R-Syntax unter Verwendung von lavaan und MBESS zu Kapitel 14: Klassische Methoden der Reliabilitätsschätzung

Jana Gäde & Karin Schermelleh-Engel

01.03.2021

1 Inhalt

1 Vorbemerkungen

Im Folgenden wird die Schätzung der klassischen Reliabilitätskoeffizienten Cronbachs Alpha, Split-Half-Reliabilität und Retest-Reliabilität anhand des Statistikprogramms R (R Development Core Team, 2021) unter Verwendung der Packages MBESS (Kelley, 2007a, 2007b, 2020) und lavaan (Rosseel, 2012) dargestellt. Für die konfirmatorischen Faktorenanalysen (CFA), die zur Testung der Voraussetzungen benötigt werden, wird das Package lavaan (Rosseel, 2012) verwendet, für die Schätzung der Reliabilitätsmaße das Package MBESS. Eine alternative Vorgehensweise würde darin bestehen, die verschiedenen Reliabilitätskoeffizienten mit präzisen, asymmetrischen Konfidenzintervallen formelbasiert selbst in R zu programmieren. Beispiele dazu finden sich ebenfalls im Zusatzmaterial (Schermelleh-Engel, Gäde & Irmer, 2021a, 2021b).

1.1 Beispieldaten

Zur Demonstration der Schätzung der Reliabilitätskoeffizienten Cronbachs Alpha, Split-Half-Reliabilität und Retest-Reliabilität wurden zunächst mit dem Programm M*plus* (Muthén & Muthén, 2017) künstliche Daten (*N* = 200) generiert, die für die Analysen verwendet wurden. Die Variablen in den Datensätzen stellen Itemvariablen dar, die jeweils zu einer Testwertvariablen aufsummiert werden sollen. Es soll gezeigt werden, wie die Reliabilität einer solchen Testwertvariablen anhand von Cronbachs Alpha, der Split-Half-Reliabilität und der Retest-Reliabilität geschätzt werden kann. Drei Datensätze mit jeweils 8 Variablen werden für die Reliabilitätsschätzungen verwendet:

- Beispiel 1: Zur Schätzung der Reliabilität anhand von Cronbachs Alpha werden 8 Itemvariablen verwendet (Alpha_Labels.dat).
- Beispiel 2: Dieselben 8 Itemvariablen wie in Beispiel 1 sollen auch hier werden verwendet. Diesmal soll die Reliabilität anhand der Korrelation zweier Halb-Testwertvariablen geschätzt und mittels Spearman-Brown-Korrektur zur Reliabilität der Testwertvariablen des Gesamttests aufgewertet werden (Split-Half-Reliabilität). Dazu werden die Itemvariablen in 2 Teilskalen mit jeweils 4 Items aufgeteilt (Split_Half_Labels.dat).
- Beispiel 3: Die Reliabilität soll anhand der Retest-Reliabilität geschätzt werden. Dazu wird die Korrelation zwischen den Testwerten aus einer ersten und einer zweiten Messung ermittelt. Die Testwerte ergeben sich aus 4 Itemvariablen, die zu 2 Messzeitpunkten wiederholt gemessen wurden. Der Datensatz umfasst durch die Wiederholungsmessung ebenfalls 8 Variablen (Retest_Labels.dat).

1.2 Überprüfung der Voraussetzungen

Die Voraussetzungen der klassischen Reliabilitätskoeffizienten können anhand eines Modelltests im Rahmen der konfirmatorischen Faktorenanalyse (CFA, vgl. Gäde, Schermelleh-Engel & Brandt, 2020) bspw. unter Verwendung des R-Packages lavaan (Rosseel, 2012) überprüft werden. Dazu werden Faktormodelle mit den entsprechenden Restriktionen spezifiziert. Ein guter Modellfit würde bedeuten, dass die Voraussetzungen erfüllt sind.

! Die klassischen Reliabilitätskoeffizienten dürfen nur dann als Reliabilität einer Testwertvariablen interpretiert werden, wenn die Voraussetzungen des jeweiligen Reliabilitätskoeffizienten erfüllt sind.

Zeigt sich ein schlechter Modellfit, der darauf verweist, dass die Voraussetzungen nicht erfüllt sind, sollte ein anderes Reliabilitätsmaß verwendet werden, das auf weniger strengen Voraussetzungen beruht. Da die Überprüfung der Voraussetzungen mittels CFA erfolgt, kann auf dem zugrundeliegenden Faktormodell aufbauend statt der klassischen Methode eine modellbasierte Methode der Reliabilitätsschätzung verwendet werden (Schermelleh-Engel & Gäde, 2020).

1.3 Schätzung der Koeffizienten

Die klassischen Methoden der Reliabilitätsschätzung sind in vielen Statistikprogrammen standardmäßig implementiert und daher leicht verfügbar. Trotz der leichten Verfügbarkeit sollte jedoch nicht vergessen werden, dass die klassischen Reliabilitätskoeffizienten auf strengen Voraussetzungen beruhen und nur dann sinnvoll als Reliabilität einer Testwertvariablen interpretiert werden können, wenn die Voraussetzungen erfüllt sind (vgl. Gäde, Schermelleh-Engel & Werner, 2020). Die Voraussetzungen sollten daher immer überprüft werden.

Neben der Punktschätzung des Reliabilitätskoeffizienten ist eine Intervallschätzung in Form eines zweiseitigen 95%-Konfidenzintervalls nötig, um eine Aussage über die Präzision der Schätzung zu erhalten (Kelley & Pornprasertmanit 2016; Raykov 2002). Da der Wertebereich der Reliabilitätskoeffizienten auf die Grenzen null und eins beschränkt ist und somit das Konfidenzintervall – vor allem in der Nähe der Grenzwerte – nicht symmetrisch sein kann, sollte ein asymmetrisches Konfidenzintervall (Raykov, 2002) bestimmt werden.

Im Folgenden werden Möglichkeiten zur Schätzung eines Konfidenzintervalls mit dem Package MBESS (Kelley, 2007a, 2007b, 2020) gezeigt. Die hier vorgeschlagenen Methoden sind natürlich nicht die einzigen, die verwendet werden können, sondern stellen nur eine mögliche Alternative dar.

! Asymmetrische Konfidenzintervalle können bspw. in R (oder auch in dem Programm M*plus*) direkt programmiert und geschätzt werden (Schermelleh-Engel, Gäde & Irmer, 2021a, 2021b).

1.4 Literatur

Gäde, J.C., Schermelleh-Engel, K. & Brandt, H. (2020). Konfirmatorische Faktorenanalyse (CFA; Kapitel 24). In H. Moosbrugger & A. Kelava (Hrsg.), *Testtheorie und Fragebogenkonstruktion* (3., vollständig neu bearbeitete, erweiterte und aktualisierte Auflage). Heidelberg: Springer.

Gäde, J.C., Schermelleh-Engel, K. & Werner, C.S. (2020). Klassische Methoden der Reliabilitätsschätzung (Kapitel 14). In H. Moosbrugger & A. Kelava (Hrsg.), *Testtheorie und Fragebogenkonstruktion* (3., vollständig neu bearbeitete, erweiterte und aktualisierte Auflage). Heidelberg: Springer.

Kelley, K. (2007a). Methods for the Behavioral, Educational, and Social Sciences: An R Package. *Behavior Research Methods, 39,* 979–984.

Kelley, K. (2007b). Confidence intervals for standardized effect sizes: Theory, application, and implementation. *Journal of Statistical Software, 20,* 1-24.

Kelley, K. (2020). MBESS (Version 4.8.0) [Computer software and manual]. Retrievable from www.cran.r-project.org/.

Kelley, K. & Pornprasertmanit, S. (2016). Confidence intervals for population reliability coefficients: Evaluation of methods, recommendations, and software for homogeneous composite measures. *Psychological Methods, 21,* 69–92.

Muthén, L. K. & Muthén, B. O. (2017). *Mplus User's Guide* (8th ed.). Los Angeles, CA: Muthén & Muthén.

Raykov, T. (2002) Analytic estimation of standard error and confidence interval for scale reliability. *Multivariate Behavioral Research, 37*, 89–103.

R Development Core Team (2021). *R: A Language and Environment for Statistical Computing* (version 4.0.4). Vienna, Austria: R Foundation for Statistical Computing.

Zu Kap. 14 – R-Syntax unter Verwendung von lavaan und MBESS

Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software, 48*, 1–36.

Schermelleh-Engel, K. & Gäde, J.C. (2020). Modellbasierte Methoden der Reliabilitätsschätzung (Kapitel 15). In H. Moosbrugger & A. Kelava (Hrsg.), *Testtheorie und Fragebogenkonstruktion* (3., vollständig neu bearbeitete, erweiterte und aktualisierte Auflage). Heidelberg: Springer.

Schermelleh-Engel, K., Gäde, J.C. & Irmer, J.P. (2021a). Mplus-Syntax - Direkte Programmierung zu Kapitel 14: Klassische Methoden der Reliabilitätsschätzung. Zusatzmaterialien zu H. Moosbrugger & A. Kelava (Hrsg.), Testtheorie und Fragebogenkonstruktion (3., vollständig neu bearbeitete, erweiterte und aktualisierte Auflage). Heidelberg: Springer. http://www.lehrbuch-psychologie.springer.com.

Schermelleh-Engel, K., Gäde, J.C. & Irmer, J.P. (2021b). R-Syntax - Direkte Programmierung zu Kapitel 14: Klassische Methoden der Reliabilitätsschätzung. Zusatzmaterialien zu H. Moosbrugger & A. Kelava (Hrsg.), Testtheorie und Fragebogenkonstruktion (3., vollständig neu bearbeitete, erweiterte und aktualisierte Auflage). Heidelberg: Springer. http://www.lehrbuch-psychologie.springer.com.

2 Cronbachs Alpha

2.1 Überprüfung der Voraussetzungen von Cronbachs Alpha

Cronbachs Alpha darf nur dann als Reliabilität einer Testwertvariablen interpretiert werden, wenn strenge Voraussetzungen erfüllt sind. Die Voraussetzungen von Cronbachs Alpha sind die Eindimensionalität der Messungen sowie die essentielle τ-Äquivalenz der Itemvariablen, d.h., alle Faktorladungen müssen identisch sein, während die Fehlervarianzen unterschiedlich sein dürfen. Diese Voraussetzungen können mittels CFA überprüft werden (▶ Box 2.a).

Box 2.a) R-Input zur Überprüfung der Voraussetzungen von Cronbachs Alpha

```
# Vor der ersten Anwendung: R-Package lavaan zur Analyse von latenten
# Variablenmodellen installieren, dazu einen nahegelegenen CRAN Mirror angeben
# Eine Liste der CRAN Mirrors: https://cran.r-project.org/mirrors.html
install.packages("lavaan", repos='https://cran.uni-muenster.de/')

# Vor jeder Anwendung muss das installierte Package lavaan zur Nutzung
# geladen werden
library(lavaan)

# Arbeitsverzeichnis auswählen ("Dateipfad/zum/Arbeitsverzeichnis")
setwd("C:/Daten/Kap14")

# Daten einlesen
# Trennzeichen: Tabstopps (sep="")
# Variablennamen stehen in erster Zeile (header=TRUE)
Alpha_Daten <- read.table("Alpha_labels.dat", sep="", header=TRUE)

# Definition eines eindimensionalen Modells mod mit dem Faktor 'ETA'
# Alle Faktorladungen erhalten dasselbe Label p1 und werden damit
# gleichgesetzt
# Bedingung der essentiellen Tau-Äquivalenz
mod <- 'ETA=~ p1*Y1 + p1*Y2 + p1*Y3 + p1*Y4 + p1*Y5 + p1*Y6 + p1*Y7 + p1*Y8

# Den Fehlervarianzen werden unterschiedliche Labels (e1 bis e8) zugewiesen
# und können sich somit unterscheiden
Y1~~e1*Y1
Y2~~e2*Y2
Y3~~e3*Y3
Y4~~e4*Y4
Y5~~e5*Y5
Y6~~e6*Y6
Y7~~e7*Y7
Y8~~e8*Y8
'
# Analyse des Modells mod
# ML-Schätzung (estimator= "ML")
# Die latente Varianz wurde auf eins fixiert (std.lv=TRUE)
fit.mod <- sem(mod, data= Alpha_Daten, estimator= "ML", std.lv=TRUE)

# Ergebnisse anzeigen
summary(fit.mod)
```

Zur Überprüfung der Voraussetzungen wurden alle Faktorladungen gleichgesetzt und die Fehlervarianzen frei geschätzt (▶ Box 2.a), wie auch in den Ergebnissen zu sehen ist (▶ Box 2.b). Das eindimensionale Modell mit den Restriktionen der essentiellen τ-Äquivalenz ergab für das gegebene Beispiel einen hinreichend guten Modellfit ($\chi^2(27) = 35.892$; $p = .118$, ▶ Box 2.b).
Somit kann davon ausgegangen werden, dass die Voraussetzungen der Eindimensionalität und der essentiellen τ-Äquivalenz erfüllt sind. Cronbachs Alpha darf daher geschätzt werden (▶ Absch. 2.2).

Box 2.b) R-Output zur Überprüfung der Voraussetzungen von Cronbachs Alpha

```
lavaan 0.6-7 ended normally after 11 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         16
  Number of equality constraints                     7

  Number of observations                           200

  Model Test User Model:

  Test statistic                                35.892
  Degrees of freedom                                27
  P-value (Chi-square)                           0.118
# Der Modellfit ist hinreichend gut

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  ETA =~
    Y1        (p1)    1.133    0.060   18.837    0.000
    Y2        (p1)    1.133    0.060   18.837    0.000
    Y3        (p1)    1.133    0.060   18.837    0.000
    Y4        (p1)    1.133    0.060   18.837    0.000
    Y5        (p1)    1.133    0.060   18.837    0.000
    Y6        (p1)    1.133    0.060   18.837    0.000
    Y7        (p1)    1.133    0.060   18.837    0.000
    Y8        (p1)    1.133    0.060   18.837    0.000
# Die Faktorladungen der Itemvariablen wurden gleichgesetzt

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .Y1        (e1)    0.829    0.091    9.097    0.000
   .Y2        (e2)    0.566    0.065    8.675    0.000
   .Y3        (e3)    0.572    0.066    8.688    0.000
   .Y4        (e4)    0.469    0.056    8.399    0.000
   .Y5        (e5)    0.778    0.086    9.037    0.000
   .Y6        (e6)    0.668    0.075    8.877    0.000
   .Y7        (e7)    0.671    0.076    8.883    0.000
   .Y8        (e8)    0.650    0.073    8.846    0.000
    ETA               1.000
# Die Fehlervarianzen der Itemvariablen wurden frei geschätzt

# Der Modellfit ist gut, somit sind die Voraussetzungen der Eindimensionalität
# und der essentiellen Tau-Äquivalenz erfüllt.
# Cronbachs Alpha kann nun anhand der klassischen Methode geschätzt werden.
```

2.2 Schätzung von Cronbachs Alpha

Zur Schätzung von Cronbachs Alpha kann beispielsweise das R-Package MBESS (Kelley, 2007a, b, 2017) genutzt werden, das unter anderem die Möglichkeit bietet, Punktschätzungen und Konfidenzintervalle verschiedener Reliabilitätskoeffizienten zu erhalten. Die Punkt- und Intervallschätzungen von Alpha sind auf diesem Weg leicht verfügbar.

Box 2.c) R-Input zur Schätzung von Cronbachs Alpha mit Konfidenzintervall

```
# Vor der ersten Anwendung: R-Package MBESS installieren, dazu einen
# nahegelegenen CRAN Mirror angeben
# Eine Liste der CRAN Mirrors: https://cran.r-project.org/mirrors.html
install.packages("MBESS", repos='https://cran.uni-muenster.de/')

# Vor jeder Anwendung muss das installierte Package MBESS zur Nutzung
# geladen werden
library(MBESS)

# Arbeitsverzeichnis auswählen ("Dateipfad/zum/Arbeitsverzeichnis")
setwd("C:/Daten/Kap14")

# Daten einlesen
# Trennzeichen: Tabstopps (sep="")
# Variablennamen stehen in erster Zeile (header=TRUE)
Alpha_Daten <- read.table("Alpha_labels.dat", sep="", header=TRUE)

# Über folgende Funktion kann die Reliabilität für eine Testwertvariable
# basierend auf mehreren Itemvariablen sowie das zugehörige Konfidenzintervall
# angefordert werden.
ci.reliability(data = Alpha_Daten, type = "alpha", interval.type = "bca",
                        conf.level = 0.95)

# „type“ : Art des Reliabilitätskoeffizienten
# „interval.type“ : Methoden zur Schätzung des Konfidenzintervalls
# „bca“ ist eine Bootstrap-Methode zur Schätzung des Konfidenzintervalls (bias
# -corrected and accelerated approach)
```

Für das Beispiel wurde wie von Kelley und Pornprasertmanit (2016) empfohlen eine Bootstrap-Methode zur Schätzung des Konfidenzintervalls verwendet (bias-corrected and accelerated approach, bca).
Für das Beispiel ergibt sich ein Cronbachs Alpha in Höhe von .942 (▶ Box 2.c) mit einem 95%-Konfidenzintervall von [0.929; 0.952] (▶ Box 2.d).

Box 2.d) R-Output zur Schätzung von Cronbachs Alpha mit Konfidenzintervall

```
$est                                    # Geschätzte Reliabilität
[1] 0.9417765

$se                                     # Geschätzter Standardfehler
[1] 0.006001823

$ci.lower                               # Untere Grenze des Konfidenzintervalls
[1] 0.9287486

$ci.upper                               # Obere Grenze des Konfidenzintervalls
[1] 0.9519076
$conf.level                             # 95%-Konfidenzintervall
[1] 0.95

$type                                   # Reliabilitätskoeffizient: Cronbachs Alpha
[1] "alpha"

$interval.type                          # Konfidenzintervallschätzung
[1] "bca bootstrap"
```

3 Split-Half-Reliabilität

3.1 Überprüfung der Voraussetzungen der Split-Half-Reliabilität

Die Korrelation zweier Halb-Testwertvariablen bzw. die mittels Spearman-Brown-Formel geschätzte Split-Half-Reliabilität des Gesamttests darf nur dann als Reliabilität einer Testwertvariablen interpretiert werden, wenn strenge Voraussetzungen erfüllt sind. Die Voraussetzungen der Split-Half-Reliabilität sind die Eindimensionalität der Messungen sowie die essentielle τ-Parallelität korrespondierender Itemvariablen über die beiden Testhälften hinweg. Diese Voraussetzungen können mittels CFA überprüft werden (▶ Box 3.a).

Box 3.a) R-Input zur Überprüfung der Voraussetzungen der Split-Half-Reliabilität

```
# Vor der ersten Anwendung: R-Package lavaan zur Analyse von latenten
# Variablenmodellen installieren, dazu einen nahegelegenen CRAN Mirror angeben
# Eine Liste der CRAN Mirrors: https://cran.r-project.org/mirrors.html
install.packages("lavaan", repos='https://cran.uni-muenster.de/')

# Vor jeder Anwendung muss das installierte Package lavaan zur Nutzung
# geladen werden
library(lavaan)

# Arbeitsverzeichnis auswählen ("Dateipfad/zum/Arbeitsverzeichnis")
setwd("C:/Daten/Kap14")

# Daten einlesen:
# Trennzeichen: Tabstopps (sep="")
# Variablennamen stehen in erster Zeile (header=TRUE)
SplitHalf_Daten <- read.table("Split_Half_labels.dat", sep="", header=TRUE)

# Definition des einfaktoriellen CFA-Modells mod1 mit dem Faktor 'ETA'
# Die Faktorladungen korrespondierender Itemvariablen der Testhaelften
# erhalten dasselbe Label und werden damit paarweise gleichgesetzt
# (Bedingung der essent. Tau-Parallelitaet)
mod1 <- 'ETA1 =~ p1*Y1 + p2*Y2 + p3*Y3 + p4*Y4 +
                 p1*Y5 + p2*Y6 + p3*Y7 + p4*Y8

# Den Fehlervarianzen korrespondierender Itemvariablen werden die Labels e1
# bis e4 zugewiesen. Damit sind die Fehlervarianzen ueber die Testhaelften
# hinweg paarweise gleichgesetzt (Bedingung der essent. Tau-Parallelitaet)
# Itemvariablen der 1. Testhaelfte
Y1~~e1*Y1
Y2~~e2*Y2
Y3~~e3*Y3
Y4~~e4*Y4
# Itemvariablen der 2. Testhaelfte
Y5~~e1*Y5
Y6~~e2*Y6
Y7~~e3*Y7
Y8~~e4*Y8
'
# Analyse des Modells mod1
# ML-Schätzung (estimator= "ML")
# die latente Varianz ist auf eins fixiert (std.lv=TRUE)
fit.mod1 <- sem(mod1, data= SplitHalf_Daten, estimator= "ML", std.lv=TRUE)

# Ergebnisse anzeigen
summary(fit.mod1)
```

Im unserem Beispiel gehören die Itemvariablen Y1-Y4 zur ersten Testhälfte (Half1) und die Itemvariablen Y5-Y8 zur zweiten Testhälfte (Half2). Zur Überprüfung der Voraussetzungen wurden die Faktorladungen sowie die Fehlervarianzen korrespondierender Itemvariablen über die Testhälften hinweg paarweise gleichgesetzt (▶ Box 3.a), wie auch in den Ergebnissen zu sehen ist (▶ Box 3.b).

Das eindimensionale Modell mit den Restriktionen der essentiellen τ-Parallelität über die Testhälften hinweg ergab für das gegebene Beispiel einen guten Modellfit ($\chi^2(28) = 33.592$; $p = .215$, ▶ Box 3.b). Somit kann davon ausgegangen werden, dass die Voraussetzungen der Eindimensionalität und der essentiellen τ-Parallelität erfüllt sind. Die Split-Half-Reliabilität darf daher geschätzt werden (▶ Abschn. 3.2).

Box 3.b) R-Output zur Überprüfung der Voraussetzungen der Split-Half-Reliabilität

```
lavaan 0.6-7 ended normally after 15 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         16
  Number of equality constraints                     8

  Number of observations                           200

Model Test User Model:

  Test statistic                                33.592
  Degrees of freedom                                28
  P-value (Chi-square)                           0.215
# Der Modellfit ist hinreichend gut

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  ETA1 =~
    Y1        (p1)    1.180    0.075   15.752    0.000
    Y2        (p2)    1.164    0.071   16.424    0.000
    Y3        (p3)    1.187    0.072   16.537    0.000
    Y4        (p4)    1.050    0.065   16.095    0.000
    Y5        (p1)    1.180    0.075   15.752    0.000
    Y6        (p2)    1.164    0.071   16.424    0.000
    Y7        (p3)    1.187    0.072   16.537    0.000
    Y8        (p4)    1.050    0.065   16.095    0.000
# Die Faktorladungen korrespondierender Itemvariablen der Testhaelften
# wurden paarweise gleichgesetzt

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .Y1        (e1)    0.798    0.063   12.711    0.000
   .Y2        (e2)    0.611    0.049   12.356    0.000
   .Y3        (e3)    0.609    0.050   12.285    0.000
   .Y4        (e4)    0.561    0.045   12.543    0.000
   .Y5        (e1)    0.798    0.063   12.711    0.000
   .Y6        (e2)    0.611    0.049   12.356    0.000
   .Y7        (e3)    0.609    0.050   12.285    0.000
   .Y8        (e4)    0.561    0.045   12.543    0.000
    ETA1              1.000
# Die Fehlervarianzen der Itemvariablen wurden paarweise gleichgesetzt

# Der Modellfit ist gut, somit sind die Voraussetzungen der Eindimensionalität
# und der essentiell tau-parallelen Messungen erfüllt.
# Die Split-Half-Reliabilität kann nun anhand der klassischen Methode
# geschätzt werden.
```

3.2 Schätzung der Split-Half-Reliabilität

In der R-Syntax werden zunächst die beiden Halb-Testwertvariablen Half1 und Half2 gebildet. Dazu werden die entsprechenden Itemvariablen aufsummiert (▶ Box 3.c).

Box 3.c) R-Input und -Output zur Schätzung der Split-Half-Reliabilität (mit Spearman-Brown-Formel)

```
# Arbeitsverzeichnis auswählen ("Dateipfad/zum/Arbeitsverzeichnis")
setwd("C:/Daten/Kap14")

# Daten einlesen
# Trennzeichen: Tabstopps (sep="")
# Variablennamen stehen in erster Zeile (header=TRUE)
SplitHalf_Daten <- read.table("Split_Half_Labels.dat", sep="", header=TRUE)

# Halb-Testwertvariablen berechnen
# Die Itemvariablen Y1-Y4 gehoeren zur ersten Testhaelfte (Half1),
# die Itemvariablen Y5-Y8 zur zweiten Testhaelfte (Half2)
Half1 <- SplitHalf_Daten$Y1 + SplitHalf_Daten$Y2 + SplitHalf_Daten$Y3 +
SplitHalf_Daten$Y4
Half2 <- SplitHalf_Daten$Y5 + SplitHalf_Daten$Y6 + SplitHalf_Daten$Y7 +
SplitHalf_Daten$Y8

# Korrelation der Halb-Testwertvariablen
cor(Half1, Half2)
[1] 0.8894164

# Speichern der Korrelation als Halb-Testwert-Reliabilitaet:
Rel <- cor(Half1, Half2)

# Spearman-Brown-Formel der Testverlaengerung
SpearmanBrownRel <- 2*Rel/(1 + Rel)

# Split-Half-Reliabilitaet des Gesamttests nach Spearman-Brown-Korrektur
SpearmanBrownRel
[1] 0.9414721
```

Die Korrelation der zwei Halb-Testwertvariablen lässt sich über den cor-Befehl bestimmen. Für das Beispiel ergibt sich eine Korrelation der Halb-Testwertvariablen in Höhe von .889 (▶ Box 3.c). Diese Halbtest-Korrelation entspricht nur der Reliabilität eines Tests halber Länge und kann mithilfe der Spearman-Brown-Formel der Testverlängerung (▶ Kap. 14., Gl. 14.25, S. 324) rechnerisch auf die volle Testlänge aufgewertet werden, um so die Reliabilität des Gesamttests zu schätzen. Die Spearman-Brown-Formel wird als Befehl in die R-Syntax eingegeben (▶ Box 3.c). Der resultierende Wert (.941) kann als Reliabilität der Testwertvariablen des Gesamttests interpretiert werden, da die Voraussetzungen erfüllt sind. Für die Korrelation kann anhand der Fisher-Transformation auch ein Konfidenzintervall geschätzt werden. In unserem Beispiel ergibt die Spearman-Brown-Formel eine Reliabilität des Gesamttests in Höhe von .941 mit einem Konfidenzintervall von [.923; .955] (▶ Box 3.d).
Ein präzises, asymmetrisches Konfidenzintervall ließe sich in R alternativ direkt programmieren und schätzen (Schermelleh-Engel, Gäde & Irmer, 2021b).

Box 3.d) R-Input und -Output zur Schätzung des Konfidenzintervalls der Split-Half-Reliabilität (nach Spearman-Brown-Korrektur)

```
# Vor der ersten Anwendung: R-Package MBESS installieren, dazu einen
# nahegelegenen CRAN Mirror angeben
# Eine Liste der CRAN Mirrors: https://cran.r-project.org/mirrors.html
install.packages("MBESS", repos='https://cran.uni-muenster.de/')

# Vor jeder Anwendung muss das installierte Package MBESS zur Nutzung
# geladen werden
library(MBESS)

# Schaetzung des 95%-Konfidenzintervalls: Durch die Fisher-Transformation
# resultiert ein nicht symmetrisches Konfidenzintervall
# Korrelation aus vorheriger Analyse und Stichprobengroesse angeben
ci.cc(0.9414721, 200, conf.level = 0.95)

$Lower.Limit                          # Untere Grenze des Konfidenzintervalls
[1] 0.9233383

$Estimated.Correlation                # Korrelation (Split-Half-Reliabilität)
[1] 0.9414721

$Upper.Limit                          # Obere Grenze des Konfidenzintervalls
[1] 0.9554159
```

4 Retest-Reliabilität

4.1 Überprüfung der Voraussetzungen der Retest-Reliabilität

Die Korrelation zweier wiederholt gemessener Testwertvariablen darf nur dann als Reliabilität der Testwertvariablen interpretiert werden, wenn strenge Voraussetzungen erfüllt sind. Die Voraussetzungen der Retest-Reliabilität sind die Eindimensionalität sowie die essentielle τ-Parallelität der Messungen über die beiden Messzeitpunkte hinweg. Diese Voraussetzungen können mittels CFA überprüft werden (▶ Box 4.a).

Box 4.a) R-Input zur Überprüfung der Voraussetzungen der Retest-Reliabilität

```
# Vor der ersten Anwendung: R-Package lavaan zur Analyse von latenten
# Variablenmodellen installieren, dazu einen nahegelegenen CRAN Mirror angeben
# Eine Liste der CRAN Mirrors: https://cran.r-project.org/mirrors.html
install.packages("lavaan", repos='https://cran.uni-muenster.de/')

# Vor jeder Anwendung muss das installierte Package lavaan zur Nutzung
# geladen werden
library(lavaan)

# Arbeitsverzeichnis auswählen ("Dateipfad/zum/Arbeitsverzeichnis")
setwd("C:/Daten/Kap14")

# Daten einlesen:
# Trennzeichen: Tabstopps (sep="")
# Variablennamen stehen in erster Zeile (header=TRUE)
Retest_Daten <- read.table("Retest_Labels.dat", sep="", header=TRUE)

# Definition des zweifaktoriellen CFA-Modells mod2
# Faktor ETA1 entspricht der Messung zum 1. Messzeitpunkt,
# Faktor ETA2 der Messung zum 2. Messzeitpunkt
# Die Faktorladungen der einzelnen Itemvariablen erhalten zu beiden
# Messzeitpunkten dasselbe Label und werden damit gleichgesetzt
# (Bedingung der essent. Tau-Parallelitaet)
mod2 <- 'ETA1 =~ p1*Y11 + p2*Y21 + p3*Y31 + p4*Y41
         ETA2 =~ p1*Y12 + p2*Y22 + p3*Y32 + p4*Y42

# Den Fehlervarianzen der Itemvariablen werden zu beiden Messzeitpunkten die
# Labels e1 bis e4 zugewiesen. Damit ist für jede Itemvariable die
# Fehlervarianz zu beiden Messzeitpunkten gleichgesetzt
# (Bedingung der essent. # Tau-Parallelitaet)
# Itemvariablen – gemessen zum 1.Messzeitpunkt
Y11~~e1*Y11
Y21~~e2*Y21
Y31~~e3*Y31
Y41~~e4*Y41
# Itemvariablen – gemessen zum 2. Messzeitpunkt
Y12~~e1*Y12
Y22~~e2*Y22
Y32~~e3*Y32
Y42~~e4*Y42
'
# Analyse des Modells mod2
# ML-Schätzung (estimator= "ML")
# alle latenten Varianzen auf eins fixiert (std.lv=TRUE)
fit.mod2 <- sem(mod2, data= Retest_Daten, estimator= "ML", std.lv=TRUE)

# Ergebnisse anzeigen
summary(fit.mod2)
```

Im vorliegenden Beispiel sind die Itemvariablen Y11-Y41 dem ersten Messzeitpunkt (t1) zugeordnet und die Itemvariablen Y12-Y42 dem zweiten Messzeitpunkt (t2). Zur Überprüfung der Voraussetzungen wurden die

Faktorladungen sowie die Fehlervarianzen für jede Itemvariable über die Messzeitpunkte hinweg gleichgesetzt (▶ Box 4.a), wie auch in den Ergebnissen zu sehen ist (▶ Box 4.b).

Das korrelierte zweifaktorielle Modell mit den Restriktionen der essentiellen τ-Parallelität über die Zeit ergab für das vorliegende Beispiel einen hinreichend guten Modellfit ($\chi^2(27) = 20.637$; $p = .803$, ▶ Box 4.b). Somit kann davon ausgegangen werden, dass die Voraussetzungen der Eindimensionalität und der essentiellen τ-Parallelität erfüllt sind. Die Retest-Reliabilität darf daher geschätzt werden (▶ Absch. 4.2).

Box 4.b) R-Output zur Überprüfung der Voraussetzungen der Retest-Reliabilität

```
lavaan 0.6-7 ended normally after 19 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of free parameters                         17
  Number of equality constraints                     8

  Number of observations                           200

Model Test User Model:

  Test statistic                                20.637
  Degrees of freedom                                27
  P-value (Chi-square)                           0.803
# Der Modellfit ist hinreichend gut

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  ETA1 =~
    Y11       (p1)    0.911    0.048   19.066    0.000
    Y21       (p2)    0.788    0.048   16.487    0.000
    Y31       (p3)    0.699    0.049   14.345    0.000
    Y41       (p4)    0.593    0.044   13.462    0.000
  ETA2 =~
    Y12       (p1)    0.911    0.048   19.066    0.000
    Y22       (p2)    0.788    0.048   16.487    0.000
    Y32       (p3)    0.699    0.049   14.345    0.000
    Y42       (p4)    0.593    0.044   13.462    0.000
# Die Faktorladungen der Itemvariablen wurden über die Zeit
# hinweg gleichgesetzt

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  ETA1 ~~
    ETA2              0.789    0.037   21.437    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .Y11       (e1)    0.178    0.026    6.877    0.000
   .Y21       (e2)    0.351    0.031   11.219    0.000
   .Y31       (e3)    0.489    0.039   12.566    0.000
   .Y41       (e4)    0.436    0.034   12.885    0.000
   .Y12       (e1)    0.178    0.026    6.877    0.000
   .Y22       (e2)    0.351    0.031   11.219    0.000
   .Y32       (e3)    0.489    0.039   12.566    0.000
   .Y42       (e4)    0.436    0.034   12.885    0.000
    ETA1              1.000
    ETA2              1.000
# Die Fehlervarianzen der Itemvariablen wurden über die Zeit gleichgesetzt

# Der Modellfit ist gut, somit sind die Voraussetzungen erfüllt. Die Retest-
# Reliabilität kann nun anhand der klassischen Formel berechnet werden.
```

4.2 Schätzung der Retest-Reliabilität

In der R-Syntax werden zunächst die beiden Testwertvariablen Test_t1 und Test_t2 gebildet. Dazu werden die entsprechenden Itemvariablen aufsummiert (▶ Box 4.c).

Box 4.c) R-Input und -Output zur Schätzung der Retest-Reliabilität

```
# Arbeitsverzeichnis auswählen ("Dateipfad/zum/Arbeitsverzeichnis")
setwd("C:/Daten/Kap14")

# Daten einlesen
# Trennzeichen: Tabstopps (sep="")
# Variablennamen stehen in erster Zeile (header=TRUE)
Retest_Daten <- read.table("Retest_Labels.dat", sep="", header=TRUE)

# Testwertvariable berechnen zu den Messzeitpunkten t1 und t2
Test_t1 <- Retest_Daten$Y11 + Retest_Daten$Y21 + Retest_Daten$Y31 +
Retest_Daten$Y41
Test_t2 <- Retest_Daten$Y12 + Retest_Daten$Y22 + Retest_Daten$Y32 +
Retest_Daten$Y42

# Schätzung der Korrelation zwischen den Testwertvariablen zu den zwei
# Messzeitpunkten
cor(Test_t1, Test_t2)
[1] 0.7016045
```

Die Korrelation der zwei Testwertvariablen kann über den cor-Befehl bestimmt werden und darf als Retest-Reliabilität interpretiert werden, da die Voraussetzungen erfüllt sind. Für die Korrelation kann anhand der Fisher-Transformation auch ein Konfidenzintervall geschätzt werden. Für das Beispiel ergibt sich eine Retest-Reliabilität in Höhe von .702 (▶ Box 4.c) mit einem Konfidenzintervall von [.624; .766] (▶ Box 4.d). Ein präzises, asymmetrisches Konfidenzintervall ließe sich alternativ direkt programmieren und schätzen (Schermelleh-Engel, Gäde & Irmer, 2021b).

Box 4.d) R-Input und -Output zur Schätzung des Konfidenzintervalls der Retest-Reliabilität

```
# Vor der ersten Anwendung: R-Package MBESS installieren, dazu einen
# nahegelegenen CRAN Mirror angeben
# Eine Liste der CRAN Mirrors: https://cran.r-project.org/mirrors.html
install.packages("MBESS", repos='https://cran.uni-muenster.de/')

# Vor jeder Anwendung muss das installierte Package MBESS zur Nutzung
# geladen werden
library(MBESS)

# Schaetzung des 95%-Konfidenzintervalls: Durch die Fisher-Transformation
# resultiert ein nicht symmetrisches Konfidenzintervall
# Korrelation aus vorheriger Analyse und Stichprobengroesse angeben
ci.cc(0.7016045, 200, conf.level = 0.95)

$Lower.Limit                          # Untere Grenze des Konfidenzintervalls
[1] 0.6235618

$Estimated.Correlation                # Korrelation (Retest-Reliabilität)
[1] 0.7016045

$Upper.Limit                          # Obere Grenze des Konfidenzintervalls
[1] 0.7658014
```