
## [1] 0.5031663

```

Die mittlere communalität ist mit .503 unter der Schwelle von .60, die für Kaisers Kriterium bei Stichproben ab 250 VP angegeben wird. Nach Field (2012) ist der Wert dennoch akzeptabel, da hier eine deutlich größere Stichprobe ( $n = 2571$ ) vorliegt, für die diese Cut-Off-Werte vermutlich unterschritten werden dürfen. Darüber hinaus spricht auch der Scree-Plot tendenziell für die Extraktion von 4 Faktoren.

```
items_parallel <- psych::fa.parallel(raq_data, fa = "pc")
```

## Parallel Analysis Scree Plots



### Unteraufgabe 5

```
## Parallel analysis suggests that the number of factors = NA and the number of components = 4
```

Auch die Parallelanalyse legt uns nahe, vier Faktoren (in diesem Fall *components*) zu extrahieren.

**Unteraufgabe 6** Mit einem Wert von 0.96 weist das Modell eine gute Passung auf.

## 5) Rotation und Scores

Faktorrotation ist ein wichtiger Schritt in der Faktorenanalyse, der es uns erlaubt, Faktoren deutlicher voneinander abzugrenzen. Das funktioniert durch eine Art “Drehung” (siehe Abbildung) des Koordinatensystem der Faktoren. Es gibt zwei Arten von Rotation:

Rotation	Bedeutung
Orthogonal	Faktoren werden rotiert, aber bleiben unabhängig voneinander (Korrelation = 0).
Oblique	Faktoren werden rotiert und dürfen miteinander korrelieren.

Genauereres können Sie in Field (2012), Kapitel 17.3.9 finden.

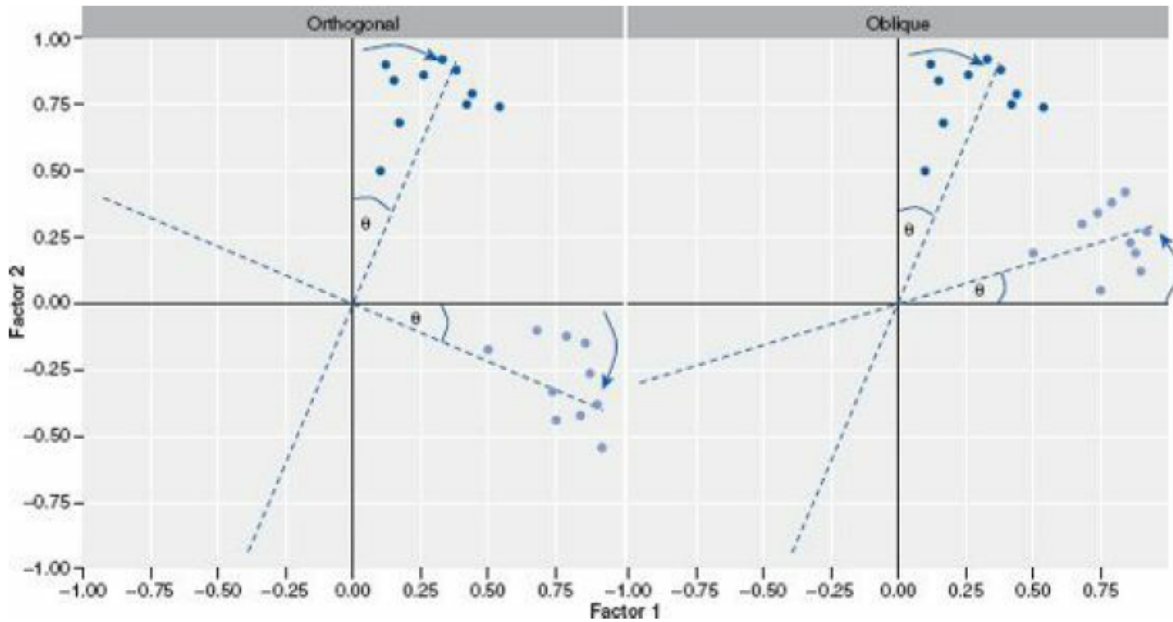


Figure 2: **Abbildung:** Screenshot aus Field (2012), S. 871, zur Erklärung von Faktorrotation. Auf der X-Achse ist die Korrelation mit Faktor 1 dargestellt, auf der Y-Achse die Korrelation mit Faktor zwei. Die Achsen werden so gedreht, dass die Korrelation mit *einem* der Faktoren maximiert, und mit allen anderen Faktoren minimiert wird.

1. Führen Sie eine erneute PCA durch, und geben Sie bei diesem Mal `rotate = "varimax"` für die Rotation ein. Dadurch wird eine orthogonale Rotation durchgeführt. Speichern Sie die Analyse unter dem Namen `pc3`
  - a) Lassen Sie sich den Output mit dem Befehl `print.psych()` anzeigen. Wenn Sie dabei `cut = 0.3` und `sort = TRUE` als Argumente angeben, werden nur Faktorladungen ab 0.3 angezeigt, und die Variablen werden nach der Höhe ihrer Faktorladungen sortiert.
2. Wiederholen Sie den Schritt aus 1., geben Sie bei diesem Mal aber `rotate = "oblimin"` an und speichern Sie die Analyse unter dem Namen `pc4`.
  - a) Lassen Sie sich erneut den Output mit `print.psych()` anzeigen.
  - b) Vergleichen Sie den Output kurz mit dem Output von `pc3`
  - c) Der Output von `pc4` enthält die Korrelationen der vier Faktoren. Können Sie diese im Output finden?
3. Sehen Sie sich nun die Fragen des RAQ an, die in `pc4` zu Faktoren zusammengefasst wurden.
  - a) Machen die Faktoren inhaltlich Sinn?
  - b) Wie würden Sie die Faktoren benennen?
  - c) Macht es inhaltlich Sinn, dass die Faktoren korrelieren dürfen? D.h. macht es Sinn, oblique Rotation zu verwenden?
  - d) Sind die Korrelationen zwischen den vier Faktoren inhaltlich sinnvoll?
4. *Factor Scores* enthalten die Werte einzelner Versuchspersonen für die extrahierten Faktoren. Extrahieren Sie die *Factor Scores* aus dem Output (`$scores`) und hängen Sie sie mit sinnvollen Variablennamen an den Rohdatensatz an. Geben Sie dem Objekt einen neuen Namen! **Achtung: Die Scores sind nur dann im Output-Objekt enthalten, wenn Sie den Rohdatensatz für die Analyse verwendet haben, nicht wenn Sie die Korrelationsmatrix verwendet haben.**
  - a) Haben Sie eine Idee, wie man in der Folge mit die *Factor Scores* verwenden könnte?

## Lösung

```
# Analyse durchführen
pc3 <- psych::principal(raq_data, nfactors = 4, rotate = "varimax")

# Output anzeigen
print.psych(pc3, cut = 0.3, sort = TRUE)
```

### Unteraufgabe 1

```
## Principal Components Analysis
## Call: psych::principal(r = raq_data, nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      item  RC3   RC1   RC4   RC2   h2   u2  com
## Q06    6  0.80          0.65 0.35 1.0
## Q18   18  0.68  0.33          0.60 0.40 1.5
## Q13   13  0.65          0.54 0.46 1.6
## Q07    7  0.64  0.33          0.55 0.45 1.7
## Q14   14  0.58  0.36          0.49 0.51 1.8
## Q10   10  0.55          0.33 0.67 1.2
## Q15   15  0.46          0.38 0.62 2.6
## Q20   20          0.68          0.48 0.52 1.1
## Q21   21          0.66          0.55 0.45 1.5
## Q03    3          -0.57          0.37 0.53 0.47 2.3
## Q12   12  0.47  0.52          0.51 0.49 2.1
## Q04    4  0.32  0.52  0.31          0.47 0.53 2.4
## Q16   16  0.33  0.51  0.31          0.49 0.51 2.6
## Q01    1          0.50  0.36          0.43 0.57 2.4
## Q05    5  0.32  0.43          0.34 0.66 2.5
## Q08    8          0.83          0.74 0.26 1.1
## Q17   17          0.75          0.68 0.32 1.5
## Q11   11          0.75          0.69 0.31 1.5
## Q09    9          0.65 0.48 0.52 1.3
## Q22   22          0.65 0.46 0.54 1.2
## Q23   23          0.59 0.41 0.59 1.4
## Q02    2          -0.34          0.54 0.41 0.59 1.7
## Q19   19          -0.37          0.43 0.34 0.66 2.2
##
##              RC3  RC1  RC4  RC2
## SS loadings      3.73 3.34 2.55 1.95
## Proportion Var   0.16 0.15 0.11 0.08
## Cumulative Var   0.16 0.31 0.42 0.50
## Proportion Explained 0.32 0.29 0.22 0.17
## Cumulative Proportion 0.32 0.61 0.83 1.00
##
## Mean item complexity = 1.8
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is 0.06
## with the empirical chi square 4006.15 with prob < 0
##
## Fit based upon off diagonal values = 0.96
```



```

# Analyse durchführen
pc4 <- psych::principal(raq_data, nfactors = 4, rotate = "oblimin")

# Output anzeigen
print.psych(pc4, cut = 0.3, sort = TRUE)

```

## Unteraufgabe 2

```

## Principal Components Analysis
## Call: psych::principal(r = raq_data, nfactors = 4, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      item  TC1   TC4   TC3   TC2   h2   u2 com
## Q06    6  0.87                0.65 0.35 1.1
## Q18   18  0.70                0.60 0.40 1.1
## Q07    7  0.64                0.55 0.45 1.2
## Q13   13  0.64                0.54 0.46 1.1
## Q10   10  0.57                0.33 0.67 1.2
## Q14   14  0.57                0.49 0.51 1.3
## Q12   12  0.45          0.43      0.51 0.49 2.0
## Q15   15  0.40                0.38 0.62 1.9
## Q08    8          0.90                0.74 0.26 1.0
## Q11   11          0.78                0.69 0.31 1.0
## Q17   17          0.78                0.68 0.32 1.0
## Q20   20                0.71      0.48 0.52 1.1
## Q21   21                0.60      0.55 0.45 1.3
## Q03    3                -0.51      0.53 0.47 1.8
## Q04    4                0.41      0.47 0.53 2.6
## Q16   16                0.41      0.49 0.51 2.4
## Q01    1          0.33 0.40      0.43 0.57 2.4
## Q05    5                0.34      0.34 0.66 2.7
## Q22   22                0.65 0.46 0.54 1.2
## Q09    9                0.63 0.48 0.52 1.4
## Q23   23                0.61 0.41 0.59 1.6
## Q02    2                -0.36 0.51 0.41 0.59 1.9
## Q19   19                -0.35 0.38 0.34 0.66 2.1
##
##              TC1  TC4  TC3  TC2
## SS loadings      3.90 2.88 2.94 1.85
## Proportion Var   0.17 0.13 0.13 0.08
## Cumulative Var   0.17 0.29 0.42 0.50
## Proportion Explained 0.34 0.25 0.25 0.16
## Cumulative Proportion 0.34 0.59 0.84 1.00
##
## With component correlations of
##      TC1  TC4  TC3  TC2
## TC1  1.00  0.44  0.36 -0.18
## TC4  0.44  1.00  0.31 -0.10
## TC3  0.36  0.31  1.00 -0.17
## TC2 -0.18 -0.10 -0.17  1.00
##
## Mean item complexity = 1.6
## Test of the hypothesis that 4 components are sufficient.

```

```
##
## The root mean square of the residuals (RMSR) is 0.06
## with the empirical chi square 4006.15 with prob < 0
##
## Fit based upon off diagonal values = 0.96
```

**Unteraufgabe 3** Die Faktoren scheinen inhaltlich Sinn zu ergeben. Field gibt Ihnen die Namen: 1. Fear of computers (TC1) 2. Fear of statistics (TC4) 3. Fear of mathematics (TC3) 4. Peer evaluation (TC2)

Da alle vier Faktoren Ängste beinhalten, ergibt es Sinn, dass sie miteinander korrelieren. Es ist daher auch inhaltlich sinnvoll, hier oblique Rotation einzusetzen.

Die relativ hohen Korrelationen zwischen TC1, TC4 und TC3 ergeben Sinn, da es sich um inhaltlich verwandte Konzepte handelt. Die geringeren Korrelationen mit TC2 erscheinen ebenfalls sinnvoll.

```
# Factor scores aus dem Modell extrahieren
scores <- pc4$scores

# Sinnvolle Spaltennamen vergeben
colnames(scores) <- c("fear_comp", "fear_stats", "fear_math", "fear_peer_eval")

# Scores und Rohdaten zusammenbinden
raq_data_scores <- cbind(raq_data, scores)
```

**Unteraufgabe 4** Die Faktorscores könnte man nun in der Folge z.B. als Prädiktoren in ein Regressionsmodell aufnehmen.

## 6) Unterschied PCA - EFA

Der wichtigste theoretische Unterschied wurde oben bereits kurz erwähnt. Wir möchten durch diese Aufgabe nun noch einen praktischen Unterschied in den Fokus stellen.

1. Führen Sie eine EFA mit dem Befehl `fa()` aus dem Paket `psych()` durch. Verwenden Sie die gleichen Argumente, wie bei `pc2`, also ohne Rotation. Die Funktionen sind gleich aufgebaut. Speichern Sie das Ergebnis unter dem Namen `fa1`.
2. Führen Sie eine zweite EFA durch, und extrahieren Sie diesmal 10 Faktoren (nach Joliffes Kriterium). Speichern Sie das Ergebnis unter dem Namen `fa2`.
3. Führen Sie eine weitere PCA durch, und extrahieren Sie diesmal ebenfalls 10 Faktoren ohne Rotation. Speichern Sie das Ergebnis unter dem Namen `pc5`.
4. Verwenden Sie die Funktion `head()`, um sich die Faktor Scores für die ersten sechs VP anzeigen zu lassen.
  - a) Vergleichen Sie die Faktor Scores der ersten vier Faktoren in der Berechnung von `fa1` und `fa2`
  - b) Vergleichen Sie die Faktor Scores der ersten vier Faktoren in der Berechnung von `pc2` und `pc5`
  - c) Was fällt dabei auf?

## Lösung

```
fa1 <- psych::fa(raq_data, nfactors = 4, rotate = "none")
```

### Unteraufgabe 1

```
fa2 <- psych::fa(raq_data, nfactors = 10, rotate = "none")
```

### Unteraufgabe 2

```
pc5 <- psych::principal(raq_data, nfactors = 10, rotate = "none")
```

### Unteraufgabe 3

```
# a)  
fa1$scores %>% head()
```

### Unteraufgabe 4

```
##           MR1           MR2           MR3           MR4  
## [1,]  1.12929066  1.0836217 -0.72346449  0.2981590  
## [2,]  0.57074542 -0.2376027  0.03663449 -0.3602815  
## [3,]  0.02563598  0.5075821  0.75049048  0.1906974  
## [4,] -0.57019904  0.7244243 -0.39525454 -0.4942844  
## [5,]  0.52745076 -0.2307131 -0.90621634  0.2706405  
## [6,] -0.76422989  0.5264649 -1.52869713  0.6392913
```

```
fa2$scores %>% head()
```

```
##           MR1           MR2           MR3           MR4           MR5           MR6           MR7           MR8  
## [1,]  1.12943990  1.1754210  1.5165187 -0.3294827  0.78782142  0.36858800 -0.1284385 -0.3053437  0.3  
## [2,]  0.51951224 -0.6122700 -0.7116855 -0.4328797 -0.10022246 -1.73018927 -0.7531633  0.8295174 -0.1  
## [3,]  0.04541655  0.9161946 -0.3185354  0.8132163  0.03427845  0.68250188  0.1819962 -0.3799708 -0.2  
## [4,] -0.50369767 -0.1195866 -0.5110000 -1.3275876  0.55824535 -0.33766887  0.7341396  0.0557121  1.4  
## [5,]  0.52633649 -0.5311493 -0.3801560 -0.9411793 -1.25620231 -0.03242136  0.5857954 -0.5896305 -0.0  
## [6,] -0.98203326  1.2499543  0.8345435 -0.4954983 -2.01377386 -1.27413668 -1.7633160  0.1322869 -0.5  
##           MR10  
## [1,]  0.230225683  
## [2,] -0.005372166  
## [3,]  0.585253089  
## [4,]  0.036917681  
## [5,] -0.017913949  
## [6,] -0.284332291
```

```
# b)
pc2$scores %>% head()
```

```
##           PC1           PC2           PC3           PC4
## [1,]  1.173758797  1.1108958 -1.2891595 -0.38759096
## [2,]  0.570078292 -0.1798303  0.3251963 -0.01212751
## [3,]  0.006509094  0.8039893  0.5184742  0.60730398
## [4,] -0.637584856  0.5108495 -0.4696227 -0.83510173
## [5,]  0.587687572 -0.3888630 -0.7160384 -0.43247530
## [6,] -0.881833937  0.4708284 -1.8625047 -0.14544207
```

```
pc5$scores %>% head()
```

```
##           PC1           PC2           PC3           PC4           PC5           PC6           PC7           PC8
## [1,]  1.173758797  1.1108958 -1.2891595 -0.38759096  1.2087498 -1.5574909  1.36766911  0.1323084  0.4
## [2,]  0.570078292 -0.1798303  0.3251963 -0.01212751 -0.9560550  0.4647447 -2.35726972 -1.1039647  0.9
## [3,]  0.006509094  0.8039893  0.5184742  0.60730398  0.3325874  1.9422485 -0.14897081  1.4778462 -0.1
## [4,] -0.637584856  0.5108495 -0.4696227 -0.83510173  0.8847922 -0.2008793 -1.80490370 -0.6602219  0.4
## [5,]  0.587687572 -0.3888630 -0.7160384 -0.43247530 -1.2150603 -0.6986948  0.44799485 -1.0063365 -0.2
## [6,] -0.881833937  0.4708284 -1.8625047 -0.14544207 -1.2126434 -0.4977185 -0.09233893  0.7480276 -1.9
##           PC10
## [1,] -0.5007619
## [2,] -2.0517548
## [3,]  0.6393485
## [4,]  1.0952371
## [5,]  0.8121233
## [6,] -2.0131043
```

Die Faktor Scores der ersten vier Faktoren sind bei `fa1` und `fa2` unterschiedlich, während sie bei `pc2` und `pc5` gleich bleiben.

## 7) Rendern

Lassen Sie die Datei mit `Strg + Shift + K` (Windows) oder `Cmd + Shift + K` (Mac) rendern. Sie sollten nun im “Viewer” unten rechts eine “schön aufpolierte” Version ihrer Datei sehen. Falls das klappt: Herzlichen Glückwunsch! Ihr Code kann vollständig ohne Fehlermeldung gerendert werden. Falls nicht: Nur mut, das wird schon noch! Gehen Sie auf Fehlersuche! Ansonsten schaffen wir es ja in der Übung vielleicht gemeinsam.

## Literatur

*Anmerkung:* Diese Übungszettel basieren zum Teil auf Aufgaben aus dem Lehrbuch *Discovering Statistics Using R* (Field, Miles & Field, 2012). Sie wurden für den Zweck dieser Übung modifiziert, und der verwendete R-Code wurde aktualisiert.

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. London: SAGE Publications Ltd.

## English

## Links

[Exercise sheet as PDF](#)

### Exercise sheet with solutions included

[Exercise sheet with solutions included as PDF](#)

The source code of this sheet as `.Rmd` (Right click and “save as” to download ...)

## Some hints

1. Please try to solve this sheet in an `.Rmd` file. You can create one from scratch using `File > New file > R Markdown...`. You can delete the text beneath *Setup Chunk* (starting from line 11). Alternatively, you can download our template file under [this link](#) (right click > save as...).
2. You'll find a lot of the important information on the [website of this course](#)
3. Please don't hesitate to search the web for help with this sheet. In fact, being able to effectively search the web for problem solutions is a very useful skill, even R pros work this way all the time! The best starting point for this is the [R section on the programming site Stackoverflow](#)
4. On the R Studio website, you'll find highly helpful [cheat sheets](#) for many of R topics. The [base R cheat sheet](#) might be a good starting point.

## Ressources

Since this is a hands-on seminar, we won't be able to present each and every new command to you explicitly. Instead, you'll find here references to helpful resources that you can use for completing this sheets.

Ressource	Description
Field, chapter 17	Book chapter explaining step by step the why and how of principal component analysis in R.

## Highly recommended!

## Hint of the week

There is a package named `conflicted` that automatically shows error messages when you use a function, that exists in two or more packages. So your attention is drawn to errors of that type. The only thing, you have to do, is to install this package (`install.packages("conflicted")`) and load it in your script (`require(conflicted)`).

### Example

```
library(conflicted)
library(dplyr)
```

```
filter(mtcars, am & cyl == 8)
```

```
Fehler: filter found in 2 packages. You must indicate which one you want with ::
* dplyr::filter
* stats::filter
```

## 1) Read data, some general remarks

1. Load the relevant packages, among these `psych` and set an adequate working directory.
2. Load the data set `raq.dat` from the link <https://pzezula.pages.gwdg.de/data/raq.dat> and call the data object `raq_data`.
3. Each line has the answers of one subject.
4. Read the sections “Overview of the items” and “What is it about?” to get into the the topic.

### Solutions

```
# load packages
library(tidyverse)
library(psych)
```

```
# set working directory
# put your directory here
setwd("P:/mv")
```

### Subtask 1

**Subtask 2** See the instructions.

```
# this would be the syntax for locally stored data ...
# raq_data <- read_delim("data/raq.dat", delim = "\t")
# or
raq_data <- read_delim("https://pzezula.pages.gwdg.de/data/raq.dat", delim = "\t")
```

### Subtask 3

```
##
## -- Column specification -----
## cols(
##   .default = col_double()
## )
## i Use `spec()` for the full column specifications.
```

```
raq_data <- read_delim("http://md.psych.bio.uni-goettingen.de/mv/data/af/raq.dat", delim = "\t")
```

```
##
## -- Column specification -----
## cols(
##   .default = col_double()
## )
## i Use `spec()` for the full column specifications.
```

**Subtask 4** See the instructions.

**A look at the items**

The data show answers to a fictive questionnaire, the "R Anxiety Questionnaire (RAQ) with items for fear of statistics (Field, p. 873). The items are:

SD = Strongly Disagree, D = Disagree, N = Neither, A = Agree, SA = Strongly Agree		SD	D	N	A	SA
1	Statistics make me cry	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	My friends will think I'm stupid for not being able to cope with R	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	Standard deviations excite me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	I dream that Pearson is attacking me with correlation coefficients	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	I don't understand statistics	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	I have little experience of computers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	All computers hate me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	I have never been good at mathematics	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	My friends are better at statistics than me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Computers are useful only for playing games	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	I did badly at mathematics at school	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	People try to tell you that R makes statistics easier to understand but it doesn't	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	I worry that I will cause irreparable damage because of my incompetence with computers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	Computers have minds of their own and deliberately go wrong whenever I use them	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	Computers are out to get me	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	I weep openly at the mention of central tendency	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
17	I slip into a coma whenever I see an equation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
18	R always crashes when I try to use it	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
19	Everybody looks at me when I use R	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
20	I can't sleep for thoughts of eigenvectors	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
21	I wake up under my duvet thinking that I am trapped under a normal distribution	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
22	My friends are better at R than I am	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
23	If I am good at statistics people will think I am a nerd	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**What is it about?**

This exercise sheet is about explorative factor analysis (EFA) and principal component analysis (PCA).

Generally spoken, both are ways to aggregate variables to groups. This helps us to treat multicollinearity in multiple regression, when predictors correlate too high. It is also useful to refer to latent constructs (f. e. traits) on base of indicator variables. Moreover, we can use confirmatory factor analyses (CFA) to test theories that were set up on base of f. e. an EFA.

Here we only use the first two types, that is **Principal Components Analysis (PCA)** and **Exploratory Factor Analysis (EFA)**. Both are quite similar, indeed, we can look at PCA to be a special case of EFA.

**Table:** Overview on Types of Factor Analysis

Analysis	Meaning	Application
Principal Components Analysis (PCA)	Variables, that share information, are aggregated	f. e. to deal with multicollinearity in regression analysis
Explorative Factor Analysis (EFA)	Variables, that base on the same hidden characteristic, are aggregated	f. e. construction of questionnaires
Confirmatory Factor Analysis (CFA)	Variables, that base on the same characteristic of f. e. a person, are aggregated.	f. e. tests of theories of factor structures.

## 2) A first look at the data

As with other analyses we have to check first, whether our data are suitable for factor analysis. This means, whether we have sufficient but not too strong relations between the variables. The analysis won't work, if two variables are practically identical ( $\text{cor} > .90$ ).

Primarily we inspect the data manually, then we make a test and look at two indices, that give us informations whether the data are suitable for factor analysis.

Some explanations (see Field (2012) for more information):

Criterion	Meaning	Application
correlation matrix	Are there suitable correlations between our variables?	We have problems, if 1) we almost don't have correlations above 0.30. 2) We have (a lot of) correlations above .90. 3) Some of our variables have only very low correlations with all the others.
Bartlett's Test	Is our correlation matrix a singular matrix?	If the test is significant, we <b>don't</b> have singularity. This is, what factor analyses need. A non significant test would indicate problems.
Kaiser-Meyer Olkin measure (KMO)	Is there something systematic, that can be detected by applying a factor analysis. There is an overall index and an index for every variable. We should look at both.	
Determinant of the correlation matrix	Are the relations in our correlation matrix strong (0) or weak (1)?	The determinant should be small, but not smaller than 0.00001, then a factor analysis is possible.

1. Use the command `cor()` to get a correlation matrix of `raq_data`. *Tip: You can use `round()` to round numbers and make it more readable.*



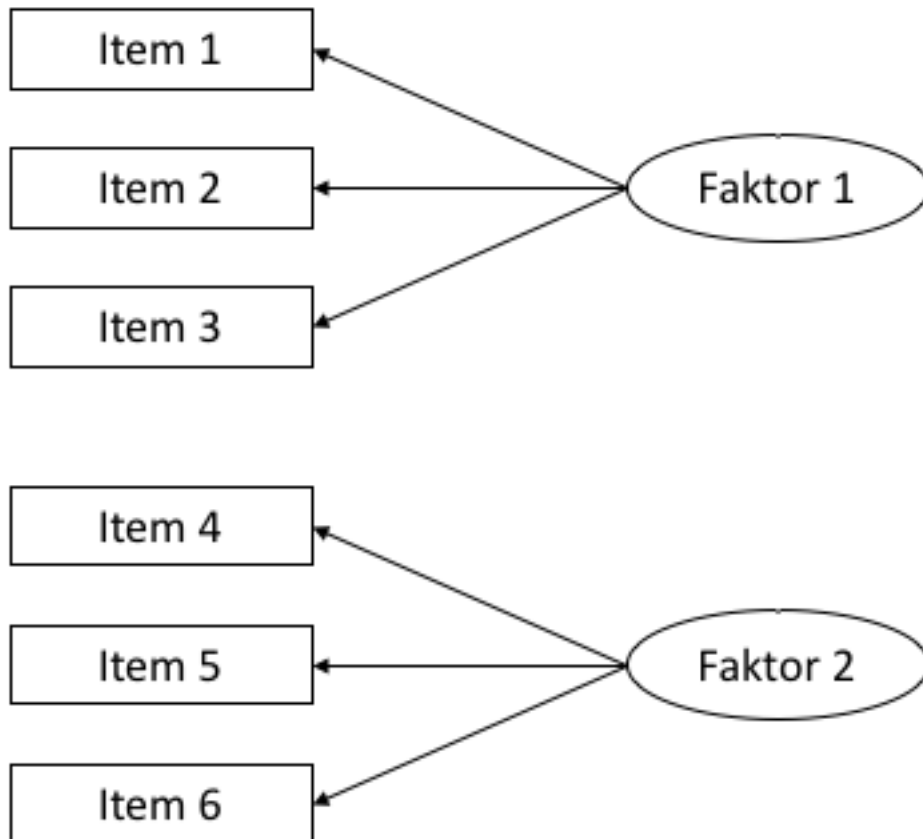


Figure 3: **Figure:** Example for a path diagram. Such diagrams are frequently used to visualize factor analyses. In this case we see the *manifest* variables, i. e. the ones, we can observe directly, here the answers of subjects to items 1 to 3. They are related to the *latent* factor 1, that we cannot observe directly. The same holds for items 4 to 6 and their relation to factor 2.

2. Inspect the correlation matrix. Do you find anything that could cause problems running a factor analysis?
3. Run the command `cor.test.bartlett()` of package `psych` on our data. Run it on base of the data matrix or based on a correlation matrix. Both ways should result in the same, if the variables contained are the same. Do you see any problem applying factor analysis? *a hint:* If your base is a correlation matrix, you have to specify argument `n =` because your sample size is needed.
4. Apply `KMO()` of package `psych` to the data or to the correlation matrix. Both ways should result in the same, if the variables contained are the same. Do you see any problem applying factor analysis?
5. Apply `det()` to the **correlation matrix**. Do you see any problem applying factor analysis?

## Solutions

```
raq_matrix <- raq_data %>% cor() %>% round(2)
```

### Subtask 1

### Subtask 2

- a) We find correlations above .30
- b) We don't have correlations above .90
- c) There are no variables, that call our attention, because they have only very low correlations with other variables.

```
# applying test to correlation matrix
psych::cor.test.bartlett(raq_matrix, n = 2571)
```

### Subtask 3

```
## $chisq
## [1] 19323.64
##
## $p.value
## [1] 0
##
## $df
## [1] 253
```

```
# apply test to dataset
psych::cor.test.bartlett(raq_data)
```

```
## R was not square, finding R from data
```

```
## $chisq
## [1] 19334.49
##
## $p.value
```

```
## [1] 0
##
## $df
## [1] 253
```

The test is significant. There are promising in the data.

```
# we apply the function to the dataset
kmo_raq <- psych::KMO(raq_data)

# ... and look at the results
kmo_raq
```

#### Subtask 4

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: psych::KMO(r = raq_data)
## Overall MSA = 0.93
## MSA for each item =
## Q01 Q02 Q03 Q04 Q05 Q06 Q07 Q08 Q09 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20
## 0.93 0.87 0.95 0.96 0.96 0.89 0.94 0.87 0.83 0.95 0.91 0.95 0.95 0.97 0.94 0.93 0.93 0.95 0.94 0.89
## Q23
## 0.77
```

```
# we look at the values ordered increasingly
kmo_raq$MSAi %>% sort()
```

```
##      Q23      Q09      Q08      Q02      Q22      Q20      Q06      Q11      Q21      Q01
## 0.7663994 0.8337295 0.8713055 0.8747754 0.8784508 0.8890514 0.8913314 0.9059338 0.9293369 0.9297610
##      Q16      Q15      Q19      Q07      Q18      Q13      Q10      Q03      Q12      Q04
## 0.9336439 0.9404402 0.9407021 0.9416800 0.9479508 0.9482270 0.9486858 0.9510378 0.9548324 0.9553403
##      Q14
## 0.9671722
```

The overall index is 0.93 and the lowest single value is .77, which is still quite good. Thus our indices show a high suitability for applying factor analysis to the data.

```
# the determinant of the correlation matrix
det(raq_matrix)
```

#### Subtask 5

```
## [1] 0.0005293425
```

The determinant is small. That indicates strong relations between the variables. But it's bigger than the recommended cutoff value of 0.00001. This means, the relations are not too strong in our data. So we can conduct a factor analysis.

### 3) Factor identification

When doing explorative analyses we don't know anything about the number of factors in our data. We can use *scree plots* and *eigenvalues* of a first analysis to decide about that.

\*\* We work here with PCA. With EFA the procedure is almost identical. We will announce differences. An exercise below is dedicated to the differences in results.\*\*

The general command for running a PCA in R is supplied by package `psych`: `principal(r = , nfactors = , rotate = )`. `r` refers to the correlation matrix or to the raw data matrix, `nfactors` is the number of factors, that we want to extract. `rotate` defines the type of rotation. We talk about rotation later.

The structure of the command for running an EFA is the same: `fa(r = , nfactors = , rotate = )`.

**Table 3.1:** Important components in the output of PCA/EFA.

Component	Where in the output?	Meaning
factor loadings	Big table up in the output	Correlations of the variables and the factors.
h2	At the right of the table with factor loadings	<i>Communality</i> : Measure for the percentage of variance, that can be explained by the factors of the model.
u2	right next to h2	<i>Uniqueness</i> : Unique part of the variance of the variable in question. It's 1 - h2
Eigenvalues (SS loadings)	table below the factor loadings, first line	sum of the squared factor loadings per factor. It is shown as percentage of explained variance. In our case with 23 variables an eigenvalue of 7.29 (PC1) means $7.29 / 23 = 0.32$ or 32% of the variance in the data. We can find that value in the second line of the table.

**Table 3.2:** Criteria to define an adequate number of factors to extract.

Criterion	Resulting number of factors to extract
Kaiser	Factors with eigenvalues above 1 presumably a good criterion if less than 30 variables are examined and the communalities <i>after</i> factor extraction are higher than 0.7. With sample sizes above 250 communalities of 0.6 are acceptable. With bigger samples we might accept even smaller communalities.
Jolliffe	Factors with eigenvalues above 0.7 in Jolliffe's opinion Kaiser's criterion was too conservative.
Scree Plot	Shows the importance of all factors of the first analysis. We look for a <i>point of inflexion</i> : A point, where the inclination of the resulting line changes rapidly. We can often detect a point, where the curve enters a sort of horizontal course after having started almost vertical. All Factors left of this point point of inflexion define the number of factors to extract.

1. Conduct a PCA on `raq_data` and store it under the name of `pc1`. We want to have all possible factors included in the first analysis, this means a total of 23. Please do not rotate, set `rotate = "none"`.

i) **Attention, difference to EFA:** In an EFA we do not extract as many factors as our variables allow, we have to set a smaller number. In this case, we could set it to 18 f. e..

2. Look at the scree-plot and at the eigenvalues in the output of PCA. Decide on base of table 3.2 how

many factors you want to extract.

3. Run the PCA again, but this time we want to extract four factors. We set `nfactors = 4` to do that. Look at the *communalities*. Is the extraction of four factors correct if we refer to Kaiser Criterion?

a) You may have a look at the mean of the *communalities* to decide ...

4. A further and **very recommendable method** to decide about the number of factors in PCA is the parallel-analysis. Random data are generated repeatedly and compared to the real data. This way the decision is not as subjective as the above mentioned possibilities. Package `psych` offers an easy way to run it: `fa.parallel(<data>, fa = <"method">)`. Put in our data instead of `<data>` and the method you want for `<"method">`. For PCA you have to set "pc" and for FA "fa". You may also put "both". Conduct a parallel-analysis on our dataset. What is your conclusion? (Find more on that under: <http://md.psych.bio.uni-goettingen.de/mv/unit/fa/fa.html#parallelanalyse> )
5. Look at the last line of the output. There you can find a further index for the fit of the model to the data. This index varies between 0 and 1. The closer to 1, the better the fit of model and data. Values above 0.95 are considered to indicate a good fit. Find a more in depth explanation for that in Field (2012, pp 889 - 891).

## Solutions

```
# Analysis
pc1 <- psych::principal(raq_matrix, nfactors = ncol(raq_matrix), rotate = "none")
```

```
# Output
pc1
```

## Subtask 1

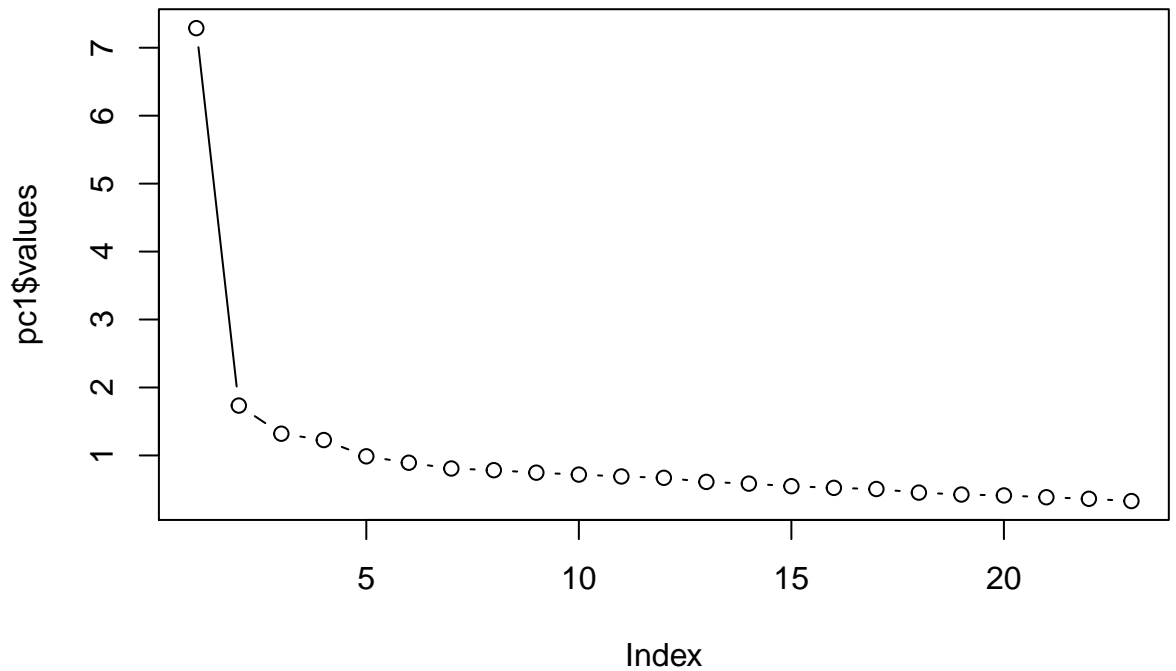
```
## Principal Components Analysis
## Call: psych::principal(r = raq_matrix, nfactors = ncol(raq_matrix),
##   rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9  PC10  PC11  PC12  PC13  PC14  PC15  PC16
## Q01  0.59  0.18 -0.21  0.13 -0.39 -0.13 -0.21 -0.10 -0.02 -0.05  0.12 -0.11  0.27 -0.29  0.18 -0.11
## Q02 -0.30  0.55  0.16  0.02 -0.01 -0.37  0.26 -0.38  0.01 -0.07  0.31  0.26  0.00  0.03 -0.24  0.03
## Q03 -0.63  0.29  0.21 -0.06  0.03  0.02  0.04 -0.06  0.21  0.11  0.12  0.03  0.10  0.08  0.40  0.09
## Q04  0.63  0.15 -0.14  0.15 -0.21 -0.14 -0.06  0.09 -0.07 -0.01 -0.03  0.33 -0.35 -0.14  0.09 -0.31
## Q05  0.56  0.10 -0.07  0.14 -0.41 -0.18 -0.05  0.09  0.26  0.03 -0.29  0.16  0.19  0.46 -0.03  0.10
## Q06  0.56  0.09  0.57 -0.04  0.18  0.02 -0.01  0.04  0.00 -0.01 -0.13  0.20  0.22 -0.07  0.11 -0.20
## Q07  0.68  0.04  0.25  0.11  0.17 -0.07  0.03  0.05 -0.08  0.12 -0.28  0.20 -0.01 -0.21 -0.02  0.25
## Q08  0.55  0.40 -0.32 -0.42  0.13  0.10 -0.06 -0.06  0.01 -0.04 -0.09  0.03 -0.02  0.04 -0.04 -0.03
## Q09 -0.28  0.63 -0.01  0.10  0.18 -0.25  0.00 -0.01  0.16  0.28 -0.26 -0.38 -0.17 -0.05  0.11 -0.09
## Q10  0.44  0.04  0.37 -0.09 -0.35  0.22  0.44  0.02  0.32 -0.26 -0.10 -0.23 -0.14 -0.15 -0.11 -0.03
## Q11  0.65  0.25 -0.20 -0.41  0.12  0.18 -0.02  0.02  0.08 -0.16  0.02  0.03  0.03  0.03 -0.01 -0.09
## Q12  0.67 -0.05  0.04  0.24  0.04 -0.09 -0.13  0.05  0.05 -0.13  0.21 -0.08 -0.43  0.21  0.10  0.09
## Q13  0.67  0.07  0.29  0.00  0.14  0.04 -0.22  0.00  0.08 -0.20  0.26 -0.08  0.01  0.09  0.15  0.13
## Q14  0.66  0.03  0.19  0.14  0.09 -0.04 -0.09 -0.04 -0.14  0.19  0.07 -0.27  0.11  0.19 -0.34 -0.25
## Q15  0.59  0.01  0.12 -0.11 -0.09  0.30  0.33 -0.06 -0.25  0.42  0.10  0.09 -0.10  0.18  0.18 -0.04
```

```

## Q16  0.68  0.01 -0.14  0.08 -0.32 -0.01  0.13 -0.11 -0.18  0.19  0.15 -0.18  0.10 -0.10  0.04  0.21
## Q17  0.64  0.33 -0.20 -0.35  0.09  0.06 -0.04  0.03 -0.04  0.01  0.02 -0.03 -0.01  0.00 -0.05  0.19
## Q18  0.70  0.03  0.29  0.13  0.17 -0.09 -0.11  0.05 -0.07 -0.11  0.07 -0.11  0.09 -0.03  0.02  0.00
## Q19 -0.43  0.39  0.09 -0.02 -0.14  0.06 -0.03  0.69  0.06  0.16  0.30  0.07  0.05 -0.07 -0.14  0.01
## Q20  0.44 -0.20 -0.42  0.28  0.34  0.02  0.34  0.05  0.32  0.03  0.18  0.03  0.18  0.03  0.10 -0.20
## Q21  0.66 -0.05 -0.20  0.27  0.25 -0.13  0.18  0.13  0.12  0.06 -0.02  0.05  0.01 -0.16 -0.09  0.24
## Q22 -0.30  0.47 -0.13  0.37  0.06  0.11  0.26  0.16 -0.45 -0.39 -0.14 -0.07  0.10  0.14  0.11  0.00
## Q23 -0.14  0.37 -0.03  0.51  0.00  0.62 -0.26 -0.23  0.17  0.08 -0.01  0.12 -0.02 -0.06 -0.14  0.05
##      PC19 PC20 PC21 PC22 PC23 h2      u2 com
## Q01  0.16  0.03 -0.21  0.02  0.00  1 0.0e+00 6.0
## Q02  0.02  0.00 -0.02  0.02  0.02  1 6.7e-16 6.3
## Q03  0.07  0.05  0.00 -0.03  0.04  1 1.2e-15 4.3
## Q04 -0.22  0.01  0.07 -0.01  0.02  1 1.1e-16 4.9
## Q05 -0.04 -0.02 -0.02  0.02  0.02  1 1.0e-15 5.4
## Q06  0.09 -0.08  0.10 -0.28 -0.15  1 1.0e-15 4.5
## Q07  0.16  0.24  0.10  0.22  0.12  1 1.2e-15 4.1
## Q08  0.14 -0.11  0.04  0.23 -0.33  1 1.1e-15 5.7
## Q09 -0.07 -0.05  0.03  0.00  0.04  1 5.6e-16 4.9
## Q10 -0.01  0.04 -0.03  0.02 -0.04  1 3.3e-16 7.8
## Q11  0.13 -0.15  0.09 -0.04  0.40  1 6.7e-16 4.1
## Q12  0.34  0.06 -0.01 -0.11 -0.05  1 1.1e-16 3.9
## Q13 -0.31 -0.09  0.13  0.13 -0.02  1 0.0e+00 4.2
## Q14  0.04  0.06  0.03 -0.02  0.03  1 7.8e-16 4.4
## Q15 -0.02 -0.07 -0.21  0.08  0.03  1 6.7e-16 5.5
## Q16 -0.02 -0.06  0.37 -0.06 -0.03  1 0.0e+00 4.0
## Q17 -0.19  0.35 -0.14 -0.28 -0.04  1 6.7e-16 4.5
## Q18 -0.09  0.01 -0.22  0.19  0.05  1 5.6e-16 3.3
## Q19  0.05  0.00  0.03  0.05 -0.02  1 2.2e-16 3.4
## Q20 -0.02  0.20  0.09  0.07 -0.04  1 6.7e-16 8.5
## Q21 -0.03 -0.35 -0.17 -0.14 -0.02  1 1.0e-15 4.5
## Q22  0.00  0.01  0.01 -0.02  0.00  1 3.3e-16 6.9
## Q23 -0.01  0.00  0.00 -0.02  0.00  1 5.6e-16 4.1
##
##
##      PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10 PC11 PC12 PC13 PC14 PC15 PC16
## SS loadings      7.29 1.73 1.32 1.23 0.99 0.89 0.81 0.78 0.75 0.72 0.69 0.67 0.61 0.58 0.55 0.53
## Proportion Var   0.32 0.08 0.06 0.05 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.02 0.02
## Cumulative Var   0.32 0.39 0.45 0.50 0.55 0.58 0.62 0.65 0.69 0.72 0.75 0.78 0.80 0.83 0.85 0.88
## Proportion Explained 0.32 0.08 0.06 0.05 0.04 0.04 0.04 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.02 0.02
## Cumulative Proportion 0.32 0.39 0.45 0.50 0.55 0.58 0.62 0.65 0.69 0.72 0.75 0.78 0.80 0.83 0.85 0.88
##
##      PC19 PC20 PC21 PC22 PC23
## SS loadings      0.42 0.41 0.38 0.36 0.33
## Proportion Var   0.02 0.02 0.02 0.02 0.01
## Cumulative Var   0.94 0.95 0.97 0.99 1.00
## Proportion Explained 0.02 0.02 0.02 0.02 0.01
## Cumulative Proportion 0.94 0.95 0.97 0.99 1.00
##
## Mean item complexity = 5
## Test of the hypothesis that 23 components are sufficient.
##
## The root mean square of the residuals (RMSR) is 0
##
## Fit based upon off diagonal values = 1

```

```
# without pipe
plot(pc1$values, type = "b")
```



### Subtask 2

```
# with pipe
# pc1$values %>% plot(type = "b")
```

`pc1$values` accesses object values in the output of our PCA. `type = "b"` means, that points and lines should be plotted (**both**).

**Subtask 3** The scree plot gets to a more or less horizontal form quite fast, especially after the 4th factor. Moreover we have four eigenvalues above 1, what would be congruent with the Kaiser Criterion. Jolliffe's Criterion would suggest to extract 10 factors on the other hand. We would keep on going with four factors at this point, because of the congruence of the scree plot and Kaiser Criterion. There is no objective decision but we can evaluate it by looking at the *communalities*.

Looking at the scree plot we could also have the impression of two factors being adequate. Don't hesitate to try that also.

```
# Analysis with four factors
pc2 <- psych::principal(raq_data, nfactors = 4, rotate = "none")
```

```
# Output
pc2
```

#### Subtask 4

```
## Principal Components Analysis
## Call: psych::principal(r = raq_data, nfactors = 4, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1  PC2  PC3  PC4  h2  u2 com
## Q01  0.59  0.18 -0.22  0.12  0.43  0.57  1.6
## Q02 -0.30  0.55  0.15  0.01  0.41  0.59  1.7
## Q03 -0.63  0.29  0.21 -0.07  0.53  0.47  1.7
## Q04  0.63  0.14 -0.15  0.15  0.47  0.53  1.3
## Q05  0.56  0.10 -0.07  0.14  0.34  0.66  1.2
## Q06  0.56  0.10  0.57 -0.05  0.65  0.35  2.1
## Q07  0.69  0.04  0.25  0.10  0.55  0.45  1.3
## Q08  0.55  0.40 -0.32 -0.42  0.74  0.26  3.5
## Q09 -0.28  0.63 -0.01  0.10  0.48  0.52  1.5
## Q10  0.44  0.03  0.36 -0.10  0.33  0.67  2.1
## Q11  0.65  0.25 -0.21 -0.40  0.69  0.31  2.2
## Q12  0.67 -0.05  0.05  0.25  0.51  0.49  1.3
## Q13  0.67  0.08  0.28 -0.01  0.54  0.46  1.4
## Q14  0.66  0.02  0.20  0.14  0.49  0.51  1.3
## Q15  0.59  0.01  0.12 -0.11  0.38  0.62  1.2
## Q16  0.68  0.01 -0.14  0.08  0.49  0.51  1.1
## Q17  0.64  0.33 -0.21 -0.34  0.68  0.32  2.4
## Q18  0.70  0.03  0.30  0.13  0.60  0.40  1.4
## Q19 -0.43  0.39  0.10 -0.01  0.34  0.66  2.1
## Q20  0.44 -0.21 -0.40  0.30  0.48  0.52  3.2
## Q21  0.66 -0.06 -0.19  0.28  0.55  0.45  1.6
## Q22 -0.30  0.47 -0.12  0.38  0.46  0.54  2.8
## Q23 -0.14  0.37 -0.02  0.51  0.41  0.59  2.0
##
##              PC1  PC2  PC3  PC4
## SS loadings      7.29  1.74  1.32  1.23
## Proportion Var   0.32  0.08  0.06  0.05
## Cumulative Var   0.32  0.39  0.45  0.50
## Proportion Explained 0.63  0.15  0.11  0.11
## Cumulative Proportion 0.63  0.78  0.89  1.00
##
## Mean item complexity = 1.8
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is 0.06
## with the empirical chi square 4006.15 with prob < 0
##
## Fit based upon off diagonal values = 0.96

## mean communality
pc2$communality %>% mean()

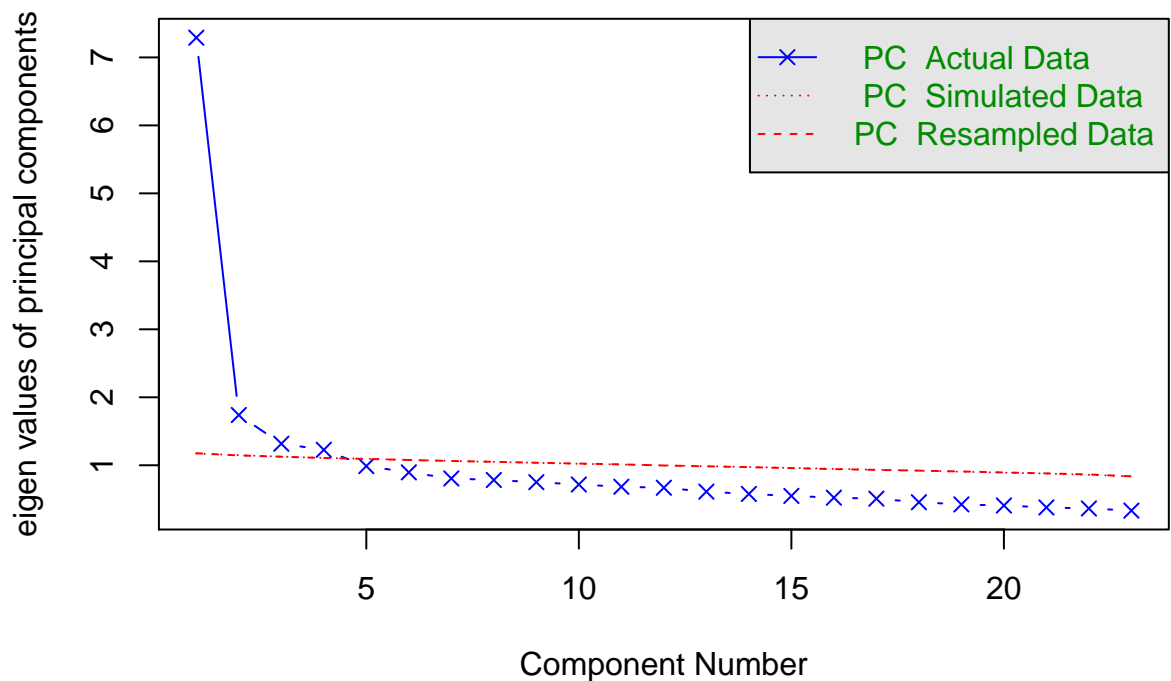
## [1] 0.5031663
```



The mean communality of .503 is below .60, which the recommended threshold of Kaiser Criterion for samples above 250 subjects. Field (2012) considers this value acceptable, because our sample is way bigger than that (n=2571). So our cut off values will presumably passed in this case. We can also interpret our scree plot as an hint to extract 4 factors.

```
items_parallel <- psych::fa.parallel(raq_data, fa = "pc")
```

### Parallel Analysis Scree Plots



#### Subtask 5

## Parallel analysis suggests that the number of factors = NA and the number of components = 4

For PCA also our parallel analysis recommends the extraction of four *components*.

**Subtask 6** Our model has an fit index of 0.96, which is considered a good fit.

### 5) Rotation and scores

Factor rotation is an important step in factor analysis, that allows us to separate factors more clearly from each other. This works by a sort of multidimensional rotation of the coordinate system of the factors. There are two types of rotation:

Rotation	Meaning
Orthogonal	Factors are rotated, but stay independent (orthogonal) from each other. Their intercorrelation is 0.
Oblique	Faktors are rotated and correlations between the rotated factors are allowed.

Find more about this in Field (2012) chapter 17.3.9

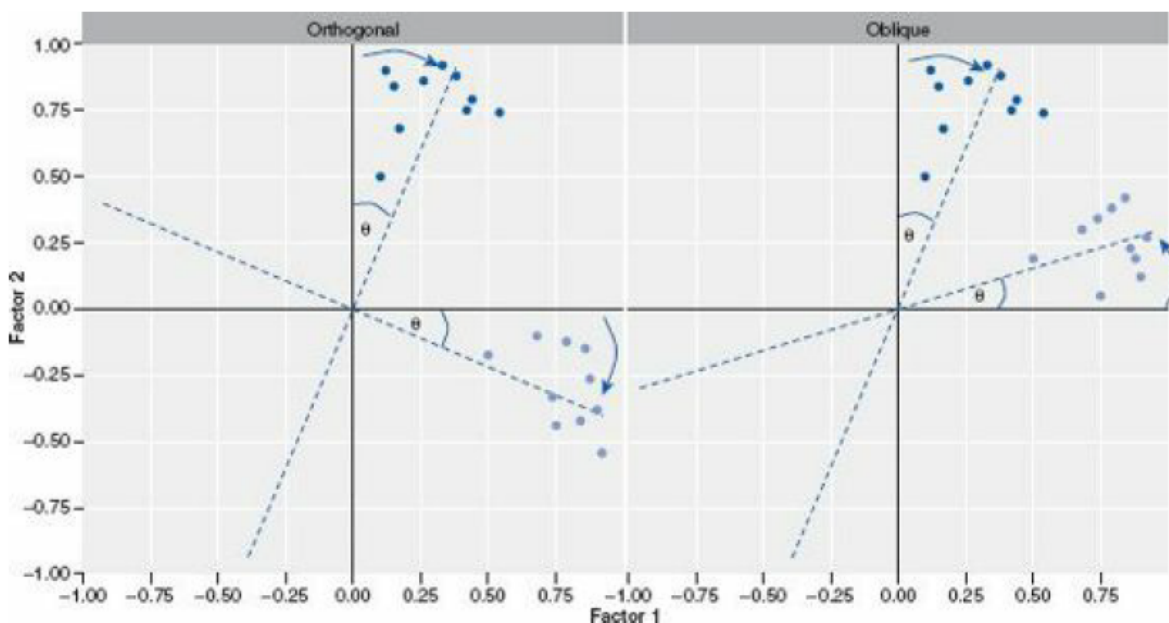


Figure 4: **Figure:** Screenshot from Field (2012), p. 871, to explain factor rotation. You see the correlation with factor 1 on the x-axis and the correlation with factor 2 on the y-axis. The axes are rotated to maximise the correlation with *one* of the factors and at the same time to minimize the correlations with all the other factors

1. Make a new PCA and use `rotate = "varimax"` to get factor rotation. This will make an orthogonal rotation. Store the result and name it `pc3`.
  - a) Generate the output via `print.psych()`. If you add the arguments `cut = 0.3` and `sort = TRUE`, only factor loadings above 0.3 are shown and the factors are sorted by factor loading.
2. Repeat the steps of 1. but add `rotate = "oblimin"` this time, store the result and name it `pc4`.
  - a) Again, get the output using `print.psych()`.
  - b) compare the new output with `pc3`.
  - c) Find the correlations of the four factors in the output of `pc4`.
3. Take a look at the questions of RAQ and how they were clustered in `pc4` into factors.
  - a) Does the combination of items to factors make sense to you? Can you find a common content in the combined questions?
  - b) What would you call the factors? Name them.
  - c) Makes it sense to let the factors correlate? In other words, does oblique rotation make sense?
  - d) If you think of the content they have: Do the correlations found between the four factors make sense?

4. *Factor Scores* indicate the level of our observations in our factors. Extract the *Factor Scores* from our output (`$scores`) and add them to our data object giving them adequate names. Give a meaningful name to our newly structured data object. **\*\*Take care:** We can only find our factor scores in our output object, when we used raw data to do the factor analysis. If we started using a correlation matrix, they are not computed.

a) Do you have any idea what we could use factor scores for?

## Solutions

```
# do the analysis
pc3 <- psych::principal(raq_data, nfactors = 4, rotate = "varimax")

# show the output
print.psych(pc3, cut = 0.3, sort = TRUE)
```

### Subtask 1

```
## Principal Components Analysis
## Call: psych::principal(r = raq_data, nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      item  RC3  RC1  RC4  RC2  h2  u2 com
## Q06    6  0.80          0.65 0.35 1.0
## Q18   18  0.68  0.33          0.60 0.40 1.5
## Q13   13  0.65          0.54 0.46 1.6
## Q07    7  0.64  0.33          0.55 0.45 1.7
## Q14   14  0.58  0.36          0.49 0.51 1.8
## Q10   10  0.55          0.33 0.67 1.2
## Q15   15  0.46          0.38 0.62 2.6
## Q20   20          0.68          0.48 0.52 1.1
## Q21   21          0.66          0.55 0.45 1.5
## Q03    3          -0.57          0.37 0.53 0.47 2.3
## Q12   12  0.47  0.52          0.51 0.49 2.1
## Q04    4  0.32  0.52  0.31          0.47 0.53 2.4
## Q16   16  0.33  0.51  0.31          0.49 0.51 2.6
## Q01    1          0.50  0.36          0.43 0.57 2.4
## Q05    5  0.32  0.43          0.34 0.66 2.5
## Q08    8          0.83          0.74 0.26 1.1
## Q17   17          0.75          0.68 0.32 1.5
## Q11   11          0.75          0.69 0.31 1.5
## Q09    9          0.65 0.48 0.52 1.3
## Q22   22          0.65 0.46 0.54 1.2
## Q23   23          0.59 0.41 0.59 1.4
## Q02    2          -0.34          0.54 0.41 0.59 1.7
## Q19   19          -0.37          0.43 0.34 0.66 2.2
##
##              RC3  RC1  RC4  RC2
## SS loadings      3.73 3.34 2.55 1.95
## Proportion Var   0.16 0.15 0.11 0.08
## Cumulative Var   0.16 0.31 0.42 0.50
## Proportion Explained 0.32 0.29 0.22 0.17
```

```

## Cumulative Proportion 0.32 0.61 0.83 1.00
##
## Mean item complexity = 1.8
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is 0.06
## with the empirical chi square 4006.15 with prob < 0
##
## Fit based upon off diagonal values = 0.96

```

```

# do the analysis
pc4 <- psych::principal(raq_data, nfactors = 4, rotate = "oblimin")

# show the output
print.psych(pc4, cut = 0.3, sort = TRUE)

```

## Subtask 2

```

## Principal Components Analysis
## Call: psych::principal(r = raq_data, nfactors = 4, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      item  TC1  TC4  TC3  TC2  h2  u2  com
## Q06    6  0.87                0.65 0.35 1.1
## Q18   18  0.70                0.60 0.40 1.1
## Q07    7  0.64                0.55 0.45 1.2
## Q13   13  0.64                0.54 0.46 1.1
## Q10   10  0.57                0.33 0.67 1.2
## Q14   14  0.57                0.49 0.51 1.3
## Q12   12  0.45          0.43    0.51 0.49 2.0
## Q15   15  0.40                0.38 0.62 1.9
## Q08    8          0.90                0.74 0.26 1.0
## Q11   11          0.78                0.69 0.31 1.0
## Q17   17          0.78                0.68 0.32 1.0
## Q20   20                0.71    0.48 0.52 1.1
## Q21   21                0.60    0.55 0.45 1.3
## Q03    3                -0.51    0.53 0.47 1.8
## Q04    4                0.41    0.47 0.53 2.6
## Q16   16                0.41    0.49 0.51 2.4
## Q01    1          0.33  0.40    0.43 0.57 2.4
## Q05    5                0.34    0.34 0.66 2.7
## Q22   22                0.65 0.46 0.54 1.2
## Q09    9                0.63 0.48 0.52 1.4
## Q23   23                0.61 0.41 0.59 1.6
## Q02    2                -0.36 0.51 0.41 0.59 1.9
## Q19   19                -0.35 0.38 0.34 0.66 2.1
##
##              TC1  TC4  TC3  TC2
## SS loadings    3.90 2.88 2.94 1.85
## Proportion Var 0.17 0.13 0.13 0.08
## Cumulative Var 0.17 0.29 0.42 0.50
## Proportion Explained 0.34 0.25 0.25 0.16

```

```

## Cumulative Proportion 0.34 0.59 0.84 1.00
##
## With component correlations of
##      TC1  TC4  TC3  TC2
## TC1  1.00  0.44  0.36 -0.18
## TC4  0.44  1.00  0.31 -0.10
## TC3  0.36  0.31  1.00 -0.17
## TC2 -0.18 -0.10 -0.17  1.00
##
## Mean item complexity = 1.6
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is 0.06
## with the empirical chi square 4006.15 with prob < 0
##
## Fit based upon off diagonal values = 0.96

```

**Subtask 3** The factors found seem to be meaningful. Field names them: 1. Fear of computers (TC1) 2. Fear of statistics (TC4) 3. Fear of mathematics (TC3) 4. Peer evaluation (TC2)

As all of the factors are variations of some type of fear, accepting correlations between them make sense. So it's adequate to run oblique rotation.

The relatively high intercorrelations of TC1, TC3, and TC4 also make sense, because they are conceptually close to each other. This holds also for the low correlations they have with TC2.

```

# extract factor scores from the result object
scores <- pc4$scores

# we set meaningful column names
colnames(scores) <- c("fear_comp", "fear_stats", "fear_math", "fear_peer_eval")

# we add the factor scores to our data table
raq_data_scores <- cbind(raq_data, scores)

```

**Subtask 4** We could use the factor scores f. e. in a regression, where we use them as predictors (explanatory variables).

## 6) Differences between PCA and EFA

The most significant difference was mentioned above already. Here we want to focus on a more practical aspect.

1. Conduct an EFA using the command `fa()` of package `psych()`. Use the same arguments you had in `pc2`, also without rotation. The commands have the same structure. Store the result and name it `fa1`.
2. Run a second EFA and extract 10 factors this time. This would be the recommendation of the Joliffes Criterion. Store the result and name it `fa2`.
3. Run another PCA and extract 10 factors without rotating them. Store the result and name it `pc5`.
4. Use the function `head()` to inspect the factor scores of the first 6 subjects (observations).
  - a) Compare the factor scores of the first four factors in the results of `fa1` and `fa2`.

- b) Compare the factor scores of the first four factors in the results of pc2 and pc5.
- c) Is there anything special, you notice?

## Solutions

```
fa1 <- psych::fa(raq_data, nfactors = 4, rotate = "none")
```

### Subtask 1

```
fa2 <- psych::fa(raq_data, nfactors = 10, rotate = "none")
```

### Subtask 2

```
pc5 <- psych::principal(raq_data, nfactors = 10, rotate = "none")
```

### Subtask 3

```
# a)
fa1$scores %>% head()
```

### Subtask 4

```
##           MR1           MR2           MR3           MR4
## [1,]  1.12929066  1.0836217 -0.72346449  0.2981590
## [2,]  0.57074542 -0.2376027  0.03663449 -0.3602815
## [3,]  0.02563598  0.5075821  0.75049048  0.1906974
## [4,] -0.57019904  0.7244243 -0.39525454 -0.4942844
## [5,]  0.52745076 -0.2307131 -0.90621634  0.2706405
## [6,] -0.76422989  0.5264649 -1.52869713  0.6392913
```

```
fa2$scores %>% head()
```

```
##           MR1           MR2           MR3           MR4           MR5           MR6           MR7           MR8
## [1,]  1.12943990  1.1754210  1.5165187 -0.3294827  0.78782142  0.36858800 -0.1284385 -0.3053437  0.3
## [2,]  0.51951224 -0.6122700 -0.7116855 -0.4328797 -0.10022246 -1.73018927 -0.7531633  0.8295174 -0.1
## [3,]  0.04541655  0.9161946 -0.3185354  0.8132163  0.03427845  0.68250188  0.1819962 -0.3799708 -0.2
## [4,] -0.50369767 -0.1195866 -0.5110000 -1.3275876  0.55824535 -0.33766887  0.7341396  0.0557121  1.4
## [5,]  0.52633649 -0.5311493 -0.3801560 -0.9411793 -1.25620231 -0.03242136  0.5857954 -0.5896305 -0.0
## [6,] -0.98203326  1.2499543  0.8345435 -0.4954983 -2.01377386 -1.27413668 -1.7633160  0.1322869 -0.5
##           MR10
## [1,]  0.230225683
```

```
## [2,] -0.005372166
## [3,]  0.585253089
## [4,]  0.036917681
## [5,] -0.017913949
## [6,] -0.284332291
```

```
# b)
pc2$scores %>% head()
```

```
##           PC1           PC2           PC3           PC4
## [1,]  1.173758797  1.1108958 -1.2891595 -0.38759096
## [2,]  0.570078292 -0.1798303  0.3251963 -0.01212751
## [3,]  0.006509094  0.8039893  0.5184742  0.60730398
## [4,] -0.637584856  0.5108495 -0.4696227 -0.83510173
## [5,]  0.587687572 -0.3888630 -0.7160384 -0.43247530
## [6,] -0.881833937  0.4708284 -1.8625047 -0.14544207
```

```
pc5$scores %>% head()
```

```
##           PC1           PC2           PC3           PC4           PC5           PC6           PC7           PC8
## [1,]  1.173758797  1.1108958 -1.2891595 -0.38759096  1.2087498 -1.5574909  1.36766911  0.1323084  0.0
## [2,]  0.570078292 -0.1798303  0.3251963 -0.01212751 -0.9560550  0.4647447 -2.35726972 -1.1039647  0.0
## [3,]  0.006509094  0.8039893  0.5184742  0.60730398  0.3325874  1.9422485 -0.14897081  1.4778462 -0.0
## [4,] -0.637584856  0.5108495 -0.4696227 -0.83510173  0.8847922 -0.2008793 -1.80490370 -0.6602219  0.0
## [5,]  0.587687572 -0.3888630 -0.7160384 -0.43247530 -1.2150603 -0.6986948  0.44799485 -1.0063365 -0.0
## [6,] -0.881833937  0.4708284 -1.8625047 -0.14544207 -1.2126434 -0.4977185 -0.09233893  0.7480276 -1.0
##           PC10
## [1,] -0.5007619
## [2,] -2.0517548
## [3,]  0.6393485
## [4,]  1.0952371
## [5,]  0.8121233
## [6,] -2.0131043
```

The factor scores of the first four factors differ between `fa1` and `fa2`, whereas they keep the same in `pc2` and `pc5`.

## 7) Rendering

Render or knit your Rmd file using the shortcut `strg + shift + k` (Windows) or `cmd + shift + k`. If that works: Well done! If not, look at the error message, in special the lines in the syntax, where the error occurred. Correct the error and start over again. We can help you in our exercise hour.

## Literature

*Annotation:* This exercise sheet is based in part on the exercises from the textbook *Discovering Statistics Using R* (Field, Miles & Field, 2012). We modified it for the purpose of this exercise and actualized the R-Code.

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. London: SAGE Publications Ltd.