Georg Hosoya

Einführung in die Analyse testtheoretischer Modelle mit `R`

Eine praxisorientierte Ergänzung zu dem Buch

Testtheorie und Testkonstruktion von Michael Eid und Katharina Schmidt

Inhaltsverzeichnis

1 Vorbemerkungen

Die Reihenfolge der Syntax orientiert sich an der Ordnung der Präsentation der Beispiele im Buch. Zunächst wird die gesamte kommentierte Syntax des jeweiligen Kapitels zum Überblick dargestellt. Daraufhin werden die einzelnen Befehle sowie deren Ausgaben kurz erläutert.

Die hier gegebenen kurzen Erläuterungen können keine detaillierte Auseinandersetzung mit R, den verwendeten Paketen und den mathematischen Grundlagen der Modelle ersetzen und dienen lediglich als eine Einstiegshilfe in die Materie der psychometrischen Analyse mit R.

1.1 Verwendete Pakete und deren Installation

Es wurden hauptsächlich die Pakete `eRm`, `ltm`, `lavaan`, `psych` und `psychomix` für folgende Modelle verwendet:

`eRm` (Mair & Hatzinger, 2007):

- Das dichotome Rasch-Modell
- Das Partial-Credit-Modell

`ltm` (Rizopoulous, 2006):

- Das einparametrische logistische Modell (1PL, Rasch-Modell)
- Das zweiparametrische logistische Modell (2PL, Birnbaum-Modell)
- Das dreiparametrische logistische Modell (3PL, Birnbaum-Modell mit Rateparametern)
- Das generalisierte Partial-Credit-Modell

`lavaan` (Rosseel, 2012):

- Das Modell essenziell τ-äquivalenter Messung
- Das Modell τ-äquivalenter Messung
- Das Modell essenziell τ-paralleler Messung
- Das Modell τ-paralleler Messung
- Das Modell τ-kongenerischer Messung
- Mehrdimensionales Modell für kontinuierliche Variablen
- Mehrkomponentenmodell (Latent-State-Trait-Modell)

`psych` (Revelle, 2013):

- Faktorenanalyse für kategoriale Variablen

`psychomix` (Frick et al., 2012):

- Mischverteilungs-Rasch-Modell

Um die Pakete zu nutzen, müssen diese zunächst einmal auf dem Rechner installiert werden. Der entsprechende Befehl ist z.B. für das Paket `eRm`:

```
> install.packages("eRm")
```

Hierzu ist eine Internetanbindung nötig. Das `R`-Skript `install.R` enthält den Code zur Installation aller benötigten Pakete. Dieses Skript sollte auf einem Rechner einmal ausgeführt werden, bevor die anderen Skripte genutzt werden können. Für eine einstiegsorientierte Einführung in `R` sei z.B. das Buch von Luhmann (2013) empfohlen.

1.2 Wechseln des Arbeitsverzeichnisses

Um die Skripte auszuführen, wechseln Sie in `R` bitte in das Verzeichnis, in dem die Daten liegen.

Die entsprechenden Befehle sind z.B.:

`> setwd("C:/Pfad_zu_den_Daten")` (für Windows)

oder

`> setwd("/Pfad_zu_den_Daten")` (für UNIX/Linux/Mac).

2 Beispielsyntax für Kapitel 4: Eindimensionale Modelle für dichotome Antwortvariablen

Für Analysen mit dem Rasch-Modell eignet sich das `R`-Paket `eRm`. Die Parameter des zweiparametrischen logistischen Modells und des dreiparametrischen logistischen Modells können mit dem Paket `ltm` geschätzt werden.

2.1 Überblick Gesamtsyntax

```
#### Syntax Kapitel 4 ####
# Benötigt für spss.system.file()
library("memisc")

# *.sav-file einlesen
data<-as.data.set(spss.system.file('ids_new.sav'))

# Konvertierung der Daten in den Typ data.frame
data<-data.frame(data)

# Schätzung der "klassischen" Itemschwierigkeiten
# und Itemvarianzen
mean<-apply(data,2,mean)
var<-apply(data,2,var)

# Berechnung der Gamma-Koeffizienten,
# bzw. Yules Q
library(vcdExtra)
tab1<-table(data$Freude_1, data$Wut_1)
GKgamma(tab1, level=0.95)

## Dichotomes Rasch-Modell mit eRm() #############
library("eRm")

# Konvertierung der Test-Daten in den Typ matrix
# und Auswahl der 10 Items (ohne Geschlecht)
data.mat<-as.matrix(data[,2:11], dimnames=names(data)[2:11])

# bedingte Maximum-Likelihood-Schätzung (CML-Schätzung)
# der Itemparameter
rm1<-RM(data.mat)
summary(rm1)

# Schätzung der Personenwerte
p.par<-person.parameter(rm1)
print(p.par)
summary(p.par)

# bedingter Likelihood-Quotienten-Test mit Geschlecht
# als Split-Kriterium
lrt1<-LRtest(rm1, splitcr=data$sex)
summary(lrt1)

# Grafischer Modelltest
plotGOF(lrt1, conf=list(), ylab="Jungen", xlab="Mädchen")

# Darstellung der Itemcharakteristikfunktionen
plotICC(rm1)

# Gemeinsame Darstellung der Itemcharakteristikfunktionen
plotjointICC(rm1)

# Darstellung der Person-Item-Map
plotPImap(rm1)
```

```
# Waldtest mit Geschlecht
# als Split-Kriterium
Waldtest(rm1, splitcr=data$sex)

# Martin-Loef-Test
MLoef(rm1, splitcr="median")

# Item-Fit
itemfit(p.par)

# Personen-Fit
personfit(p.par)

## Zweiparametrisches logistisches Testmodell (2PL)
#  mit ltm()

library("ltm")

#Schätzung des 2PL
two_pl<-ltm(data.mat~z1)
summary(two_pl)

# Schätzung des 2PL unter Annahme
# identischer Diskrimationsparameter (Rasch-Modell)
two_pl_ident<-rasch(data = data.mat)
summary(two_pl_ident)


# Likelihood-Ratio-Tests zum Vergleich
# des 2PL-Modells mit dem Rasch-Modell
anova(two_pl_ident, two_pl)


# Schätzung des 3PL mit Optimizer "nlminb"
three_pl<-tpm(data.mat, control=list(optimizer="nlminb"))
summary(three_pl)

# Darstellung der Itemcharakteristikfunktionen
plot(two_pl)

# Ausgabe der geschätzten Personenwerte
factor.scores(two_pl)


## Mischverteilungs-Rasch-Modell mit psychomix()
# Modell mit zwei Klassen

install.packages("psychomix", dependencies = TRUE)
# Das Paket muss nur einmal installiert werden, es wird dann mit dem
#"library"-Befehl geladen. Immer wenn R neu aufgerufen wird, muss ein
# Paket neu geladen werden

library("psychomix")

mrm1<-raschmix(data=data.mat, k=2, scores="meanvar")
summary(mrm1)
```

```
## Anmerkung: in Eid und Schmidt (2014)
## wurde WINMIRA (Davier, 2001) verwendet
```

2.2 Daten einlesen und deskriptive Statistiken

```
> library("memisc")
```

Zunächst muss die SPSS-Datei `ids.sav` eingelesen werden. Hierbei kommt das Paket `memisc` zum Einsatz, welches die Funktion `spss.system.file()` enthält.

```
> data<-as.data.set(spss.system.file('ids_new.sav'))
> data<-data.frame(data)
```

Die Daten werden schrittweise eingelesen. Zunächst werden sie über die Funktion `as.data.set()` in `R` importiert und hiernach zur einfacheren Handhabung bei folgenden Analysen in den Typ `data frame` umgewandelt.

```
> mean<-apply(data,2,mean)
> var<-apply(data,2,var)
```

Zur Berechnung der Mittelwerte und Varianzen der Variablen, bzw. Items im Daten-Frame kann die Funktion `apply()` verwendet werden. Hierbei ist zu beachten, dass die erste Spalte des Daten-Frames kein Item, sondern die Kodierung des Geschlechts der Teilnehmer (sex) enthält. Diese Schritte sind interessant, um sich einen ersten Eindruck über die deskriptiven Charakteristiken der Items zu verschaffen.

2.3 Berechnung der Gamma-Koeffizienten bzw. Yules Q

```
> library(vcdExtra)
```

Zur Berechnung der Gamma-Koeffizienten wird die Bibliothek `vcdExtra` benötigt.

```
> tab1<-table(data$Freude_1, data$Wut_1)
```

Es wird eine Kontingenztafel der gemeinsamen Verteilung von Item 1 und Item 2 erzeugt.

```
> GKgamma(tab1, level=0.95)
gamma         : 0.875
std. error    : 0.068
CI            : 0.742 1
```

Der geschätzte Gamma-Koeffizient inklusive des Standardfehlers und des 95%-Konfidenzintervalls wird ausgegeben. Dieser Vorgang muss für alle Itemkombinationen wiederholt werden.

2.4 Analysen mit den dichotomen Rasch-Modell

2.4.1 Laden des Paketes `eRm`

```
> library("eRm")
```

Das Paket `eRm` enthält umfangreiche Funktionen zur psychometrischen Analyse vor dem theoretischen Hintergrund des Rasch-Modells. Erwähnenswert ist, dass das Paket explizit auf der bedingten Maximum-Likelihood-Schätzmethode (*conditional maximum likelihood estimation*, CML) aufbaut. Dieses Paket wird zur Durchführung der Beispiel-Analysen mit dem Rasch-Modell verwendet.

2.4.2 Selektion und Konvertierung der Daten

```
> data.mat<-as.matrix(data[,2:11], dimnames=names(data)[2:11])
```

Das Paket `eRm` benötigt als Datenformat den Typ `matrix`. Von daher werden die Items im Daten-Frame `data` in den Typ `matrix` konvertiert. Es werden nur die Spalten 2 bis 11 des Daten-Frames selektiert, da die erste Spalte die Variable Geschlecht (`sex`) enthält. In die Rasch-Analyse sollen aber nur die 10 Items eingehen. Zudem werden die Namen der Items über das Argument `dimnames=names(data)[2:11]` in die neu erzeugte Matrix `data.mat` übernommen. Diese Matrix dient als Ausgangspunkt der weiteren Analysen.

2.4.3 Bedingte Maximum-Likelihood-Schätzung

```
> rm1<-RM(data.mat)
```

Die Itemparameter des dichotomen Rasch-Modells werden anhand der Funktion `RM()` geschätzt. Die Ausgabe wird dem Objekt `rm1` zugewiesen.

```
> summary(rm1)
```

Die Ergebnisse der Schätzung werden mit der Funktion `summary()` in strukturierter Form ausgegeben. Die Ausgabe wird im Folgenden insgesamt dargestellt und dann Schritt für Schritt kurz erläutert.

```
Results of RM estimation:
```

```
Call:  RM(X = data.mat)

Conditional log-likelihood: -496.0098
Number of iterations: 13
Number of parameters: 9

Item (Category) Difficulty Parameters (eta): with 0.95 CI:
          Estimate Std. Error lower CI upper CI
Wut_1       -0.958      0.256   -1.459   -0.456
Angst_1      0.909      0.169    0.577    1.240
Trauer_1     0.705      0.173    0.366    1.045
Ueber_1     -0.115      0.202   -0.511    0.281
Trauer_2     0.645      0.174    0.303    0.987
Angst_2      1.180      0.166    0.856    1.505
Wut_2       -1.490      0.306   -2.090   -0.891
Ueber_2      1.413      0.164    1.091    1.734
Freude_2    -0.686      0.236   -1.148   -0.224

Item Easiness Parameters (beta) with 0.95 CI:
               Estimate Std. Error lower CI upper CI
beta Freude_1    1.603      0.318    0.979    2.226
beta Wut_1       0.958      0.256    0.456    1.459
beta Angst_1    -0.909      0.169   -1.240   -0.577
beta Trauer_1   -0.705      0.173   -1.045   -0.366
beta Ueber_1     0.115      0.202   -0.281    0.511
beta Trauer_2   -0.645      0.174   -0.987   -0.303
beta Angst_2    -1.180      0.166   -1.505   -0.856
beta Wut_2       1.490      0.306    0.891    2.090
beta Ueber_2    -1.413      0.164   -1.734   -1.091
beta Freude_2    0.686      0.236    0.224    1.148
```

Die logarithmierte bedingte Likelihood am Ort des Maximums beträgt -496.01. Dieser Wert ist für Likelihood-Quotienten-Tests von Interesse, die ebenfalls im Paket `eRm` implementiert sind. Die Anzahl der Iterationen bis zur Konvergenz der Schätzung betrug 13 Schritte. Insgesamt wurden in der Implementation 9 Parameter geschätzt, da implizit eine Summennormierung durchgeführt wurde.

```
Item (Category) Difficulty Parameters (eta): with 0.95 CI:
          Estimate Std. Error lower CI upper CI
Wut_1       -0.958      0.256   -1.459   -0.456
[…]
```

Die geschätzte Schwierigkeit des Items `Wut_1` beträgt unter der Annahme des bedingten Rasch-Modells `-0.958`. Der Standardfehler dieses Schätzers ist `0.256.` Das 0.95-Konfidenzintervall ist CI[-1.459, 0.456]. Für das Item Freude_1 wird kein Parameter ausgegeben, da sich dessen geschätzter Schwierigkeitsparameter aufgrund der Summennormierung aus den anderen geschätzten Schwierigkeitsparametern ergibt. Für das Item Freude_1 wird aber – wie für die anderen Items auch – der geschätzte Leichtigkeitsparameter ausgegeben (Schwierigkeitsparameter multipliziert mit -1):

```
Item Easiness Parameters (beta) with 0.95 CI:
              Estimate Std. Error lower CI upper CI
beta Freude_1    1.603      0.318    0.979    2.226
[…]
```

Die Implementation der Rasch-Modelle in `eRm` baut auf einem integrativen CML-Framework auf. Für eine Vertiefung sei das Buch von Koller, Alexandrowicz und Hatzinger (2012) und die Dokumentation des Paketes `eRm` (Mair & Hatzinger, 2007) empfohlen.

2.4.4 Maximum-Likelihood-Schätzung der Personenparameter

```
> p.par<-person.parameter(rm1)
```

Mit der bedingten Maximum-Likelihood-Schätzmethode werden die Itemparameter unabhängig von den Personenwerten geschätzt. Aufbauend auf den geschätzten Itemparametern werden daraufhin die Personenwerte mit der Maximum-Likelihood-Methode geschätzt (s. Abschnitt 4.5.3.6 in Eid & Schmidt, 2014). Dies geschieht durch die Funktion `person.parameter()` im Paket `eRm`. Die Ausgabe der Funktion wird dem Objekt `p.par` zugewiesen.

```
> print(p.par)
```

Eine Tabelle der Summenwerte und der Personenwerte (`person parameters)` wird ausgegeben.

```
Person Parameters:

 Raw Score    Estimate Std.Error
         0 -3.61779726        NA
         1 -2.63251737 1.1028495
         2 -1.71234712 0.8589364
         3 -1.05807884 0.7702325
         4 -0.49882926 0.7304911
         5  0.02159298 0.7157188
         6  0.53543739 0.7216184
         7  1.07645082 0.7552292
         8  1.70456439 0.8421463
         9  2.59497001 1.0891755
        10  3.54991655        NA
```

In der ersten Spalte sind die Rohwerte (Summenwerte) abgetragen, die zweite Spalte zeigt die geschätzten Personenwerte (Personenparameter) und die dritte Spalte enthält die entsprechenden Standardfehler der Maximum-Likelihood-Schätzer. Ein `NA` (*not applicable* oder *not available*) bedeutet, dass der entsprechende Standardfehler nicht geschätzt werden konnte. Dies ist für den geringsten und höchstmöglichen Summenwert mit der Maximum-Likelihood-Methode nicht möglich (s. Abschnitt 4.5.3.6 in Eid & Schmidt, 2014).

```
> summary(p.par)
```

Dieser Befehl zeigt die geschätzten Personenwerte einer jeden Person in dem Datensatz, deren Standardfehler sowie die dazugehörigen 0.95-Konfidenzintervalle.

```
Estimation of Ability Parameters

Collapsed log-likelihood: -40.02453
Number of iterations: 9
Number of parameters: 9
ML estimated ability parameters (without spline interpolated
values):
             Estimate Std. Err.       2.5 %      97.5 %
theta 1    2.59497001 1.0891755  0.46022531  4.72971471
theta 3    0.53543739 0.7216184 -0.87890864  1.94978342
[…]
```

Der geschätzte Personenwert oder Fähigkeitsparameter *(ability parameter)* wird in der Ausgabe mit `theta1` usw. bezeichnet. Die *collapsed log-likelihood* ist die Likelihood des Modells, das der Maximum-Likelihood-Schätzung der Personenwerte zugrunde liegt. Diese ist von technischem Interesse. Die Personenwerte müssen nicht für jede Person einzeln, sondern können über Gruppen von Personen mit identischen Rohwerten geschätzt werden, da im dichotomen Rasch-Modell der Summenwert einer Person eine suffiziente Statistik für den zu schätzenden Parameter darstellt. Von daher erklärt sich der relativ geringe Wert von lediglich -40.025. „Collapsed" bedeutet hier „zusammengefasst". Insgesamt wurden 9 Iterationen für die Konvergenz benötigt und 9 verschiedene Personenwerte wurden geschätzt.

2.4.5 Der bedingte Likelihood-Quotienten-Test nach Andersen

```
> lrt1<-LRtest(rm1, splitcr=data$sex)
> summary(lrt1)
```

Zur Testung der Annahme der Gleichheit der Itemparameter zwischen Subpopulationen kommt der bedingte Likelihood-Quotienten-Test nach Andersen (s. Abschnitt 4.5.4.1.2 in Eid & Schmidt, 2014) mit dem Split-Kriterium „Geschlecht" zum Einsatz. Die Funktion `LRtest()` übernimmt als Argument das Objekt `rm1`, das durch die Funktion `RM()` erzeugt wurde. Das Split-Kriterium (Trennkriterium für die beiden Gruppen) wird über das Argument `splitcr=data$sex` angegeben.

```
Andersen LR-test:
```

```
LR-value: 9.298
Chi-square df: 9
p-value:  0.41


Subject Subgroup: data$sex 0:
Log-likelihood:  -276.0868
[...]
```

Die Teststatistik des bedingten Likelihood-Quotienten-Tests beträgt mit 9 Freiheitsgraden χ^2 = 9.30. Die Überschreitungswahrscheinlichkeit beträgt p = .41. Damit kann die Nullhypothese, dass sich die Itemparameter in den Geschlechtsgruppen nicht unterscheiden, beibehalten werden.

```
Subject Subgroup: data$sex 0:
Log-likelihood:  -276.0868

Beta Parameters:
         beta Freude_1
Estimate         1.4293717
Std.Err.         0.4089078
[…]

Subject Subgroup: data$sex 1:
Log-likelihood:  -215.2741

Beta Parameters:
         beta Freude_1
Estimate         1.8201896
Std.Err.         0.5073775
[…]
```

Die geschätzten Itemparameter und deren Standardfehler werden für beide Geschlechtsgruppen separat ausgegeben. Gleiches gilt für die Log-Likelihood der auf die beiden Gruppen separat angepassten bedingten Rasch-Modelle.

2.4.6 Grafischer Modelltest

```
> plotGOF(lrt1, conf=list(), ylab="Jungen", xlab="Mädchen")
```

Es wird eine Grafik angezeigt, in der die geschätzten Itemparameter der Mädchen und der Jungen gegeneinander abgetragen werden (s. Abschnitt 4.5.4.1.1 in Eid & Schmidt, 2014). Das Erstellen dieser Grafik wird auch als grafischer Modelltest bezeichnet. Die Kreise geben die Größe der Konfidenzintervalle an.

2.4.7 Person-Item-Karte

```
> plotPImap(rm1)
```

Eine Person-Item-Karte *(person-item map)* wird erstellt. Im oberen Teil der Abbildung wird ein Histogramm der Personenwerte gezeigt. Im unteren Teil der Abbildung sind die Anordnungen der Itemschwierigkeiten auf der latenten Dimension abgetragen. Die „kleinen Striche“ unter dem Histogramm zeigen die Anordungen der geschätzen Itemschwierigkeiten oder Schwellen (im Partial-Credit-Modell, s. Abschnitt 4.4) auf der latenten Dimension an.

2.4.8 Darstellung der Itemcharakteristikfunktionen

```
> plotICC(rm1)
```

Die Itemcharakteristikfunktionen (s. Abschnitt 4.5.1 in Eid & Schmidt, 2014) des auf die Daten angepassten Modells werden grafisch ausgegeben.

```
> plotjointICC(rm1)
```

Die Itemcharakteristikfunktionen werden gemeinsam dargestellt.

2.4.9 Wald-Test zur Überprüfung der Gleichheit der Itemschwierigkeiten in Subgruppen

```
> Waldtest(rm1, splitcr=data$sex)
```

Es wird ein Wald-Test (s. Abschnitt 4.5.4.1.3 in Eid & Schmidt, 2014) mit Geschlecht als Split-Kriterium durchgeführt, d.h., für jedes Item wird die Nullhypothese geprüft, dass sich die Itemschwierigkeiten in den beiden Geschlechtsgruppen nicht unterscheiden.

```
Wald test on item level (z-values):

                   z-statistic
beta Freunde_1          -0.600
beta Wut_1               1.143
beta Trauer_1           -1.357
[…]
                   p-value
beta Freude_1        0.549
beta Wut_1           0.253
[…]
```

Für jedes Item werden die Teststatistik und die entsprechende Überschreitungswahrscheinlichkeit ausgegeben.

2.4.10 Martin-Löf-Test

```
> MLoef(rm1, splitcr="median")
```

Ein Martin-Löf-Test (s. Abschnitt 4.5.4.4 in Eid & Schmidt, 2014) mit dem Split-Kriterium „Median der Rohwertverteilung der Items“ wird durchgeführt.

```
Martin-Loef-Test (split criterion: median)
LR-value: 22.668
Chi-square df: 24
p-value: 0.539
```

Die Teststatistik, deren Freiheitsgrade und die Überschreitungswahrscheinlichkeit werden ausgegeben.

2.4.11 Residualstatistiken: Itemfit

```
> itemfit(p.par)

Itemfit Statistics:
               Chisq  df p-value Outfit MSQ Infit MSQ Outfit t Infit t
Freunde_1     75.284 156   1.000      0.480     0.863    -1.38   -0.49
Wut_1        176.465 156   0.125      1.124     1.093     0.50    0.55
[…]
```

Die Itemfit-Statistiken zur Identifikation von abweichenden Items (s. Abschnitt 4.5.4.6 in Eid & Schmidt, 2014) werden berechnet und ausgegeben. Die χ^2-Teststatistik ist ein residualbasiertes Abweichungsmaß, welches bei Geltung der Nullhypothese der Modellpassung χ^2-verteilt ist. Die Freiheitsgrade des Tests (df = 156) und die Überschreitungswahrscheinlichkeiten werden pro Item dargestellt. Zudem werden der Outfit, der Infit und die dazugehörigen *t*-verteilten Teststatistiken ausgegeben. Out- und Infit-Statistiken werden von Eid und Schmidt (2014) nicht behandelt (s. hierzu Koller, Alexandrowicz & Hatzinger, 2012).

2.4.12 Residualstatistiken: Personenfit

```
> personfit(p.par)

Personfit Statistics:
     Chisq df p-value Outfit MSQ Infit MSQ Outfit t Infit t
1    4.148  9   0.901      0.415     0.816    -0.04    0.00
3    7.037  9   0.633      0.704     0.825    -0.63   -0.53
[…]
```

Die entsprechenden residualbasierten Teststatistiken und Fit-Indizes werden über die Personen berechnet und ausgegeben (s. Abschnitt 4.5.4.7 in Eid & Schmidt, 2014). Person 2 taucht in der Ausgabe nicht auf, da diese alle Items gelöst hat und somit kein Personenwert mit der Maximum-Likelihood-Methode geschätzt werden konnte.

2.5 Analysen mit dem zweiparametrischen logistischen Modell (Birnbaum-Modell) für dichotome Antwortvariablen

Zur Analyse der Daten mit dem zweiparametrischen logistischen Modell (Birnbaum-Modell, 2PL-Modell) in `R` kommt das Paket `ltm` zum Einsatz.

```
> two_pl<-ltm(data.mat~z1)
> summary(two_pl)
```

Die Funktion `ltm` übernimmt als Argument die Modellformel `data.mat~z1`, welche das zweiparametrische logistische Modell (2PL-Modell) ohne Restriktionen auf den Diskriminationsparametern definiert. Das Objekt `data.mat` ist die Datenmatrix, die schon für die Analysen mit eRm verwendet wurde. Der Ausdruck `z1` bedeutet, dass a priori im Einklang mit der Hypothese der Eindimensionalität lediglich eine latente Variable spezifiziert wird. Das Ergebnis der Parameterschätzung wird auf der Konsole ausgegeben.

```
Call:
ltm(formula = data.mat ~ z1)

Model Summary:
   log.Lik      AIC      BIC
 -876.3027 1792.605 1857.859
[…]
```

Es werden die Log-Likelihood des auf die Daten angepassten Modells und die informationstheoretischen Indizes AIC und BIC ausgegeben, die für den Modellvergleich interessant sind.

Die Log-Likelihood des Modells beträgt -876.303. Der AIC-Index beträgt 1729.605 und der BIC beträgt 1857.859.

```
Coefficients:
                  value std.err  z.vals
Dffclt.Freude_1 -1.6387  0.2374 -6.9016
Dffclt.Wut_1    -1.9338  0.4004 -4.8304
Dffclt.Angst_1  -0.7381  0.2485 -2.9703
Dffclt.Trauer_1 -1.0637  0.3330 -3.1938
Dffclt.Ueber_1  -1.3770  0.2891 -4.7637
Dffclt.Trauer_2 -1.0415  0.3063 -3.3998
Dffclt.Angst_2  -0.3845  0.1674 -2.2968
Dffclt.Wut_2    -1.4876  0.1965 -7.5704
Dffclt.Ueber_2  -0.1923  0.1564 -1.2294
```

```
Dffclt.Freude_2 -1.1954  0.1756 -6.8092
Dscrmn.Freude_1  2.7433  0.8207  3.3426
Dscrmn.Wut_1     1.3229  0.3647  3.6270
Dscrmn.Angst_1   0.9638  0.2832  3.4038
Dscrmn.Trauer_1  0.8193  0.2439  3.3592
Dscrmn.Ueber_1   1.3126  0.3395  3.8659
Dscrmn.Trauer_2  0.9097  0.2618  3.4749
Dscrmn.Angst_2   1.2615  0.3421  3.6871
Dscrmn.Wut_2     3.5343  1.1687  3.0241
Dscrmn.Ueber_2   1.2315  0.3379  3.6442
Dscrmn.Freude_2  3.1003  0.9554  3.2449
[…]
```

Die geschätzten Itemparameter werden ausgegeben. In der oberen Hälfte der Ausgabe befinden sich die Itemschwierigkeiten (`Dffclt.*`), deren Standardfehler sowie eine *z*-verteilte Teststatistik der Nullhypothese, dass die Schwierigkeitsparameter in der Population Null betragen. In der unteren Hälfte der Ausgabe befinden sich die geschätzten Diskriminationsparameter (`Dscrm.*`) sowie deren Standardfehler und Teststatistiken.

```
Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.0016
quasi-Newton: BFGS
```

Es werden technische Details zur Parameterschätzung ausgegeben. Die latente Variable wurde mittels der Gauss-Hermite-Quadratur mit 21 Stützpunkten „herausintegriert". Die Parameterschätzung mittels des BFGS-Algorithmus ist konvergiert (`Convergence: 0`). Der maximale Betrag des Gradienten ist 0.0016.

```
> two_pl_ident<-rasch(data = data.mat)
> summary(two_pl_ident)
```

Das 2PL-Modell mit identischen Ladungen (das dichotome Rasch-Modell) wird mittels der Funktion `rasch()` im Paket `ltm` angepasst (s. hierzu Eid & Schmidt, 2004, Abschnitt 4.5.3.4).

```
Call:
rasch(data = data.mat)

Model Summary:
   log.Lik      AIC      BIC
 -890.9735 1803.947 1839.837
```

Die Struktur der Ausgabe unterscheidet sich im Wesentlichen nicht von derjenigen des 2PL-Modells ohne Restriktionen auf den Ladungsparametern. Zunächst werden die Log-Likelihood und die informationstheoretischen Indizes ausgegeben.

```
Coefficients:
                       value std.err  z.vals
Dffclt.Trauer_1      -2.2878  0.2785 -8.2132
Dffclt.Ueber_1       -1.9089  0.2405 -7.9385
[…]
Dscrmn                1.3553  0.1222 11.0924
```

Es folgen die Modellkoeffizienten in Form der Schwierigkeiten und des für alle Items identischen Diskriminationsparameters (`Dscrm`). Die entsprechenden Standardfehler und Teststatistiken werden ebenfalls berichtet.

```
Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 5e-04
quasi-Newton: BFGS
```

Bei der Schätzung kam standardmäßig der BFGS-Algorithmus zum Einsatz und es wurde die Gauss-Hermite-Quadratur mit 21 Stützpunkten zur Integration über die latente Trait-Verteilung verwendet. Der Algorithmus ist konvergiert.

2.5.1 Vergleich des 2PL-Modells mit dem Rasch-Modell

```
> anova(two_pl_ident, two_pl)
```

Mittels eines Likelihood-Ratio-Tests wird geprüft, ob das 2PL-Modell mit frei schätzbaren Ladungen die Daten signifikant besser erklärt, als das Modell mit identischen Ladungen (Rasch-Modell).

```
Likelihood Ratio Table
                 AIC     BIC log.Lik   LRT df p.value
two_pl_ident 1803.95 1839.84 -890.97
two_pl       1792.61 1857.86 -876.30 29.34  9   0.001
```

Der Likelihood-Quotienten-Test legt nahe, dass das 2PL-Modell mit frei geschätzten Ladungen (`two_pl`) die Daten besser erklärt als das Modell mit der Annahme der identischen Ladungen (`two_pl_ident`). Die χ^2-verteilte Teststatistik beträgt 29.34 mit 9 Freiheitsgraden. Die

Überschreitungswahrscheinlichkeit ist .001. Auch der AIC-Index legt nahe, dass das 2PL-Modell mit unterschiedlichen Diskriminationsparametern die Daten besser beschreibt, der BIC-Index deutet jedoch auf eine bessere und sparsamere Passung des Modells mit identischen Ladungsparametern hin.

2.5.2 Die Analyse des dreiparametrischen Modells (3PL-Modell)

Das 3PL-Modell wird mit der Funktion `tpm()` im Paket `ltm` angepasst.

```
> three_pl<-tpm(data.mat, control=list(optimizer="nlminb"))
> summary(three_pl)
```

Das 3PL-Modell unterscheidet sich vom 2PL-Modell dahingehend, dass pro Item ein sogenannter „Rateparameter“ in die Modellgleichung eingefügt wird. Da mit dem BFGS-Algorithmus Konvergenzprobleme bei der Parameterschätzung auftraten, wurde über das Argument `control` die Optimierungs-Routine „`nlminb`“ ausgewählt. Das Ergebnis der Schätzung wird über die Funktion `summary()` ausgegeben.

```
Model Summary:
   log.Lik      AIC     BIC
 -876.2546 1812.509 1910.39

Coefficients:
                  value std.err  z.vals
Gussng.Freude_1  0.1505  0.3708  0.4059
Gussng.Wut_1     0.0000  0.0005  0.0009
[…]
Dffclt.Freude_1 -1.4363  0.5365 -2.6774
Dffclt.Wut_1    -1.9275  0.3942 -4.8900
[…]
Dscrmn.Freude_1  3.3381  2.2773  1.4658
Dscrmn.Wut_1     1.3318  0.3658  3.6413


Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Optimizer: nlminb
Convergence: 0
max(|grad|): 0.0016
```

Der wesentliche Unterschied zu den bisherigen Ausgaben besteht darin, dass nun zusätzlich die geschätzten Rateparameter (`Gussng.*`), deren Standardfehler und Teststatistiken ausgegeben werden. Die Optimierung erfolgte mit den PORT-Routinen „`nlminb`“ und ist konvergiert. Eine Inspektion der Rateparameter in der Ausgabe legt nahe, dass diese für die vorliegenden Daten nicht benötigt werden.

2.5.3 Die Ausgabe der Itemcharakteristikkurven

```
> plot(two_pl)
```

Die Itemcharakteristikkurven des entsprechenden Modells (hier das 2PL-Modell) werden grafisch ausgegeben.

2.5.4 Die Schätzung der Personenwerte

```
> factor.scores(two_pl)
```

Die Personenwerte *(factor scores)* des 2PL-Modells werden für jedes Antwortmuster geschätzt und ausgegeben. Standardmäßig geschieht dies mit der Empirical-Bayes-Methode. Die Ausgabe enthält die beobachteten Antwortmuster, von daher wird die Ausgabe hier verkürzt dargestellt.

```
Call:
ltm(formula = data.mat ~ z1)

Scoring Method: Empirical Bayes

…
   Angst_2 Wut_2 Ueber_2 Freude_2 Obs    Exp     z1 se.z1
…        0     0       0        0   6  0.994 -2.295 0.487
…        0     1       0        1   1  0.042 -1.263 0.325
…        0     1       1        1   1  0.012 -1.017 0.357
…        1     1       1        1   1  0.005 -0.832 0.395
…
```

In der Ausgabe werden zunächst die Antwortmuster dargestellt. Die Spalte (`Obs`) enthält die beobachtete Häufigkeit des Antwortmusters, die Spalte `Exp` enthält die erwartete Häufigkeit des Antwortmusters, `z1` ist der geschätzte Personenwert (geschätzte Personenfähigkeit) und `se.z1` ist dessen Standardfehler.

2.6 Mischverteilungs-Rasch-Modelle

Eine vielversprechende Bibliothek zur Schätzung von Mischverteilungs-Rasch-Modellen ist `psychomix` (Frick et al., 2012). In Eid und Schmidt (2014) wurde WINMIRA (Davier, 2001) verwendet, von daher wird auf diese Software und deren Dokumentation verwiesen. Dennoch sei hier kurz angerissen, wie das Paket zur Anpassung von Mischverteilungs-Rasch-Modellen gehandhabt werden kann.

```
> mrm1<-raschmix(data=data.mat, k=2, scores="meanvar")
```

Ein Mischverteilungs-Rasch-Modell mit k=2 Klassen wird angepasst. Über das Argument `scores="meanvar"` werden die latenten Score-Verteilungen mittels eines multinomialen Logit-Modells mit einem Mittelwert- und Varianz-Parameter modelliert.

```
> summary(mrm1)

Call:
raschmix(data = data.mat, k = 2,
    scores = "meanvar")

       prior size post>0 ratio
Comp.1 0.509   86    135 0.637
Comp.2 0.491   71    157 0.452
```

Die geschätzten Klassengrößen werden ausgegeben.

```
Item Parameters:
             Comp.1     Comp.2
Freude_1 -0.6710822 -2.0008396
Wut_1     0.3527830 -2.0162904
Angst_1  -6.0358533  2.9410778
Trauer_1  1.4793925  0.6053855
Ueber_1   1.0927986 -1.0038794
Trauer_2  1.6217470  0.2407851
Angst_2   0.9099259  2.3508389
Wut_2    -0.8296391 -1.6424890
Ueber_2   2.4304225  1.0354368
Freude_2 -0.3504949 -0.5100258
```

Die geschätzten und summennormierten Itemparameter in den Klassen werden ausgegeben.

```
'log Lik.' -851.7621 (df=25)
AIC: 1753.524   BIC: 1829.93
```

Die informationstheoretischen Indizes AIC und BIC werden ausgegeben. Für eine detaillierte Auseinandersetzung mit dem Paket sei auf dessen Dokumentation hingewiesen.

3 Beispielsyntax für Kapitel 5: Eindimensionale Modelle für Antwortvariablen mit geordneten Antwortkategorien

3.1 Überblick Gesamtsyntax

```
#### Syntax Kapitel 5 ####

library(memisc)
# *.sav-file einlesen
data<-as.data.set(spss.system.file('Daten-kapitel-5-sex.sav'))
# Konvertierung der Daten in den Typ data.frame
```

```
data<-data.frame(data)

# Deskriptive Statistiken
mean<-apply(data[2:7],2,mean)
mean
median<-apply(data[2:7],2,median)
median
var<-apply(data[2:7],2,var)
var

# Gamma-Koeffizient
library(vcdExtra)
tab1<-table(data$item_1, data$item_2)
GKgamma(tab1, level=0.95)

# relativer Informationsgehalt
item_1_factor<- factor(data$item_1, labels=c('1','2','3','4','5'))
item_1_p<-prop.table(table(item_1_factor))
item_1_i<-(-1/log(5))*sum((item_1_p*log(item_1_p)))
item_1_i

# Anpassung des Partial-Credit-Modells mit eRm()
library(eRm)

# Konvertierung der Test-Daten in den Typ matrix
data.mat<-as.matrix(data[,2:7], dimnames=names(data)[2:7])


# CML-Schätzung der Item-Parameter
pcm<-PCM(data.mat)
summary(pcm)

# Berechnung der Schwellenparameter
thresholds(pcm)

# Darstellung der Itemcharakteristikkurven
plotICC(pcm)

# Darstellung der Person-Item-Map
plotPImap(pcm)

# ML-Schätzung der Personenwerte
p.par<-person.parameter(pcm)
summary(p.par)
print(p.par)

# Likelihood-Ratio-Test mit Geschlecht
# als Split-Kriterium
lrt1<-LRtest(pcm, splitcr=data$sex)
summary(lrt1)

# Grafischer Modelltest
plotGOF(lrt1, conf=list())

# Waldtest mit Geschlecht
# als Split-Kriterium
Waldtest(pcm, splitcr=data$sex)
```

```
# Martin-Loef-Test mit
# benutzerdefiniertem Split-Kriterium
MLoef(pcm, splitcr=c(1,0,1,1,0,0))

# Item-Fit
itemfit(p.par)

# Personen-Fit
personfit(p.par)

## Anpassung des generalisierten Partial-Credit-Modells

library(ltm)
gpcm<-gpcm(data.mat)
summary(gpcm)

GoF.gpcm(gpcm, B=300) # Goodness of Fit bootstrap

## Analyse der Items 1, 3 und 4

# Auswahl der Items 1, 3 und 4
data.sub<-data.mat[,c(4,1,3)]

# Anpassen des Partial-Credit-Modells
pcm.sub<-PCM(data.sub)
summary(pcm.sub)
thresholds(pcm.sub)

# Anpassen des Ratingskalenmodells
rsm.sub<-RSM(data.sub)
summary(rsm.sub)
thresholds(rsm.sub)
```

3.2 Deskriptive Statistiken und Gamma-Koeffizient

```
> # Deskriptive Statistiken
> mean<-apply(data[2:7],2,mean)
> mean
> median<-apply(data[2:7],2,median)
> median
> var<-apply(data[2:7],2,var)
> var

> # Gamma-Koeffizient
> library(vcdExtra)
> tab1<-table(data$item_1, data$item_2)
> GKgamma(tab1, level=0.95)
```

Das Vorgehen bei der Berechnung der deskriptiven Statistiken und des Gamma-Koeffizienten gleicht im Wesentlichen dem Vorgehen, das im Kapitel 2 dargestellt wurde. Neu hinzu kommt die Berechnung des Medians.

3.3 Berechnung des relativen Informationsgehalts

```
> item_1_factor<- factor(data$item_1, labels=c('1','2','3','4','5'))
```

Item 1 wird in ein Objekt des Typs Faktor umgewandelt.

```
> item_1_h<-prop.table(table(item_1_factor))
```

Die relativen Häufigkeiten der Kategorienwahlen werden berechnet.

```
> item_1_i<-(-1/log(5))*sum((item_1_h*log(item_1_h)))
```

Der relative Informationsgehalt des Items wird berechnet.

3.4 Das Partial-Credit-Modell

Zur psychometrischen Analyse mit dem Partial-Credit-Modell (PCM) kann das Paket `eRm` verwendet werden. Die Analyseschritte und die Syntax unterscheiden sich im Wesentlichen nicht von denjenigen, die schon in dem Abschnitt zum dichotomen Rasch-Modell (Kapitel 2) dargestellt wurden. Es werden hier nur die abweichenden Punkte erläutert.

```
> pcm<-PCM(data.mat)
> summary(pcm)
```

Zur Schätzung der Parameter des Partial-Credit-Modells wird der Funktion `PCM()` die Datenmatrix als Argument übergeben. Die Ergebnisse der Analyse werden mit der Funktion `summary()` ausgegeben.

```
Results of PCM estimation:

Call:  PCM(X = data.mat)

Conditional log-likelihood: -2387.655
Number of iterations: 36
Number of parameters: 23

Item (Category) Difficulty Parameters (eta): with 0.95 CI:
          Estimate Std. Error lower CI upper CI
item_1.c2   -1.541      0.238   -2.007   -1.075
item_1.c3   -0.651      0.243   -1.128   -0.174
item_1.c4    1.054      0.264    0.536    1.573
item_2.c1   -1.398      0.220   -1.828   -0.968
[…]
Item Easiness Parameters (beta) with 0.95 CI:
               Estimate Std. Error lower CI upper CI
beta item_1.c1    1.306      0.245    0.826    1.787
beta item_1.c2    1.541      0.238    1.075    2.007
```

```
beta item_1.c3    0.651      0.243    0.174    1.128
beta item_1.c4   -1.054      0.264   -1.573   -0.536
[…]
```

Die Itemparameter wurden mit der Conditional-Maximum-Likelihood-Methode geschätzt. Die Log-Likelihood auf Basis des angepassten bedingten Partial-Credit-Modells beträgt -2387.655. Die Parameterschätzung konvergierte nach 36 Iterationen, insgesamt wurden 23 Parameter geschätzt. In der oberen Hälfte des Outputs werden die mit der Parametrisierung von Mair und Hatzinger (2007) geschätzten Kategorien-Schwierigkeiten inklusive Standardfehler und Konfidenzintervalle ausgegeben. In der unteren Hälfte des Outputs stehen die entsprechenden geschätzten Kategorien-Leichtigkeiten und deren Standardfehler und Konfidenzintervalle. Das Objekt `pcm` bildet den Ausgangspunkt für die weiteren anstehenden Analysen.

Die Parameter in der Ausgabe sind nicht mit den klassischen Schwellenparametern zu verwechseln. Diese ergeben sich jedoch durch eine lineare Transformation. Für Details sei die Original-Dokumentation zum Paket `eRm` empfohlen.

```
> thresholds(pcm)
```

Es werden die Schwellenparameter des PCM und Lokalisationen der Items (mittlere Schwellenparameter, Itemschwierigkeiten) auf der latenten Variablen berechnet.

```
Design Matrix Block 1:
       Location Threshold 1 Threshold 2 Threshold 3 Threshold 4
item_1  0.26360    -1.30633    -0.23456     0.89008     1.70523
item_2  0.54355    -1.39812     0.20236     1.27090     2.09906
item_3  0.74698    -0.48717     0.49736     0.86477     2.11297
item_4 -0.33903    -2.03104    -0.71493     0.25209     1.13777
item_5  1.70473    -0.01890     1.10021     2.12426     3.61335
item_6  0.33337    -1.72420    -0.06733     0.91611     2.20890
```

Die Schwellenparameter der Items und deren Lokalisationen werden ausgegeben. Der Teil der Ausgabe „Design Matrix Block 1" weist darauf hin, dass das Paket `eRm` sich auch dazu eignet, das linear-logistische Testmodell (s. z.B. Fischer, 1995) anzupassen.

3.5 Analysen mit dem generalisierten Partial-Credit-Modell

Das generalisierte Partial-Credit-Modell kann mit dem `R`-Paket `ltm` angepasst werden.

```
> library(ltm)
> gpcm<-gpcm(data.mat)
```

```
> summary(gpcm)
```

Die Ergebnisse der Schätzung werden ausgegeben.

```
Call:
gpcm(data = data.mat)

Model Summary:
   log.Lik     AIC      BIC
 -3850.295 7760.59 7886.968

Coefficients:
$item_1
         value std.err z.value
Catgr.1 -1.744   0.149 -11.720
Catgr.2 -0.997   0.102  -9.736
Catgr.3 -0.162   0.089  -1.814
Catgr.4  0.582   0.091   6.424
Dscrmn   2.049   0.215   9.527

[…]


$item_6
         value std.err z.value
Catgr.1 -2.622   0.343  -7.645
Catgr.2 -1.123   0.195  -5.766
Catgr.3 -0.136   0.164  -0.832
Catgr.4  0.987   0.179   5.505
Dscrmn   0.792   0.086   9.228


Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.028
optimizer: nlminb
```

Für jedes Item werden der geschätzte itemspezifische Diskriminationsparameter und die geschätzten Schwellen ausgegeben. Die übrigen Angaben wurden bereits weiter oben im Abschnitt zum Birnbaum-Modell besprochen.

```
> GoF.gpcm(gpcm, B=300)
```

Zusätzlich wird die Anpassungsgüte des Modells mittels einer Bootstrap-Goodness-of-Fit-Statistik bewertet. Insgesamt werden 300 Bootstrap-Samples aus dem angepassten Modell angefordert.

```
Parametric Bootstrap Approximation to Pearson chi-squared Goodness-
of-Fit Measure

Call:
gpcm(data = data.mat)

Tobs: 279726.7
# data-sets: 301
p-value: 0.003
```

Die Ergebnisse zeigen an, dass die Hypothese der Modellpassung verworfen werden muss (p = 0.003). Der Wert der χ^2-Statistik für die beobachteten Daten beträgt 279726.7.

3.6 Analyse der Items 1, 3 und 4

```
# Auswahl der Items 1, 3 und 4
> data.sub<-data.mat[,c(1,3,4)]
```

Aus dem Gesamtdatensatz werden die Items 1, 3 und 4 ausgewählt und in das Data-Frame `data.sub` geschrieben.

```
> pcm.sub<-PCM(data.sub)
> summary(pcm.sub)
```

Das Partial-Credit-Modell wird auf die Items angepasst und die Ergebnisse der Schätzung der Basisparameter werden ausgegeben. Für technische Details zwischen dem Zusammenhang von Basisparametern und Schwellenparametern in `eRm` siehe Mair und Hatzinger (2007).

```
Results of PCM estimation:

Call:  PCM(X = data.sub)

Conditional log-likelihood: -650.7847
Number of iterations: 22
Number of parameters: 11

Item (Category) Difficulty Parameters (eta): with 0.95 CI:
          Estimate Std. Error lower CI upper CI
item_1.c2   -1.593      0.260   -2.102   -1.084
item_1.c3    0.015      0.279   -0.532    0.562
item_1.c4    3.466      0.371    2.739    4.193
item_3.c1   -0.519      0.189   -0.889   -0.148
item_3.c2    0.398      0.233   -0.059    0.854
item_3.c3    2.163      0.309    1.557    2.769
item_3.c4    6.371      0.458    5.473    7.269
item_4.c1   -2.581      0.441   -3.445   -1.718
item_4.c2   -3.315      0.399   -4.098   -2.533
item_4.c3   -2.628      0.340   -3.296   -1.961
item_4.c4   -0.202      0.309   -0.808    0.405

Item Easiness Parameters (beta) with 0.95 CI:
```

```
               Estimate Std. Error lower CI upper CI
beta item_1.c1    1.573      0.264    1.056    2.090
beta item_1.c2    1.593      0.260    1.084    2.102
beta item_1.c3   -0.015      0.279   -0.562    0.532
beta item_1.c4   -3.466      0.371   -4.193   -2.739
beta item_3.c1    0.519      0.189    0.148    0.889
beta item_3.c2   -0.398      0.233   -0.854    0.059
beta item_3.c3   -2.163      0.309   -2.769   -1.557
beta item_3.c4   -6.371      0.458   -7.269   -5.473
beta item_4.c1    2.581      0.441    1.718    3.445
beta item_4.c2    3.315      0.399    2.533    4.098
beta item_4.c3    2.628      0.340    1.961    3.296
beta item_4.c4    0.202      0.309   -0.405    0.808
```

Um die Schwellenparameter zu berechnen wird die Funktion `thresholds()` verwendet.

```
> thresholds(pcm.sub)

Design Matrix Block 1:
       Location Threshold 1 Threshold 2 Threshold 3 Threshold 4
item_4 -0.05044    -2.58135    -0.73394     0.68684     2.42669
item_1  0.86646    -1.57340    -0.01973     1.60797     3.45100
item_3  1.59268    -0.51889     0.91642     1.76580     4.20740
```

```
> rsm.sub<-RSM(data.sub)
> summary(rsm.sub)
```

Das Ratingskalenmodell wird auf die Items angepasst, und die Ergebnisse der Schätzung der Basisparameter werden ausgegeben.

```
Number of iterations: 18
Number of parameters: 5

Item (Category) Difficulty Parameters (eta): with 0.95 CI:
       Estimate Std. Error lower CI upper CI
item_3    0.724      0.064    0.598    0.849
item_4   -0.805      0.068   -0.939   -0.672
Cat 2     1.423      0.218    0.996    1.850
Cat 3     4.189      0.424    3.357    5.020
Cat 4     8.954      0.685    7.612   10.296

Item Easiness Parameters (beta) with 0.95 CI:
               Estimate Std. Error lower CI upper CI
beta item_1.c1   -0.082      0.055   -0.189    0.025
beta item_1.c2   -1.587      0.245   -2.067   -1.106
beta item_1.c3   -4.434      0.459   -5.333   -3.535
beta item_1.c4   -9.281      0.728  -10.707   -7.855
beta item_3.c1   -0.724      0.064   -0.849   -0.598
beta item_3.c2   -2.871      0.277   -3.413   -2.328
beta item_3.c3   -6.360      0.523   -7.384   -5.335
beta item_3.c4  -11.849      0.829  -13.473  -10.225
beta item_4.c1    0.805      0.068    0.672    0.939
beta item_4.c2    0.188      0.230   -0.263    0.638
```

```
beta item_4.c3   -1.772      0.402   -2.560   -0.985
beta item_4.c4   -5.733      0.615   -6.938   -4.527
```

Eine Annahme des Ratingskalenmodells besteht darin, dass die Schwellenabstände für jedes Item identisch sind. Es ist also eine restringierte Form des Partial-Credit-Modells. Weitere Analyseschritte, wie z.B. die Durchführung eines Likelihood-Quotienten-Tests, die Schätzung der Personenparameter und die Bewertung der Residualstatistiken wurden bereits weiter oben im Abschnitt zum Partial-Credit-Modell besprochen. In der Syntax werden zusätzlich die Schwellenparameter mit der Funktion `thresholds()` berechnet und der Funkton `plotPImap()` grafisch dargestellt, um einen Vergleich der Ergebnisse mit denjenigen in Eid und Schmidt (2014) zu erleichtern:

```
 > thresholds(rsm.sub)

Design Matrix Block 1:
       Location Threshold 1 Threshold 2 Threshold 3 Threshold 4
item_4  1.43318    -0.80540     0.61780     1.96007     3.96025
item_1  2.32028     0.08169     1.50490     2.84717     4.84735
item_3  2.96229     0.72371     2.14692     3.48919     5.48937.
```

4 Beispielsyntax für Kapitel 6: Eindimensionale Modelle für metrische Antwortvariablen

4.1 Überblick Gesamtsyntax

Hinweis: Die Analysen in Eid und Schmidt (2014) wurden mit dem Computerprogramm Mplus durchgeführt. Im Folgenden wird gezeigt, wie entsprechende Analysen mit der Programmierumgebung `R` durchgeführt werden können. Unterschiede zu den Resultaten im Buch sind auf die unterschiedlichen Computerprogramme zurückführbar.

```
# Syntax Kapitel 6
library(memisc) # Für den Import von SPSS-Datei
library(psych)  # Für Schiefe und Kurtosis
library(semTools) # Nützliche Tools zur
                  # Strukturgleichungsmodellierung
library(MVN)    # Wird für Mardias Test auf multivariate
                # Normalverteilung gebraucht
library(MBESS)  # Wird zur Schätzung von Konfidenzintervallen
                # von Cronbachs alpha und McDonalds omega
                # benötigt.

# *.sav-file einlesen
data<-as.data.set(spss.system.file('daten-kapitel-6.sav'))

# Umwandlung in den Typ data.frame
data<-data.frame(data)
```

```
# Auswahl der Items 1 bis 6 aus der Datenmatrix
data.mat<-data[,2:7]

# Berechnung der Stichprobenkovarianzmatrix
cov(data.mat)

# Berechnung der Stichprobenkorrelationsmatrix
cor(data.mat)

# Berechnung der Mittelwerte der Items
apply(data.mat,2,mean)

# Berechnung der Standardabweichungen der Items
apply(data.mat,2,sd)

# Berechnung der Schiefe der Items
apply(data.mat,2,skew)

# Berechnung der Kurtosis der Items
apply(data.mat,2,kurtosi)

# Mardia's Test auf multivariate Normalverteilung
mardiaTest(data.mat)

# Laden des Pakets lavaan
library(lavaan)

# Modell essentiell tau-äquivalenter Variablen
m1<-'eta=~item_1+1*item_2+1*item_3+1*item_4+1*item_5+1*item_6'
fit1<-sem(m1, data=data.mat, meanstructure=TRUE, estimator="MLR")
summary(fit1, fit.measures=TRUE, standardized=TRUE)

# Ausgabe der Konfidenzintervalle der Schätzer
parameterEstimates(fit1, ci=TRUE, level=0.95)


# Schätzung diverser Reliabilitätskoeffizienten
reliability(fit1)

# Schätzung von Cronbachs alpha inlusive
# des Konfidenzintervalls
ci.reliability(data.mat, type="alpha", conf.level=0.95)


# Multigruppenanalyse
measurementInvariance(m1, data, group="geschl")

# Modell tau-äquivalenter Variablen
m2<-'eta=~item_1+1*item_2+1*item_3+1*item_4+1*item_5+1*item_6
      item_1~v1*1
      item_2~v1*1
      item_3~v1*1
      item_4~v1*1
      item_5~v1*1
      item_6~v1*1'

fit2<-sem(m2, data=data.mat, meanstructure=TRUE, estimator="MLR")
```

```
summary(fit2, fit.measures=TRUE, standardized=TRUE)

# Modell essentiell tau-paralleler Variablen
m3<-'eta=~item_1+1*item_2+1*item_3+1*item_4+1*item_5+1*item_6
      item_1~~v1*item_1
      item_2~~v1*item_2
      item_3~~v1*item_3
      item_4~~v1*item_4
      item_5~~v1*item_5
      item_6~~v1*item_6'

fit3<-sem(m3, data=data.mat, meanstructure=TRUE, estimator="MLR")
summary(fit3, fit.measures=TRUE, standardized=TRUE)

# Modell tau-paralleler Variablen
m4<-'eta=~item_1+1*item_2+1*item_3+1*item_4+1*item_5+1*item_6
      item_1~v1*1
      item_2~v1*1
      item_3~v1*1
      item_4~v1*1
      item_5~v1*1
      item_6~v1*1
      item_1~~v2*item_1
      item_2~~v2*item_2
      item_3~~v2*item_3
      item_4~~v2*item_4
      item_5~~v2*item_5
      item_6~~v2*item_6'

fit4<-sem(m4, data=data.mat, meanstructure=TRUE, estimator="MLR")
summary(fit4, fit.measures=TRUE, standardized=TRUE)

# Das Modell tau-kongenerischer Variablen
m5<-'eta=~item_1+item_2+item_3+item_4+item_5+item_6'

fit5<-sem(m5, data=data.mat, meanstructure=TRUE, estimator="MLR")
summary(fit5, fit.measures=TRUE, standardized=TRUE)

# Modellvergleich
fit1_ml<-sem(m1, data=data.mat, meanstructure=TRUE)
fit2_ml<-sem(m2, data=data.mat, meanstructure=TRUE)
anova(fit1_ml, fit2_ml)

# Extraktion der Personenwerte (Faktorwerte)
predict(fit1)
```

4.2 Das Einlesen der Daten und die Berechnung einiger deskriptiver Statistiken

```
> library(memisc) # Für den Import von SPSS-Datei
> library(psych)  # Für Schiefe und Kurtosis
> library(semTools) # Nützliche Tools zur
                  # Strukturgleichungsmodellierung
> library(MVN)    # Wird für Mardia's Test auf multivariate
                  # Normalverteilung gebraucht
> library(MBESS)  # Wird zur Schätzung von Konfidenzintervallen
```

```
                    # von Cronbachs alpha und McDonalds omega
                    # benötigt.
```

Die SPSS-Datei wird eingelesen und in den Objekt-Typ `data.frame` umgewandelt. Zudem werden lediglich die Items (Spalten 2 bis 7) in das Objekt `data.mat` geschrieben. Die erste Spalte der Datei enthält die Variable Geschlecht, die anderen Spalten enthalten logarithmierte Reaktionszeiten.

```
> # *.sav-file einlesen
> data<-as.data.set(spss.system.file('daten-kapitel-6.sav'))

> # Umwandlung in den Typ data.frame
> data<-data.frame(data)


> # Auswahl der Items 1 bis 6 aus der Datenmatrix
> data.mat<-data[,2:7]

> # Berechnung der Stichprobenkovarianzmatrix
> cov(data.mat)

           item_1     item_2     item_3     item_4     item_5     item_6
item_1 0.12892429 0.06372098 0.06387506 0.06694482 0.04484940 0.05684147
item_2 0.06372098 0.13601262 0.06692212 0.06995322 0.06573506 0.05659685
item_3 0.06387506 0.06692212 0.15389888 0.08154838 0.06597039 0.06510418
item_4 0.06694482 0.06995322 0.08154838 0.16613492 0.07229804 0.07579945
item_5 0.04484940 0.06573506 0.06597039 0.07229804 0.14207685 0.05470620
item_6 0.05684147 0.05659685 0.06510418 0.07579945 0.05470620 0.14692864
```

Es wird die Stichprobenkovarianzmatrix der Items berechnet. Insgesamt zeigen sich positive Kovarianzen zwischen den Items.

```
> # Berechnung der Stichprobenkorrelationsmatrix
> cor(data.mat)

          item_1    item_2    item_3    item_4    item_5    item_6
item_1 1.0000000 0.4811999 0.4534675 0.4574242 0.3313810 0.4129949
item_2 0.4811999 1.0000000 0.4625539 0.4653585 0.4728744 0.4003589
item_3 0.4534675 0.4625539 1.0000000 0.5099964 0.4461385 0.4329502
item_4 0.4574242 0.4653585 0.5099964 1.0000000 0.4705810 0.4851570
item_5 0.3313810 0.4728744 0.4461385 0.4705810 1.0000000 0.3786358
item_6 0.4129949 0.4003589 0.4329502 0.4851570 0.3786358 1.0000000
```

Es wird die Stichprobenkorrelationsmatrix zwischen den Items berechnet. Insgesamt zeigen sich relativ hohe und homogene Interkorrelationen.

```
> apply(data.mat,2,mean)
> apply(data.mat,2,sd)
> apply(data.mat,2,skew)
> apply(data.mat,2,kurtosi)
```

Die Mittelwerte, Standardabweichung, Schiefe und die Kurtosis der Items werden berechnet.

4.3 Mardia's Test auf multivariate Normalverteilung

```
> mardiaTest(data.mat)

Mardia's Multivariate Normality Test
---------------------------------------
   data : data.mat

   g1p            : 3.743293
   chi.skew       : 148.484
   p.value.skew   : 2.624706e-10

   g2p            : 52.6799
   z.kurtosis     : 3.684338
   p.value.kurt   : 0.000229298

   chi.small.skew : 150.8968
   p.value.small  : 1.203472e-10

   Result          : Data are not multivariate normal.
---------------------------------------
```

Es werden Tests auf multivariate Normalverteilung durchgeführt. Die Tests auf multivariate Schiefe (`p.value.skew = 2.624706e-10`), multivariate Schiefe für kleine Stichproben (`p.value.small = 0.000229298`) und multivariate Kurtosis (`p.value.kurt = 0.000229298`) werden signifikant, so dass die Nullhypothese der multivariaten Normalverteilung verworfen werden muss.

4.4 Das Modell essenziell τ-äquivalenter Variablen

```
> m1<-'eta=~item_1+1*item_2+1*item_3+1*item_4+1*item_5+1*item_6'
```

Das Modell wird in der `lavaan`-Syntax mit einem String spezifiziert, der den Namen `m1` erhält. Die Terme `1*item_2` bis `1*item_6` geben zum Ausdruck, dass die Ladungen dieser Items auf 1 restringiert werden. Die Ladung des ersten Items wird per Voreinstellung auf den Wert 1 restringiert.

```
> fit1<-sem(m1, data=data.mat, meanstructure=TRUE, estimator="MLR")
```

Der Modell-String `m1` wird der Funktion `sem()` zusammen mit dem Namen der Datenmatrix übergeben. Das Argument `meanstructure=TRUE` bewirkt die Ausgabe der Leichtigkeitsparameter der itemspezifischen Regressionen. Zudem wurde die Maximum-Likelihood-Schätzung mit robusten Standardfehlern über `estimator="MLR"` angefordert. Mit dem Argument `estimator="ML"` kann die gewöhnliche Maximum-Likelihood-Schätzung angefordert werden.

```
> summary(fit1, fit.measures=TRUE, standardized=TRUE)
```

Das Ergebnis der Schätzung wird auf der Konsole ausgegeben. Über die Argumente `fit.measures=TRUE` und `standardized=TRUE` werden Fit-Statistiken und zusätzlich standardisierte Koeffizienten angefordert. Es wird sehr empfohlen, den Umgang mit dem Paket `lavaan` anhand dessen Dokumentation zu vertiefen.

```
lavaan (0.5-14) converged normally after  12 iterations

  Number of observations                           238

  Estimator                                         ML      Robust
  Minimum Function Test Statistic               16.949      16.161
  Degrees of freedom                                14          14
  P-value (Chi-square)                           0.259       0.304
  Scaling correction factor                                  1.049
    for the Yuan-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              435.847     424.389
  Degrees of freedom                                15          15
  P-value                                        0.000       0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.993       0.995
  Tucker-Lewis Index (TLI)                       0.992       0.994

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)               -435.870    -435.870
  Scaling correction factor                                  1.127
    for the MLR correction
  Loglikelihood unrestricted model (H1)       -427.396    -427.396
  Scaling correction factor                                  1.087
    for the MLR correction

  Number of free parameters                         13          13
  Akaike (AIC)                                 897.740     897.740
  Bayesian (BIC)                               942.880     942.880
  Sample-size adjusted Bayesian (BIC)          901.674     901.674

Root Mean Square Error of Approximation:

  RMSEA                                          0.030       0.025
  90 Percent Confidence Interval          0.000  0.073       0.000  0.069
  P-value RMSEA <= 0.05                          0.737       0.781

Standardized Root Mean Square Residual:

  SRMR                                           0.053       0.053
```

Informationen zur Konvergenz und eine Auswahl an Fit-Indizes und Fit-Statistiken werden ausgegeben. Insgesamt zeigt sich eine recht gute Passung des Modells. Ferner werden die geschätzten Parameter ausgegeben.

```
Parameter estimates:

  Information                                 Observed
  Standard Errors                   Robust.huber.white

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  eta =~
    item_1            1.000                               0.253    0.682
    item_2            1.000                               0.253    0.689
    item_3            1.000                               0.253    0.664
    item_4            1.000                               0.253    0.656
    item_5            1.000                               0.253    0.657
    item_6            1.000                               0.253    0.650
```

Alle Ladungen der Items wurden auf 1 fixiert, daher werden keine Standardfehler und entsprechende *z*-Statistiken nach Wald ausgegeben. In der Spalte `Std.lv` sind die Ladungen angegeben, die auf Basis der auf eine Varianz von 1 standardisierten latenten Variable berechnet wurden. Die Spalte `Std.all` gibt die Ladungen an, die sich ergeben, wenn alle Variablen standardisiert werden. Diese entsprechen den Korrelationen der Items mit der latenten Variablen.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Intercepts:
    item_1            1.504    0.023   64.757    0.000    1.504    4.047
    item_2            1.423    0.024   59.647    0.000    1.423    3.870
    item_3            1.392    0.025   54.862    0.000    1.392    3.647
    item_4            1.305    0.026   49.486    0.000    1.305    3.376
    item_5            1.346    0.024   55.221    0.000    1.346    3.488
    item_6            1.306    0.025   52.662    0.000    1.306    3.350
    eta               0.000                               0.000    0.000
```

Die Intercepts sind die geschätzten Itemleichtigkeiten. Wald-Tests zur Überprüfung der Hypothese, dass die entsprechenden Parameter in der Population gleich 0 sind, und die standardisierten Lösungen werden ausgegeben. Der Mittelwert der latenten Variablen (`eta`) ist auf 0 fixiert.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Variances:
    item_1            0.074    0.008                      0.074    0.535
    item_2            0.071    0.008                      0.071    0.525
    item_3            0.081    0.009                      0.081    0.559
    item_4            0.085    0.010                      0.085    0.570
    item_5            0.085    0.010                      0.085    0.569
    item_6            0.088    0.009                      0.088    0.577
    eta               0.064    0.011                      1.000    1.000
```

Die geschätzten Varianzen der Items sowie deren Standardfehler und die standardisierten Koeffizienten werden ausgegeben. Im Rahmen der Klassischen Testtheorie sind diese Varianzen die Fehlervarianzen der Items. Die vollständig standardisierte Lösung (`Std.all`) ist hier von besonderem Interesse, da sich auf ihrer Basis die Item-Reliabilität berechnen lässt (Rel(Item 1) = 1-0.535, etc.).

```
> parameterEstimates(fit1, ci=TRUE, level=0.95)
```

Die Parameterschätzer werden separat angefordert. Die 95%-Konfidenzintervalle der geschätzten Parameter werden über die Argumente `ci=TRUE` und `level=0.95` zusätzlich ausgegeben.

```
      lhs op    rhs   est    se      z pvalue ci.lower ci.upper
1     eta =~ item_1 1.000 0.000     NA     NA    1.000    1.000
2     eta =~ item_2 1.000 0.000     NA     NA    1.000    1.000
3     eta =~ item_3 1.000 0.000     NA     NA    1.000    1.000
4     eta =~ item_4 1.000 0.000     NA     NA    1.000    1.000
5     eta =~ item_5 1.000 0.000     NA     NA    1.000    1.000
6     eta =~ item_6 1.000 0.000     NA     NA    1.000    1.000
7  item_1 ~~ item_1 0.074 0.008  9.383      0    0.058    0.089
8  item_2 ~~ item_2 0.071 0.008  9.197      0    0.056    0.086
9  item_3 ~~ item_3 0.081 0.009  9.544      0    0.065    0.098
10 item_4 ~~ item_4 0.085 0.010  8.584      0    0.066    0.105
11 item_5 ~~ item_5 0.085 0.010  8.908      0    0.066    0.103
12 item_6 ~~ item_6 0.088 0.009  9.564      0    0.070    0.106
13    eta ~~    eta 0.064 0.011  5.822      0    0.043    0.086
14 item_1 ~1        1.504 0.023 64.757      0    1.458    1.550
15 item_2 ~1        1.423 0.024 59.647      0    1.376    1.470
16 item_3 ~1        1.392 0.025 54.862      0    1.342    1.442
17 item_4 ~1        1.305 0.026 49.486      0    1.253    1.356
18 item_5 ~1        1.346 0.024 55.221      0    1.299    1.394
19 item_6 ~1        1.306 0.025 52.662      0    1.257    1.354
20    eta ~1        0.000 0.000     NA     NA    0.000    0.000
```

In den ersten drei Spalten werden die geschätzten Parameter in der `lavaan`-Syntax zur Modellspezifikation dargestellt. Die folgenden Spalten enthalten die Schätzer der entsprechenden Komponenten (`est`), die geschätzten Standardfehler (`se`), die *z*- und *p*-Werte (`z`, `pvalue`) sowie die angeforderten Konfidenzintervalle (`ci.lower`, `ci.upper`). Ein `NA` gibt an, dass ein entsprechender Wert aufgrund der Modellspezifikation nicht geschätzt wurde. Dies macht Sinn, da im vorliegenden Modell die Ladungen auf 1 fixiert wurden.

```
> reliability(fit1)
    alpha     omega    omega2    omega3 
0.8275283 0.8269606 0.8269606 0.8247808 
```

Die Reliabilitätskoeffizienten werden ausgegeben. Diese sind Cronbachs α (`alpha`), Raykovs ω (`omega`), Bentlers ω (`omega2`) und McDonalds ω (`omega3`). Nähere Informationen inklusive Formeln und Literatur finden sich in der `R`-Hilfe (`?reliability`). Für das Modell der essenziell τ-äquivalenten Messung ist lediglich Cronbachs α von Belang. Im Gegensatz zu den im Lehrbuch berichteten Koeffizienten wird bei der Berechnung dieser Koeffizienten z.T. nicht auf die vom Modell implizierten Varianzen und Kovarianzen zurückgegriffen, sondern auf die beobachteten Varianzen und Kovarianzen, wodurch sich Unterschiede ergeben.

```
> ci.reliability(data.mat, type="alpha", conf.level=0.95)
```

```
$est
[1] 0.8275283

$se
[1] 0.01738553

$ci.lower
[1] 0.7934532

$ci.upper
[1] 0.8616033

$conf.level
[1] 0.95

$est.type
[1] "alpha"

$analysis.type
[1] "analytic"

$interval.type
[1] "normal-theory"

$call
ci.reliability(data = data.mat, type = "alpha", conf.level = 0.95)

$boot
NULL
```

Im Paket `MBESS` befindet sich die Funktion `ci.reliability()`, mit der sich Konfidenzintervalle für Reliabilitätskoeffizienten berechnen lassen. Diese Funktion bietet zahlreiche Zusatzoptionen, die der Dokumentation der Funktion entnommen werden können (`?ci.reliability()`). Hier wird das 95%-Konfidenzintervall für Cronbachs α angefordert.

Multigruppenanalyse

```
> measurementInvariance(m1, data, group="geschl")
```

Zur Prüfung der Messinvarianz in unterschiedlichen Subgruppen (hier Geschlecht), wird eine Mehrgruppenanalyse durchgeführt.

```
Multig Measurement invariance tests:
Model 1: configural invariance:
  chisq      df  pvalue     cfi   rmsea     bic
 27.735  28.000   0.479   1.000   0.000 994.493

Model 2: weak invariance (equal loadings):
  chisq      df  pvalue     cfi   rmsea     bic
 27.735  28.000   0.479   1.000   0.000 994.493
```

```
[Model 1 versus model 2]
  delta.chisq      delta.df delta.p.value
            0             0             1
    delta.cfi
            0
```

Das Modell der konfiguralen Invarianz (identische Faktorstruktur in den Subgruppen) wird gegen das Modell der *schwachen Invarianz* (identische Faktorladungen in den Subgruppen) getestet. Im vorliegenden Fall sind beide Modelle identisch, da die Faktorladungen auf 1 fixiert wurden.

```
Model 3: strong invariance (equal loadings + intercepts):
  chisq      df  pvalue     cfi   rmsea     bic
 35.089  33.000   0.369   0.995   0.023 974.485

[Model 1 versus model 3]
  delta.chisq      delta.df delta.p.value
        7.354         5.000         0.196
    delta.cfi
        0.005

[Model 2 versus model 3]
  delta.chisq      delta.df delta.p.value
        7.354         5.000         0.196
    delta.cfi
        0.005
```

Das Modell der *starken Invarianz* (identische Ladungen und Itemleichtigkeiten in den Subgruppen) wird angepasst. Dieses Modell entspricht der Annahme der Gleichheit der Leichtigkeitsparamter des Modells essenziell τ-äquivalenter Variablen in beiden Subpopulationen. Dieses Modell (Modell 3) wird gegen das Modell der konfiguralen Invarianz (Modell 1) und das Modell der schwachen Invarianz (Modell 2) getestet. Beide Tests werden nicht signifikant.

```
Model 4: equal loadings + intercepts + means:
  chisq      df  pvalue     cfi   rmsea     bic
 35.999  34.000   0.375   0.995   0.022 969.922

[Model 1 versus model 4]
  delta.chisq      delta.df delta.p.value
        8.264         6.000         0.219
    delta.cfi
        0.005

[Model 3 versus model 4]
  delta.chisq      delta.df delta.p.value
         0.91          1.00          0.34
    delta.cfi
         0.00
```

Das Modell der *strikten Invarianz* (identische Ladungen, Itemleichtigkeiten und Varianzen) wird angepasst und gegen die Modelle 1 und 3 getestet. Beide Tests werden nicht signifikant. Aus diesen Modellvergleichen lässt sich schließen, dass das Modell der strikten Invarianz in den Geschlechtsgruppen die Daten am sparsamsten erklärt. Für nähere Hinweise und weiterführende Literatur sei auf die `R`-Hilfe der Funktion verwiesen `(?measurementInvariance)`.

4.5 Das Modell τ-äquivalenter Variablen

```
# Modell tau-äquivalenter Variablen
m2<-'eta=~item_1+1*item_2+1*item_3+1*item_4+1*item_5+1*item_6
      item_1~v1*1
      item_2~v1*1
      item_3~v1*1
      item_4~v1*1
      item_5~v1*1
      item_6~v1*1'

fit2<-sem(m2, data=data.mat, meanstructure=TRUE)
summary(fit2)
```

Bei der Verwendung des Modells τ-äquivalenter Messung wird davon ausgegangen, dass die Ladungen der Items alle 1 sind und dass die Items identische Leichtigkeitsparameter besitzen. Die Fehlervarianzen können frei variieren. Die Spezifikation der Restriktion der identischen Leichtigkeitsparameter der Items erfolgt durch folgenden Syntax-Block:

```
item_1~v1*1
item_2~v1*1
item_3~v1*1
item_4~v1*1
item_5~v1*1
item_6~v1*1
```

`v1` ist ein sogenanntes Parameter-Label. Da die Syntax-Zeilen für alle Items dasselbe Label verwenden, werden für jedes Item identische Leichtigkeitsparameter (Achsenabschnitte) geschätzt. Details sind der `lavaan`-Dokumentation zu entnehmen.

```
> fit2<-sem(m2, data=data.mat, meanstructure=TRUE, estimator="MLR")
```

Das Modell wird an die Daten angepasst. Es werden die Leichtigkeitsparameter (Achsenabschnitte) angefordert (`meanstructure=TRUE`).

```
> summary(fit2, fit.measures=TRUE, standardized=TRUE)
```

Es werden zusätzlich zu den geschätzten Parametern die Fit-Statistiken und die standardisierten Lösungen ausgegeben.

```
lavaan (0.5-14) converged normally after  16 iterations

  Number of observations                           238

  Estimator                                         ML      Robust
  Minimum Function Test Statistic              100.116      95.652
  Degrees of freedom                                19          19
  P-value (Chi-square)                           0.000       0.000
  Scaling correction factor                                  1.047
    for the Yuan-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              435.847     424.389
  Degrees of freedom                                15          15
  P-value                                        0.000       0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.807       0.813
  Tucker-Lewis Index (TLI)                       0.848       0.852

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)               -477.454    -477.454
  Scaling correction factor                                  1.182
    for the MLR correction
  Loglikelihood unrestricted model (H1)       -427.396    -427.396
  Scaling correction factor                                  1.087
    for the MLR correction

  Number of free parameters                          8           8
  Akaike (AIC)                                 970.908     970.908
  Bayesian (BIC)                               998.686     998.686
  Sample-size adjusted Bayesian (BIC)          973.329     973.329

Root Mean Square Error of Approximation:

  RMSEA                                          0.134       0.130
  90 Percent Confidence Interval          0.109  0.160       0.105  0.156
  P-value RMSEA <= 0.05                          0.000       0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.111       0.111
```

Insgesamt deuten die Fit-Statistiken darauf hin, dass die Annahme identischer Leichtigkeitsparameter zu restriktiv ist und die Daten nicht gut beschreibt.

```
Information                                    Observed
  Standard Errors                      Robust.huber.white

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  eta =~
    item_1            1.000                               0.252    0.637
```

```
    item_2          1.000                                 0.252    0.683
    item_3          1.000                                 0.252    0.663
    item_4          1.000                                 0.252    0.639
    item_5          1.000                                 0.252    0.655
    item_6          1.000                                 0.252    0.634
```

Die Ladungen aller Items wurden auf 1 fixiert, von daher wurden keine Parameter geschätzt.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Intercepts:
    item_1   (v1)     1.381    0.019   72.678    0.000    1.381    3.485
    item_2   (v1)     1.381    0.019   72.678    0.000    1.381    3.736
    item_3   (v1)     1.381    0.019   72.678    0.000    1.381    3.630
    item_4   (v1)     1.381    0.019   72.678    0.000    1.381    3.496
    item_5   (v1)     1.381    0.019   72.678    0.000    1.381    3.585
    item_6   (v1)     1.381    0.019   72.678    0.000    1.381    3.471
    eta               0.000                               0.000    0.000
```

Konsistent mit der Modellspezifikation sind die geschätzten Leichtigkeitsparameter identisch. Der Mittelwert der latenten Variablen ist per Voreinstellung auf 0 fixiert.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Variances:
    item_1            0.093    0.011                      0.093    0.594
    item_2            0.073    0.008                      0.073    0.534
    item_3            0.081    0.009                      0.081    0.560
    item_4            0.092    0.009                      0.092    0.592
    item_5            0.085    0.009                      0.085    0.571
    item_6            0.095    0.009                      0.095    0.598
    eta               0.064    0.011                      1.000    1.000
```

Die geschätzten Fehlervarianzen der Items und die Varianz der latenten Variablen, die Standardfehler sowie die standardisierten Lösungen werden ausgegeben.

Weitere Anmerkungen

Natürlich können die Funktionen zur Berechnung der Reliabilitätskoeffizienten und der Konfidenzintervalle dieser Koeffizienten, die im Rahmen des Modells essenziell τ-äquivalenter Variablen behandelt wurden, auch auf das Modell τ-äquivalenter Variablen angewendet werden. Gleiches gilt für die Berechnung der Konfidenzintervalle der Schätzer und die weiteren Modelle, die im Folgenden dargestellt werden.

4.6 Das Modell essenziell τ-paralleler Messung

```
> m3<-'eta=~item_1+1*item_2+1*item_3+1*item_4+1*item_5+1*item_6
      item_1~~v1*item_1
      item_2~~v1*item_2
      item_3~~v1*item_3
      item_4~~v1*item_4
      item_5~~v1*item_5
```

```
item_6~~v1*item_6'
```

Bei der Verwendung des Modells essenziell τ-paralleler Messung wird davon ausgegangen, dass die Fehlervarianzen der Items identisch sind. Die Leichtigkeitsparameter können frei variieren und die Ladungen sind alle gleich 1.

Die Restriktion der Fehlervarianzen erfolgt über den Syntax-Block:

```
item_1~~v1*item_1
item_2~~v1*item_2
item_3~~v1*item_3
item_4~~v1*item_4
item_5~~v1*item_5
item_6~~v1*item_6
```

Es wird ein Parameter-Label `v1` verwendet, was zur Folge hat, dass die geschätzten Fehlervarianzen der einzelnen Items identisch sind.

```
> fit3<-sem(m3, data=data.mat, meanstructure=TRUE, estimator="MLR")
> summary(fit3, fit.measures=TRUE, standardized=TRUE)
```

Das Modell wird an die Daten angepasst, und eine detaillierte Ausgabe wird angefordert.

```
  Estimator                                         ML      Robust
  Minimum Function Test Statistic               19.886      19.166
  Degrees of freedom                                19          19
  P-value (Chi-square)                           0.401       0.446
  Scaling correction factor                                  1.038
    for the Yuan-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              435.847     424.389
  Degrees of freedom                                15          15
  P-value                                        0.000       0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.998       1.000
  Tucker-Lewis Index (TLI)                       0.998       1.000

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)               -437.339    -437.339
  Scaling correction factor                                  1.203
    for the MLR correction
  Loglikelihood unrestricted model (H1)       -427.396    -427.396
  Scaling correction factor                                  1.087
    for the MLR correction

  Number of free parameters                          8           8
```

```
  Akaike (AIC)                                   890.677     890.677
  Bayesian (BIC)                                 918.455     918.455
  Sample-size adjusted Bayesian (BIC)            893.098     893.098

Root Mean Square Error of Approximation:

  RMSEA                                            0.014       0.006
  90 Percent Confidence Interval          0.000  0.059       0.000  0.056
  P-value RMSEA <= 0.05                            0.884       0.908

Standardized Root Mean Square Residual:

  SRMR                                             0.059       0.059
```

Insgesamt deuten die Fit-Statistiken darauf hin, dass das Modell gut passt.

```
Parameter estimates:

  Information                                 Observed
  Standard Errors                   Robust.huber.white

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  eta =~
    item_1            1.000                               0.254    0.667
    item_2            1.000                               0.254    0.667
    item_3            1.000                               0.254    0.667
    item_4            1.000                               0.254    0.667
    item_5            1.000                               0.254    0.667
    item_6            1.000                               0.254    0.667
```

Gemäß der Modellspezifikation sind alle Diskriminationsparameter (Ladungen) auf 1 fixiert.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Intercepts:
    item_1            1.504    0.023   64.757    0.000    1.504    3.949
    item_2            1.423    0.024   59.647    0.000    1.423    3.736
    item_3            1.392    0.025   54.862    0.000    1.392    3.655
    item_4            1.305    0.026   49.486    0.000    1.305    3.426
    item_5            1.346    0.024   55.221    0.000    1.346    3.535
    item_6            1.306    0.025   52.662    0.000    1.306    3.428
    eta               0.000                               0.000    0.000
```

Die geschätzten Itemleichtigkeiten dürfen variieren. Insgesamt ist Item 1 am leichtesten. Per Voreinstellung ist der Mittelwert der latenten Variablen auf 0 fixiert.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Variances:
    item_1   (v1)     0.081    0.004                      0.081    0.556
    item_2   (v1)     0.081    0.004                      0.081    0.556
    item_3   (v1)     0.081    0.004                      0.081    0.556
    item_4   (v1)     0.081    0.004                      0.081    0.556
    item_5   (v1)     0.081    0.004                      0.081    0.556
    item_6   (v1)     0.081    0.004                      0.081    0.556
    eta               0.064    0.011                      1.000    1.000
```

In Einklang mit der Modellspezifikation sind die geschätzten Fehlervarianzen der Items identisch.

4.7 Das Modell τ-paralleler Variablen

```
> m4<-'eta=~item_1+1*item_2+1*item_3+1*item_4+1*item_5+1*item_6
      item_1~v1*1
      item_2~v1*1
      item_3~v1*1
      item_4~v1*1
      item_5~v1*1
      item_6~v1*1
      item_1~~v2*item_1
      item_2~~v2*item_2
      item_3~~v2*item_3
      item_4~~v2*item_4
      item_5~~v2*item_5
      item_6~~v2*item_6'
```

Die Verwendung des Modells τ-paralleler Messung impliziert die Annahme der Gleichheit der Fehlervarianzen und der Leichtigkeitsparameter (`Intercepts`). Zur Spezifikation dieser Restriktionen werden die Intercepts (z.B. `item_1~v1*1`) und die Varianzkomponenten (z.B. `item_1~~v2*item_1`) mithilfe von jeweils zwei Wertelabels (`v1` und `v2`) gleichgesetzt.

```
> fit4<-sem(m4, data=data.mat, meanstructure=TRUE, estimator="MLR")
> summary(fit4, fit.measures=TRUE, standardized=TRUE)
```

Das Modell wird an die Daten angepasst, und eine detaillierte Ausgabe wird angefordert.

```
  Estimator                                         ML       Robust
  Minimum Function Test Statistic              104.462      101.211
  Degrees of freedom                                24           24
  P-value (Chi-square)                           0.000        0.000
  Scaling correction factor                                   1.032
    for the Yuan-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              435.847      424.389
  Degrees of freedom                                15           15
  P-value                                        0.000        0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.809        0.811
  Tucker-Lewis Index (TLI)                       0.881        0.882

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)               -479.627     -479.627
  Scaling correction factor                                   1.523
```

```
    for the MLR correction
  Loglikelihood unrestricted model (H1)       -427.396    -427.396
  Scaling correction factor                                  1.087
    for the MLR correction

  Number of free parameters                          3           3
  Akaike (AIC)                                 965.254     965.254
  Bayesian (BIC)                               975.670     975.670
  Sample-size adjusted Bayesian (BIC)          966.161     966.161

Root Mean Square Error of Approximation:

  RMSEA                                          0.119       0.116
  90 Percent Confidence Interval          0.096  0.142       0.094  0.140
  P-value RMSEA <= 0.05                          0.000       0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.109       0.109
```

Insgesamt zeigt sich eine sehr schlechte Passung des Modells.

```
Parameter estimates:

  Information                                 Observed
  Standard Errors                   Robust.huber.white

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  eta =~
    item_1            1.000                                0.252    0.650
    item_2            1.000                                0.252    0.650
    item_3            1.000                                0.252    0.650
    item_4            1.000                                0.252    0.650
    item_5            1.000                                0.252    0.650
    item_6            1.000                                0.252    0.650
```

Gemäß der Modellspezifikation sind alle Ladungen auf 1 restringiert.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Intercepts:
    item_1   (v1)     1.379    0.018   76.247    0.000    1.379    3.561
    item_2   (v1)     1.379    0.018   76.247    0.000    1.379    3.561
    item_3   (v1)     1.379    0.018   76.247    0.000    1.379    3.561
    item_4   (v1)     1.379    0.018   76.247    0.000    1.379    3.561
    item_5   (v1)     1.379    0.018   76.247    0.000    1.379    3.561
    item_6   (v1)     1.379    0.018   76.247    0.000    1.379    3.561
    eta               0.000                                0.000    0.000
```

Die geschätzten Itemleichtigkeiten sind aufgrund der Modellrestriktion alle identisch.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Variances:
    item_1   (v2)     0.087    0.004                       0.087    0.577
    item_2   (v2)     0.087    0.004                       0.087    0.577
    item_3   (v2)     0.087    0.004                       0.087    0.577
    item_4   (v2)     0.087    0.004                       0.087    0.577
    item_5   (v2)     0.087    0.004                       0.087    0.577
```

```
    item_6    (v2)      0.087    0.004                           0.087    0.577
    eta                 0.063    0.011                           1.000    1.000
```

Die geschätzten Fehlervarianzen der Items sind aufgrund der Modellrestriktion alle identisch.

4.8 Das Modell τ-kongenerischer Variablen

```
> m5<-'eta=~item_1+item_2+item_3+item_4+item_5+item_6'
```

Das Modell τ-kongenerischer Messung ist das allgemeinste der vorgestellten Messmodelle für kontinuierliche abhängige Variablen. Die Fehlervarianzen, Ladungen und Leichtigkeitsparameter werden frei geschätzt.

```
> fit5<-sem(m5, data=data.mat, meanstructure=TRUE)
> summary(fit5, fit.measures=TRUE, standardized=TRUE)
```

Das Modell wird an die Daten angepasst, und eine detaillierte Ausgabe wird angefordert.

```
  Minimum Function Test Statistic                9.568       8.894
  Degrees of freedom                                 9           9
  P-value (Chi-square)                           0.387       0.447
  Scaling correction factor                                  1.076
    for the Yuan-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              435.847     424.389
  Degrees of freedom                                15          15
  P-value                                        0.000       0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.999       1.000
  Tucker-Lewis Index (TLI)                       0.998       1.000

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)               -432.180    -432.180
  Scaling correction factor                                  1.092
    for the MLR correction
  Loglikelihood unrestricted model (H1)       -427.396    -427.396
  Scaling correction factor                                  1.087
    for the MLR correction

  Number of free parameters                         18          18
  Akaike (AIC)                                 900.360     900.360
  Bayesian (BIC)                               962.861     962.861
  Sample-size adjusted Bayesian (BIC)          905.806     905.806

Root Mean Square Error of Approximation:
```

```
  RMSEA                                         0.016       0.000
  90 Percent Confidence Interval         0.000  0.076       0.000  0.070
  P-value RMSEA <= 0.05                         0.763       0.815

Standardized Root Mean Square Residual:

  SRMR                                          0.021       0.021
```

Die Fit-Indizes zeigen insgesamt eine gute Passung des Modells.

```
Parameter estimates:

  Information                                 Observed
  Standard Errors                   Robust.huber.white

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  eta =~
    item_1            1.000                               0.229    0.638
    item_2            1.098    0.123    8.925    0.000    0.251    0.682
    item_3            1.194    0.146    8.163    0.000    0.273    0.697
    item_4            1.294    0.158    8.213    0.000    0.296    0.728
    item_5            1.032    0.146    7.053    0.000    0.236    0.628
    item_6            1.049    0.149    7.037    0.000    0.240    0.628
```

Die geschätzten Ladungen, die Standardfehler, entsprechende *z*-Tests und die standardisierten Koeffizienten werden ausgegeben.

```
Intercepts:
    item_1            1.504    0.023   64.757    0.000    1.504    4.198
    item_2            1.423    0.024   59.647    0.000    1.423    3.866
    item_3            1.392    0.025   54.862    0.000    1.392    3.556
    item_4            1.305    0.026   49.486    0.000    1.305    3.208
    item_5            1.346    0.024   55.221    0.000    1.346    3.579
    item_6            1.306    0.025   52.662    0.000    1.306    3.414
    eta               0.000                               0.000    0.000
```

Die geschätzten Leichtigkeitsparameter und der auf 0 fixierte Mittelwert der Verteilung der latenten Variablen werden ausgegeben.

```
Variances:
    item_1            0.076    0.008                      0.076    0.593
    item_2            0.072    0.008                      0.072    0.534
    item_3            0.079    0.009                      0.079    0.514
    item_4            0.078    0.010                      0.078    0.471
    item_5            0.086    0.010                      0.086    0.606
    item_6            0.089    0.010                      0.089    0.606
    eta               0.052    0.014                      1.000    1.000
```

Die geschätzten Varianzen der Items und die Varianzen der latenten Variablen werden ausgegeben.

4.9 Modellvergleiche mittels Likelihood-Quotienten-Tests

```
> fit1_ml<-sem(m1, data=data.mat, meanstructure=TRUE)
> fit2_ml<-sem(m2, data=data.mat, meanstructure=TRUE)
> anova(fit1_ml, fit2_ml)
```

Mit der Funktion `anova()` können Likelihood-Quotienten-Tests durchgeführt werden, die es erlauben zu prüfen, ob ein Modell A die Daten besser beschreibt als ein Modell B, sofern die Modelle ineinander verschachtelt sind bzw. durch Parameter-Restriktionen ineinander überführbar sind. Zudem müssen beide Modelle mit der Maximum-Likelihood-Methode angepasst worden sein. Für die MLR-Methoden muss auf die bei Eid und Schmidt (2014, Abschnitt 6.4.5.1.2) beschriebene Korrektur zurückgegriffen werden. Hier wird ein Vergleich des Modells essenziell τ-äquivalenter Variablen mit einem Modell τ-äquivalenter Variablen vorgenommen. Es interessiert, ob das Modell mit mehr Parametern, also das Modell essenziell τ-äquivalenter Variablen, die Daten besser beschreibt als das Modell τ-äquivalenter Variablen.

```
Chi Square Difference Test

        Df    AIC    BIC   Chisq Chisq diff
fit1_ml 14 897.74 942.88  16.949
fit2_ml 19 970.91 998.69 100.116     83.168
        Df diff Pr(>Chisq)
fit1_ml
fit2_ml       5  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In der Ausgabe werden die Freiheitsgrade, die informationstheoretischen Indizes AIC und BIC, die χ^2-Teststatistik des jeweiligen Modells, die Differenz der χ^2-Quadrat-Statistiken und die Parameterdifferenz sowie die Überschreitungswahrscheinlichkeit angegeben. Alle Angaben beziehen sich auf die ML-Methode.

Das Ergebnis des Tests legt nahe, dass das Modell der essenziell τ-äquivalenten Variablen die Daten besser beschreibt als das Modell der τ-äquivalenten Variablen. Dies zeigt auch der AIC-Koeffizient an.

Die informationstheoretischen Indizes sprechen allerdings eher für eine Präferenz des sparsameren Modells.

4.10 Schätzung der Personenwerte

```
> predict(fit1)
```

Es werden die Personenwerte (Faktorwerte) des Modells essenziell τ-äquivalenter Variablen (Objekt `fit1`) geschätzt.

```
       eta
  [1,]  0.132
  [2,]  0.261
  [3,]  0.004
  [4,] -0.006
  [5,] -0.245
  [6,] -0.205
  [7,]  0.441
```

[…]

5 Beispielsyntax für Kapitel 7: Einführung in mehrdimensionale Testmodelle

Das Paket `lavaan` kann zur Schätzung mehrdimensionaler Modelle verwendet werden, sofern das Messmodell mit der konfirmatorischen Faktorenanalyse kompatibel ist.

Mehrdimensionale Modelle mit ordinalen und kontinuierlichen Variablen und Mehrkomponentenmodelle, wie z.B. Latent-State-Trait-Modelle gehören zu dieser Klasse.

Die explorative Faktorenanalyse für ordinale Daten kann mit der Software Mplus durchgeführt werden. Aber auch die Funktion `fa()` im Paket `psych` eignet sich für die Analyse, sofern diese auf der polychorischen Korrelationsmatrix aufbaut.

5.1 Überblick Gesamtsyntax

```
# Syntax Kapitel 7
library(lavaan)
library(memisc)

## Zweifaktorielles Modell der emotionalen Klarheit
data1<-
as.data.set(spss.system.file('kapitel_7_beispiel_1_emotionale_klarhe
it.sav'))

# Umwandlung in den Typ data.frame
data1<-data.frame(data1)

m_emo<-'rakt=~reakt_1+1*reakt_2+1*reakt_3+1*reakt_4+
        1*reakt_5+1*reakt_6
        klar=~1*kla_th1+1*kla_th2
        reakt_1~~v1*reakt_1
        reakt_2~~v1*reakt_2
        reakt_3~~v1*reakt_3
        reakt_4~~v1*reakt_4
        reakt_5~~v1*reakt_5
        reakt_6~~v1*reakt_6
        kla_th1~~v2*kla_th1
        kla_th2~~v2*kla_th2
```

```
        '

fit_emo<-sem(m_emo, data=data1, meanstructure=TRUE, estimator="MLR")
summary(fit_emo, fit.measures=TRUE, standardized=TRUE)

## Multikomponenten-Modell
# Einlesen der Daten
data2<-read.table("kapitel_7_lst_beispiel.dat", col.names=c("Y1",
"Y2", "Y3","Y4"))

m_lst<-'eta1=~Y1+1*Y2+1*Y3+1*Y4
        eta2=~Y1 + 1*Y2
        eta3=~Y3 + 1*Y4
        eta1~~0*eta2
        eta1~~0*eta3
        eta2~~0*eta3
        Y1~~v*Y1
        Y2~~v*Y2
        Y3~~v*Y3
        Y4~~v*Y4'

fit_lst<-sem(m_lst, data=data2, meanstructure=TRUE, estimator="MLR")
summary(fit_lst, fit.measures=TRUE, standardized=TRUE)

## Exploratorische Faktorenanalyse (Alternative: Mplus)
# Einlesen der Daten
data3<-
as.data.set(spss.system.file('kapitel_7_beispiel_exploratorische_fak
torenanalyse.sav'))
data3<-data.frame(data3)

N=dim(data3)[1]

# Laden der benötigten Pakete
library(psych)
library(polycor)
library(GPArotation)

# Schätzung der polychorischen Korrelationsmatrix
poly_cor<-polychoric(data3)

# Ausgabe der Ergebnisse
# (Korrelations-Matrix und Schwellen tau)
print(poly_cor)

# Durchführung der Analyse mit Oblimin-Rotation
faPC <- fa(r=poly_cor$rho, nfactors=2,n.obs=N, rotate="oblimin")

# Darstellung der Ergebnisse
print(faPC)
```

5.2 Zweidimensionales Modell zur Analyse der emotionalen Klarheit

Das Laden von Daten wurde in vorherigen Abschnitten beschrieben, von daher wird hier nicht mehr darauf eingegangen.

```
m_emo<-'rakt=~reakt_1+1*reakt_2+1*reakt_3+1*reakt_4+
        1*reakt_5+1*reakt_6
        klar=~1*kla_th1+1*kla_th2
        reakt_1~~v1*reakt_1
        reakt_2~~v1*reakt_2
        reakt_3~~v1*reakt_3
        reakt_4~~v1*reakt_4
        reakt_5~~v1*reakt_5
        klar_th1~~v2*klar_th1
        klar_th2~~v2*klar_th2'
```

In dem spezifizierten Modell wird davon ausgegangen, dass die logarithmierten Reaktionszeiten und die Selbstratings jeweils auf einen Faktor laden, wobei jeweils essenzielle τ-Äquivalenz vorliegt. Die zwei latenten Variablen können miteinander kovariieren.

In der Syntax werden zwei Faktoren definiert. Die Items `reakt_1` bis `reakt_6` laden jeweils auf dem Faktor `reakt` und die Items `kla_th1` und `kla_th2` laden jeweils auf dem Faktor `klar`. Per Standardeinstellung wird die Kovarianz zwischen den latenten Variablen geschätzt.

Weiterhin wurden die Fehlervarianzen der konstruktspezifischen Items jeweils gleichgesetzt, sodass zwei Modelle essenziell τ-paralleler Variablen vorliegen.

```
> fit_emo<-sem(m_emo, data=data1, meanstructure=TRUE,
estimator="MLR")
> summary(fit_emo, fit.measures=TRUE, standardized=TRUE)

lavaan (0.5-14) converged normally after  28 iterations

  Number of observations                           238

  Estimator                                         ML      Robust
  Minimum Function Test Statistic               23.560      23.223
  Degrees of freedom                                31          31
  P-value (Chi-square)                           0.828       0.841
  Scaling correction factor                                  1.015
    for the Yuan-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              666.758     649.897
  Degrees of freedom                                28          28
  P-value                                        0.000       0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    1.000       1.000
  Tucker-Lewis Index (TLI)                       1.011       1.011

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -1283.975   -1283.975
  Scaling correction factor                                  1.160
    for the MLR correction
  Loglikelihood unrestricted model (H1)      -1272.195   -1272.195
```

```
  Scaling correction factor                                 1.057
    for the MLR correction

  Number of free parameters                       13           13
  Akaike (AIC)                              2593.950     2593.950
  Bayesian (BIC)                            2639.090     2639.090
  Sample-size adjusted Bayesian (BIC)       2597.884     2597.884

Root Mean Square Error of Approximation:

  RMSEA                                        0.000        0.000
  90 Percent Confidence Interval        0.000  0.030        0.000  0.028
  P-value RMSEA <= 0.05                        0.996        0.997

Standardized Root Mean Square Residual:

  SRMR                                         0.049        0.049
```

Insgesamt deuten die Fit-Statistiken auf eine sehr gute Passung des Modells hin.

```
Parameter estimates:

  Information                                 Observed
  Standard Errors                   Robust.huber.white

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  rakt =~
    reakt_1           1.000                               0.254    0.667
    reakt_2           1.000                               0.254    0.667
    reakt_3           1.000                               0.254    0.667
    reakt_4           1.000                               0.254    0.667
    reakt_5           1.000                               0.254    0.667
    reakt_6           1.000                               0.254    0.667
  klar =~
    kla_th1           1.000                               1.611    0.886
    kla_th2           1.000                               1.611    0.886
```

Gemäß der Modelldefinition sind alle Ladungen auf 1 fixiert.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all

Covariances:
  rakt ~~
    klar             -0.009    0.031   -0.276    0.782   -0.021   -0.021
```

Die geschätzte Korrelation (`Std.all`) der latenten Variablen beträgt -0.021 und ist nicht signifikant von 0 verschieden.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Intercepts:
    reakt_1           1.504    0.023   64.757    0.000    1.504    3.949
    reakt_2           1.423    0.024   59.647    0.000    1.423    3.736
    reakt_3           1.392    0.025   54.862    0.000    1.392    3.655
    reakt_4           1.305    0.026   49.486    0.000    1.305    3.426
    reakt_5           1.346    0.024   55.221    0.000    1.346    3.535
```

```
    reakt_6           1.306    0.025   52.662    0.000    1.306    3.428
    kla_th1           9.189    0.117   78.282    0.000    9.189    5.051
    kla_th2           9.206    0.118   77.711    0.000    9.206    5.060
    rakt              0.000                               0.000    0.000
    klar              0.000                               0.000    0.000
```

Die geschätzten Leichtigkeitsparameter der Items und die Mittelwerte der latenten Variablen werden ausgegeben.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Variances:
    reakt_1  (v1)     0.081    0.004                      0.081    0.556
    reakt_2  (v1)     0.081    0.004                      0.081    0.556
    reakt_3  (v1)     0.081    0.004                      0.081    0.556
    reakt_4  (v1)     0.081    0.004                      0.081    0.556
    reakt_5  (v1)     0.081    0.004                      0.081    0.556
    reakt_6  (v1)     0.081    0.004                      0.081    0.556
    kla_th1  (v2)     0.714    0.080                      0.714    0.216
    kla_th2  (v2)     0.714    0.080                      0.714    0.216
    rakt              0.064    0.011                      1.000    1.000
    klar              2.596    0.268                      1.000    1.000
```

Die geschätzten Fehlervarianzen der Items und die geschätzten Varianzen der latenten Variablen werden ausgegeben.

5.3 Multikomponentenmodell

```
m_lst<-'eta1=~Y1+1*Y2+1*Y3+1*Y4
        eta2=~Y1 + 1*Y2
        eta3=~Y3 + 1*Y4
        eta1~~0*eta2
        eta1~~0*eta3
        eta2~~0*eta3
        Y1~~v*Y1
        Y2~~v*Y2
        Y3~~v*Y3
        Y4~~v*Y4'
```

In diesem Multikomponentenmodell, einem Latent-State-Trait-Modell, laden alle Items mit einer Ladung von 1 auf der latenten Variablen η_1, die den Trait repräsentiert. Zudem werden die messgelegenheitsspezifischen latenten Variablen η_2 und η_3 definiert, auf welche die zu dem jeweiligen Messzeitpunkt erhobenen Items ebenfalls mit einer Ladung von 1 laden. Korrelationen zwischen den latenten Variablen werden nicht zugelassen und alle Fehlervarianzen der Items werden über das Label `v` gleich gesetzt. Damit wird eine essenziell τ-parallele Messstruktur impliziert.

```
> fit_lst<-sem(m_lst, data=data2, meanstructure=TRUE,
estimator="MLR")
> summary(fit_lst, fit.measures=TRUE, standardized=TRUE)
```

Das Modell wird angepasst und ein detaillierter Output wird angefordert.

```
lavaan (0.5-14) converged normally after  16 iterations

  Number of observations                           212

  Estimator                                         ML      Robust
  Minimum Function Test Statistic                6.340       4.089
  Degrees of freedom                                 6           6
  P-value (Chi-square)                           0.386       0.665
  Scaling correction factor                                  1.550
    for the Yuan-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              362.843     193.124
  Degrees of freedom                                 6           6
  P-value                                        0.000       0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.999       1.000
  Tucker-Lewis Index (TLI)                       0.999       1.010

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)               -922.363    -922.363
  Scaling correction factor                                  1.381
    for the MLR correction
  Loglikelihood unrestricted model (H1)       -919.193    -919.193
  Scaling correction factor                                  1.453
    for the MLR correction

  Number of free parameters                          8           8
  Akaike (AIC)                                1860.725    1860.725
  Bayesian (BIC)                              1887.578    1887.578
  Sample-size adjusted Bayesian (BIC)         1862.229    1862.229

Root Mean Square Error of Approximation:

  RMSEA                                          0.016       0.000
  90 Percent Confidence Interval          0.000  0.092       0.000
0.056
  P-value RMSEA <= 0.05                          0.676       0.928

Standardized Root Mean Square Residual:

  SRMR                                           0.041       0.041
```

Insgesamt deuten die Fit-Statistiken auf eine gute Modellpassung hin.

```
Parameter estimates:

  Information                                    Observed
```

```
  Standard Errors                    Robust.huber.white

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  eta1 =~
    Y1                1.000                               0.641    0.719
    Y2                1.000                               0.641    0.719
    Y3                1.000                               0.641    0.726
    Y4                1.000                               0.641    0.726
  eta2 =~
    Y1                1.000                               0.354    0.397
    Y2                1.000                               0.354    0.397
  eta3 =~
    Y3                1.000                               0.331    0.375
    Y4                1.000                               0.331    0.375
```

Gemäß der Modelldefinition sind alle Ladungen auf 1 fixiert.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Covariances:
  eta1 ~~
    eta2              0.000                               0.000    0.000
    eta3              0.000                               0.000    0.000
  eta2 ~~
    eta3              0.000                               0.000    0.000
```

Die Kovarianzen zwischen den latenten Variablen sind alle auf 0 fixiert.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Intercepts:
    Y1                3.880    0.064   60.523    0.000    3.880    4.352
    Y2                4.059    0.058   69.451    0.000    4.059    4.553
    Y3                3.960    0.060   65.874    0.000    3.960    4.487
    Y4                4.066    0.061   66.789    0.000    4.066    4.608
    eta1              0.000                               0.000    0.000
    eta2              0.000                               0.000    0.000
    eta3              0.000                               0.000    0.000
```

Die geschätzten Itemleichtigkeiten und die Mittelwerte der latenten Variablen werden ausgegeben.

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Variances:
    Y1         (v)    0.259    0.031                      0.259    0.326
    Y2         (v)    0.259    0.031                      0.259    0.326
    Y3         (v)    0.259    0.031                      0.259    0.332
    Y4         (v)    0.259    0.031                      0.259    0.332
    eta1              0.410    0.067                      1.000    1.000
    eta2              0.125    0.044                      1.000    1.000
    eta3              0.109    0.046                      1.000    1.000
```

Die geschätzten Fehlervarianzen der Items und die geschätzten Varianzen der latenten Variablen werden ausgegeben.

5.4 Exploratorische Faktorenanalyse für kategoriale Daten

Die exploratorische Faktorenanalyse für kategoriale Daten ist relativ unkompliziert mit der Analyse-Software Mplus (Muthén & Muthén, 2012) durchführbar. Mplus stellt auch Konfidenzintervalle und Modellgütekoeffizienten zur Verfügung. In `R` ist es möglich, mit der Funktion `fa()` im Paket `psych` eine exploratorische Faktorenanalyse für ordinale Daten auf Basis der polychorischen Korrelationen durchzuführen.

```
data3<-
as.data.set(spss.system.file('kapitel_7_beispiel_exploratorische_fak
torenanalyse.sav'))
data3<-data.frame(data3)
```

Zunächst werden die Daten geladen und in das Format `data.frame` konvertiert.

```
> N=dim(data3)[1]
```

Die Anzahl der Personen im Datensatz wird ermittelt.

```
> library(psych)
> library(polycor)
> library(GPArotation)
```

Die benötigten Pakete werden geladen. Im Paket `psych` ist die exploratorische Faktorenanalyse implementiert. Das Paket `polycor` wird für die Schätzung der polychorischen Korrelationsmatrix benötigt. Im Paket `GPArotation` sind verschiedene Rotationsmethoden implementiert, u.a. die Oblimin-Rotation.

```
> poly_cor<-polychoric(data3)
```

Die polychorische Korrelationsmatrix der Items im Daten-Frame `data3` wird geschätzt.

```
> print(poly_cor)
Call: polychoric(x = data3)
Polychoric correlations
       itm_1 itm_2 itm_3 itm_4 itm_5 itm_6
item_1 1.00
item_2 0.71  1.00
item_3 0.43  0.50  1.00
item_4 0.70  0.78  0.50  1.00
item_5 0.31  0.42  0.59  0.45  1.00
item_6 0.51  0.57  0.65  0.58  0.70  1.00

 with tau of
          1    2      3    4
item_1 -2.6 -1.4 -0.356 1.03
```

```
item_2 -2.6 -1.7 -0.683 0.99
item_3 -2.3 -1.2 -0.258 1.05
item_4 -2.5 -1.7 -0.599 0.90
item_5 -2.5 -1.2 -0.013 1.29
item_6 -2.2 -1.1  0.039 1.55
```

Die geschätzte polychorische Korrelationsmatrix und die geschätzten Schwellenparameter werden ausgegeben. Insgesamt zeigen sich relativ hohe Interkorrelationen zwischen den Items.

```
> faPC <- fa(r=poly_cor$rho, nfactors=2, n.obs=N, rotate="oblimin")
```

Die exploratorische Faktorenanalyse wird auf Basis der polychorischen Korrelationsmatrix `poly_cor$roh` durchgeführt. Es werden zwei Faktoren extrahiert. Die Anzahl der Beobachtungen `n.obs` entspricht N=482, die Rotationsmethode wird mit `rotate="oblimin"` eingestellt.

```
> print(faPC)
```

Das Ergebnis der Analyse wird ausgegeben.

```
Factor Analysis using method =  minres
Call: fa(r = poly_cor$rho, nfactors = 2, n.obs = N, rotate =
"oblimin")
Standardized loadings (pattern matrix) based upon correlation matrix
         MR1   MR2   h2   u2 com
item_1  0.84 -0.06 0.65 0.35 1.0
item_2  0.87  0.02 0.79 0.21 1.0
item_3  0.14  0.64 0.55 0.45 1.1
item_4  0.82  0.08 0.76 0.24 1.0
item_5 -0.11  0.90 0.70 0.30 1.0
item_6  0.16  0.76 0.75 0.25 1.1
```

Zur Schätzung kam standardmäßig die Methode Residuenminimierung (`minres`) zum Einsatz. Es werden die standardisierten Ladungen der Items auf den Faktoren (`MR1, MR2`), die Kommunalitäten (`h2`) und Fehlervarianzen ausgegeben (`u2`).

```
                       MR1  MR2
SS loadings           2.30 1.89
Proportion Var        0.38 0.32
Cumulative Var        0.38 0.70
Proportion Explained  0.55 0.45
Cumulative Proportion 0.55 1.00
```

Es folgen Statistiken zur Varianzaufklärung durch die Faktoren. Die Zeile `SS loadings` bezeichnet die Summe der quadrierten Faktorladungen, die Zeile `Proportion Var` bezeichnet die Varianzaufklärung der jeweiligen Faktoren und `Proportion Explained` gibt an, wie viel

Prozent der durch die zwei Faktoren aufgeklärten Varianz auf den jeweiligen Faktor zurückzuführen sind. Die entsprechenden kumulierten Statistiken werden ebenfalls ausgegeben.

```
 With factor correlations of 
     MR1  MR2
MR1 1.00 0.63
MR2 0.63 1.00
```

Die Interkorrelationsmatrix der extrahierten Faktoren wird ausgegeben. Die geschätzte Korrelation der Faktoren beträgt r=0.63.

7 Literatur

Davier, M. von (2001). *WINMIRA 2001* [Computer-Programm]. Verfügbar unter www.von-davier.com (27.09.2013).

Eid, M. & Schmidt, K. (2014). *Testtheorie und Testkonstruktion*. Göttingen: Hogrefe.

Fischer, G. H. (1995). The linear logistic test model. In G.H. Fischer & I.W. Molenaar (Eds.), *Rasch models – foundations, recent developments and applications*. Berlin: Springer.

Frick, H., Strobl, C., Leisch, F. & Zeileis, A. (2012). Flexible Rasch mixture models with package psychomix. *Journal of Statistical Software, 48* (7), 1–25.

Koller, I., Alexandrowicz, R. & Hatzinger, R. (2012). *Das Rasch-Modell in der Praxis. Eine Einführung in eRm*. Wien: Facultas.

Luhmann, M. (2013). *R für Einsteiger. Einführung in die Statistiksoftware für die Sozialwissenschaften*. Weinheim: Beltz.

Mair, P. & Hatzinger, H. (2007). Extended Rasch modeling: the eRm package for the application of Rasch models in R. *Journal of Statistical Software, 20* (9).

Muthén, L.K. & Muthén, B.O. (2012). *Mplus user's guide* (7th ed.). Los Angeles, CA: Muthén & Muthén.

Revelle, W. (2013). *Package 'psych': Procedures for psychological, psychometric, and personality research.* Verfügbar unter http://cran.r-project.org/web/packages/psych/psych.pdf (24.02.2014).

Rizopulous, D. (2006). ltm: an R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, *17* (5).

Rosseel, Y. (2012). lavaan: an R package for structural equation modeling. *Journal of Statistical Software*, *48* (5).