Basics (and advanced stuff) of questionnaire development and analysis

Julius Fenn

26 04 2021

Contents

This is an R Markdown document. Instructions for writing these documents and background information can be found in the book R Markdown: The Definitive Guide. When you execute code within the document, the results appear beneath the code.

show data set

Datensatz ist der Campus-File des IQB-Ländervergleichs 2011 der Primarstufe (Zugang über https://www.iqb.hu-berlin.de/fdz/Datenzugang), Bedeutung Variablen einsichtig über Suchfunktion Skalenhandbuch.

```
dim(datenLV)
```

```
## [1] 3005   33
```

```
knitr::kable(datenLV[1:4,], digits = 2)
```

idsch_FDZ	idstud_FDZ	tr_sex	tr_age	Emigr	EDezh	EHisei	EHisced	SBuecher	SHausZt	tr_[illegible]	Not_[illegible]	Not_[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1	1	maennlich	10.42	keinMig	Kind spricht zu Hause immer oder fast immer Deutsch	55	ISCED level 5B	100 Buecher	30 Minuten bis zu einer Stunde	2	2	nein	4	1	4	4	3	3	3	4	4	3	1	1	4.00	3.00	3.75	-0.17	-0.63	-0.
1	2	maennlich	9.83	Mig	Kind spricht zu Hause immer oder fast immer Deutsch	49	ISCED level 5B	mehr 200 Buecher	30 Minuten bis zu einer Stunde	2	2	nein	2	2	2	2	3	1	3	3	3	3	1	1	2.25	3.25	3.50	-0.44	-0.98	0.
1	3	maennlich	10.50	NA	Kind spricht zu Hause nie oder manchmal Deutsch	NA	NA	25 Buecher	30 Minuten bis zu einer Stunde	4	3	nein	3	1	4	3	3	1	3	3	4	1	1	2	3.50	3.25	3.00	-1.21	-1.06	-0.
1	4	maennlich	10.67	NA	NA	NA	NA	10 Buecher	weniger als 30 Minuten	2	3	nein	3	3	4	4	4	1	4	4	1	4	4	4	3.25	4.00	1.75	0.26	-1.17	0.

```
summary(datenLV)
```

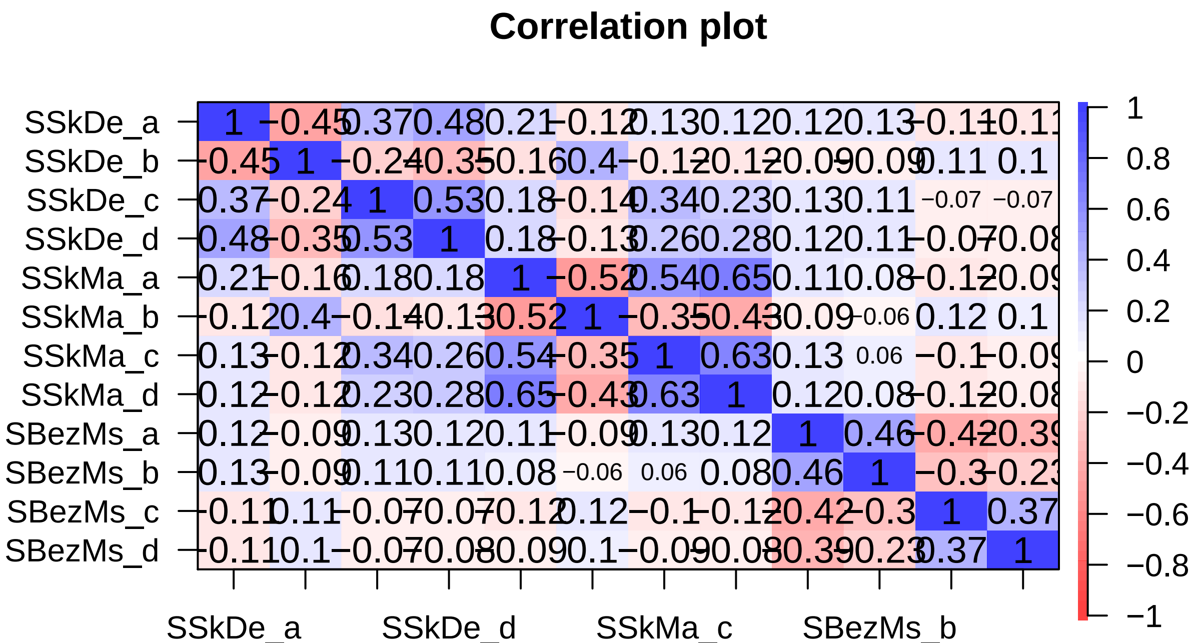
```
##    idsch_FDZ       idstud_FDZ         tr_sex         tr_age
##  Min.   :  1.0   Min.   :   1   maennlich:1530   Min.   : 6.833
##  1st Qu.: 53.0   1st Qu.: 752   weiblich :1475   1st Qu.:10.083
##  Median :104.0   Median :1503                    Median :10.417
##  Mean   :103.3   Mean   :1503                    Mean   :10.425
##  3rd Qu.:155.0   3rd Qu.:2254                    3rd Qu.:10.750
##  Max.   :201.0   Max.   :3005                    Max.   :13.000
##                                                  NA's   :7
##      Emigr                                                    EDezh
##  Mig    : 493   Kind spricht zu Hause immer oder fast immer Deutsch:2346
##  keinMig:1966   Kind spricht zu Hause nie oder manchmal Deutsch    : 191
##  NA's   : 546   NA's                                               : 468
##
##
```

```
##
##
##      EHisei              EHisced_akt               SBuecher
##  Min.   :10.00   ISCED level 1 :  23   10 Buecher      : 153
##  1st Qu.:37.00   ISCED level 2 :  81   25 Buecher      : 557
##  Median :48.00   ISCED level 3A:  10   100 Buecher     :1112
##  Mean   :49.57   ISCED level 5B:1556   200 Buecher     : 563
##  3rd Qu.:61.00   ISCED level 5A: 724   mehr 200 Buecher: 550
##  Max.   :89.00   ISCED level 6 : 126   NA's            :  70
##  NA's   :622     NA's          : 485
##                              SLesZt        tr_NotDe       tr_NotMa
##  weniger als 30 Minuten        : 862   Min.   :1.00   Min.   :1.000
##  30 Minuten bis zu einer Stunde:1173   1st Qu.:2.00   1st Qu.:2.000
##  1-2 Stunden                   : 475   Median :2.00   Median :2.000
##  2 Stunden oder mehr           : 398   Mean   :2.46   Mean   :2.509
##  NA's                          :  97   3rd Qu.:3.00   3rd Qu.:3.000
##                                        Max.   :5.00   Max.   :5.000
##                                        NA's   :131    NA's   :127
##  tr_Wdh_r       SSkDe_a         SSkDe_b         SSkDe_c         SSkDe_d
##  nein:2830   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  ja  : 168   1st Qu.:3.000   1st Qu.:1.000   1st Qu.:3.000   1st Qu.:3.000
##  NA's:   7   Median :3.000   Median :2.000   Median :3.000   Median :3.000
##              Mean   :3.116   Mean   :2.126   Mean   :3.303   Mean   :3.334
##              3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:4.000
##              Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :4.000
##              NA's   :92      NA's   :121     NA's   :126     NA's   :123
##     SSkMa_a        SSkMa_b         SSkMa_c         SSkMa_d         SBezMs_a
##  Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.00   1st Qu.:1.000   1st Qu.:3.000   1st Qu.:3.000   1st Qu.:3.000
##  Median :3.00   Median :2.000   Median :3.000   Median :4.000   Median :3.000
##  Mean   :3.17   Mean   :2.065   Mean   :3.314   Mean   :3.331   Mean   :3.365
##  3rd Qu.:4.00   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :4.00   Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :4.000
##  NA's   :82     NA's   :115     NA's   :121     NA's   :102     NA's   :164
##     SBezMs_b       SBezMs_c        SBezMs_d         SSkDe           SSkMa
##  Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.00   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.750   1st Qu.:2.750
##  Median :3.00   Median :1.000   Median :1.000   Median :3.250   Median :3.250
##  Mean   :3.11   Mean   :1.633   Mean   :1.493   Mean   :3.156   Mean   :3.187
##  3rd Qu.:4.00   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:3.750   3rd Qu.:4.000
##  Max.   :4.00   Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :4.000
##  NA's   :199    NA's   :173     NA's   :172     NA's   :95      NA's   :85
##      SBezMs        wle_lesen          wle_hoeren        wle_mathe
##  Min.   :1.000   Min.   :-5.06686   Min.   :-5.8078   Min.   :-3.4768
##  1st Qu.:3.000   1st Qu.:-0.68091   1st Qu.:-0.5796   1st Qu.:-0.6384
##  Median :3.500   Median : 0.12715   Median : 0.1526   Median : 0.1035
##  Mean   :3.335   Mean   : 0.09367   Mean   : 0.1094   Mean   : 0.1061
##  3rd Qu.:3.750   3rd Qu.: 0.88132   3rd Qu.: 0.8188   3rd Qu.: 0.8379
##  Max.   :4.000   Max.   : 4.24000   Max.   : 3.5043   Max.   : 4.7832
##  NA's   :145
##                          schoolEconDis         schoolMiganteil
##  <33% oekonomisch benachteiligt: 175   < 20% Miganteil:1645
##  33-66% oekonomisch mittel     :2320   > 20% Miganteil:1360
##  >66% oekonomisch bevorzugt    : 510
```

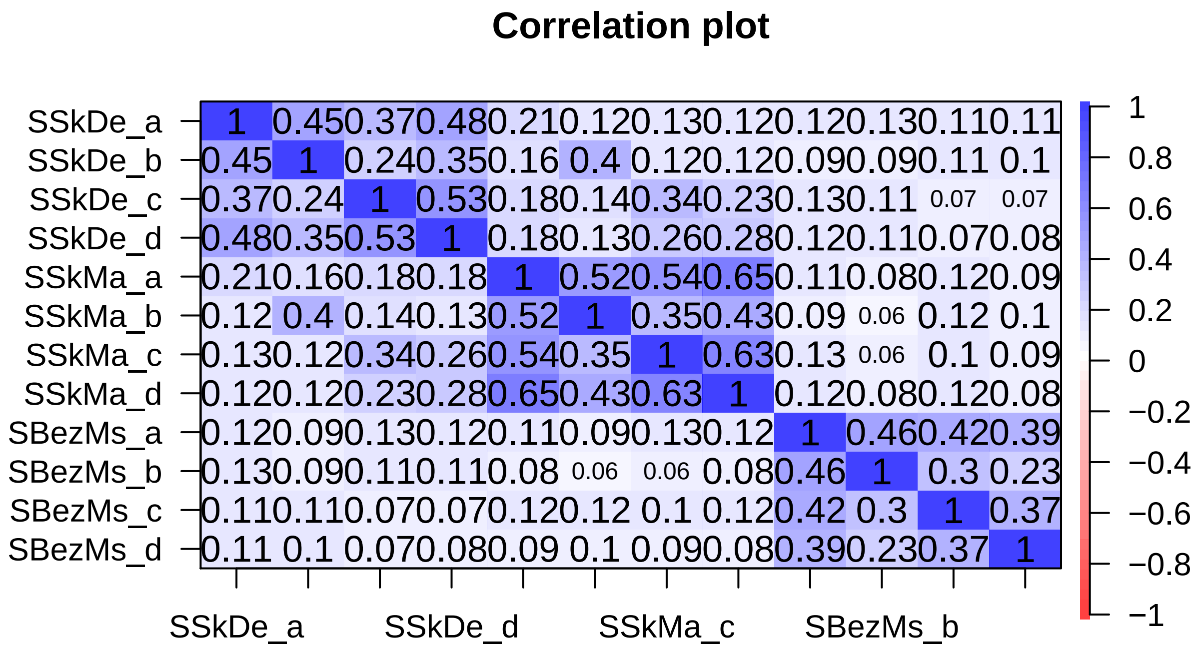
```
##
##
##
##
```

tidy and transform data

```
psych::corPlot(r = datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBezMs_"]
```




```
# recode
datenLV$SSkMa_b <- 5 - datenLV$SSkMa_b
datenLV$SSkDe_b  <- 5 - datenLV$SSkDe_b
datenLV$SBezMs_c <- 5 - datenLV$SBezMs_c
datenLV$SBezMs_d <- 5 - datenLV$SBezMs_d

psych::corPlot(r = datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBezMs_"]
```



describe, visualize your data and get a first impression using R base functions

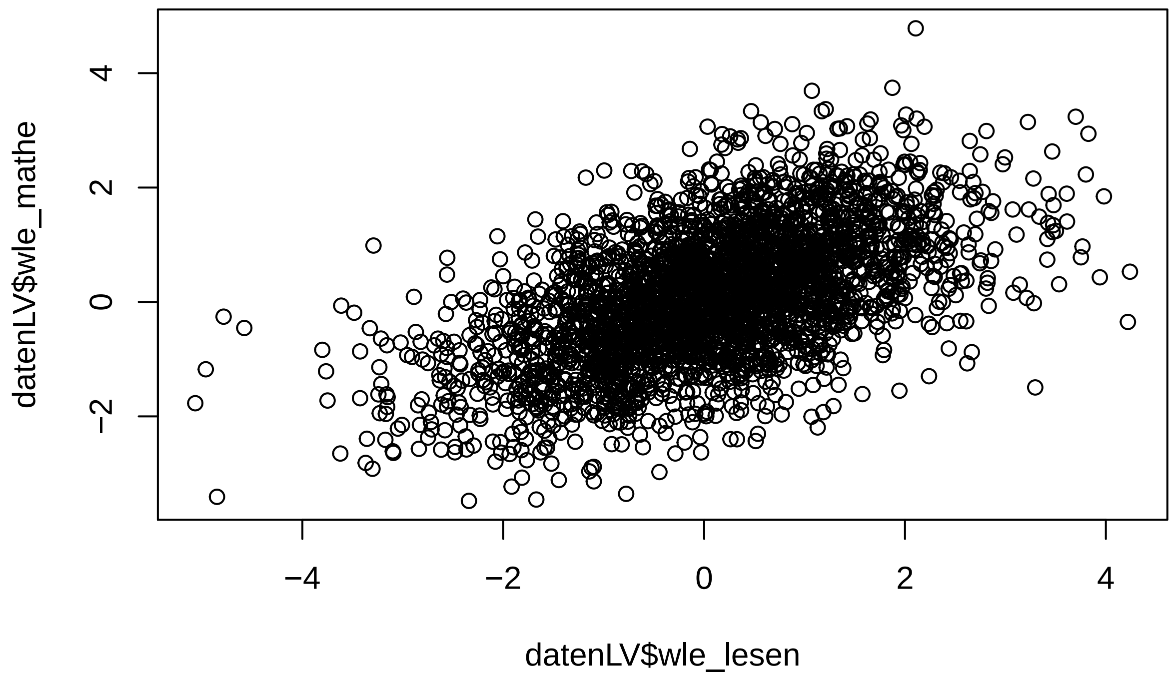
Insgesamt weisst der Datensatz N=3005 Schüler/innen in 201 Schulen auf. Die Schulen weisen folgende Anzahl von Schüler/innen auf:

```r
proSchule <- aggregate(datenLV$idsch_FDZ,by=list(datenLV$idsch_FDZ),FUN=length) # using base functions
summary(proSchule$x)
```

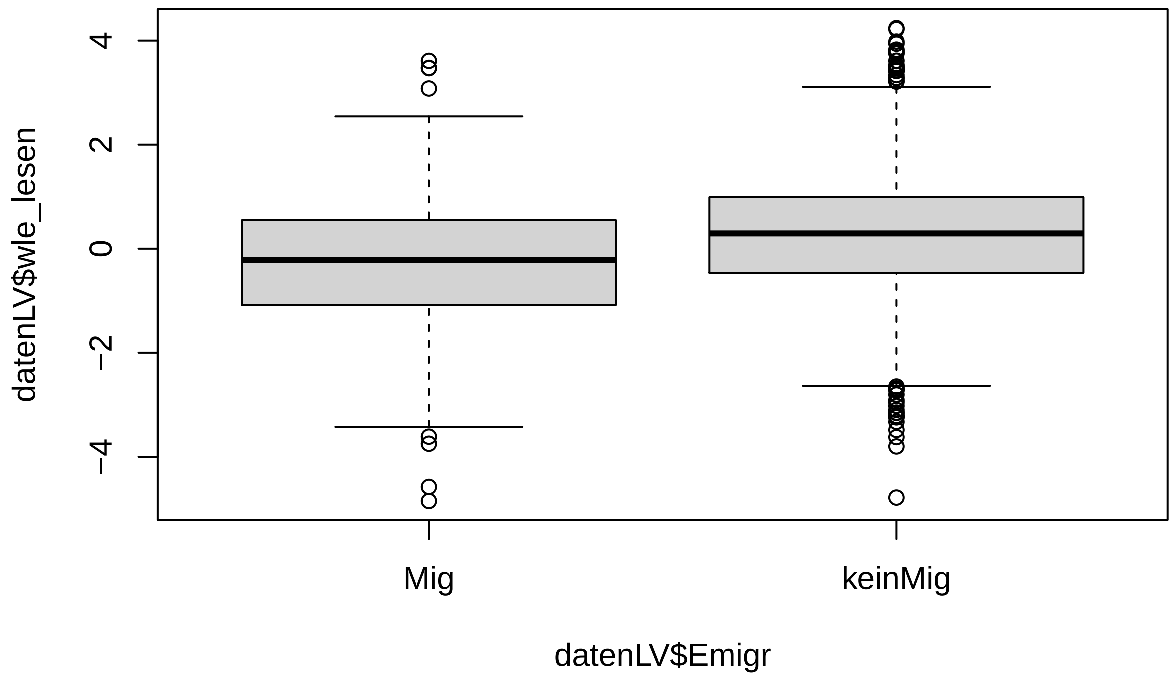
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   10.00   15.00   14.95   20.00   20.00
```

```r
rm(proSchule)
```

```r
plot(datenLV$wle_lesen, datenLV$wle_mathe)
```




```r
boxplot(datenLV$wle_lesen ~ datenLV$Emigr)
```




```
t.test(datenLV$wle_lesen ~ datenLV$Emigr) # = unequal variances t-test
```

```
##
##  Welch Two Sample t-test
##
## data:  datenLV$wle_lesen by datenLV$Emigr
## t = -8.6373, df = 720.46, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6554145 -0.4126442
## sample estimates:
##     mean in group Mig mean in group keinMig
##            -0.2612065             0.2728229
```

Bestehen lineare Zusammenhänge mit einer (normalverteilten) numerischen Variablen?

```
## linear regression
summary(lm(formula = wle_lesen ~ Emigr*tr_sex + SSkMa, data = datenLV))
```

```
##
## Call:
## lm(formula = wle_lesen ~ Emigr * tr_sex + SSkMa, data = datenLV)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3692 -0.7007  0.0184  0.7098  3.8719
##
## Coefficients:
```

```
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -1.98582    0.12494 -15.894  < 2e-16 ***
## EmigrkeinMig                  0.39521    0.07802   5.065 4.38e-07 ***
## tr_sexweiblich                0.30605    0.10250   2.986  0.00286 **
## SSkMa                         0.52233    0.03229  16.174  < 2e-16 ***
## EmigrkeinMig:tr_sexweiblich  0.04767    0.11380   0.419  0.67535
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.108 on 2399 degrees of freedom
##   (601 observations deleted due to missingness)
## Multiple R-squared:  0.1324, Adjusted R-squared:  0.1309
## F-statistic: 91.49 on 4 and 2399 DF,  p-value: < 2.2e-16
```

Bestehen lineare Zusammenhänge mit einer binären Variablen? empfohlene Seite: https://www.methodenberatung.uzh.ch/de/datenanalyse_spss/zusammenhaenge/lreg.html

```r
## logistic regression (if > 2 -> ordinal logistic regression)
summary(glm(Emigr ~ wle_lesen+wle_hoeren+wle_mathe,
               data = datenLV, family = binomial))
```

```
##
## Call:
## glm(formula = Emigr ~ wle_lesen + wle_hoeren + wle_mathe, family = binomial,
##     data = datenLV)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3848   0.4196   0.5748   0.6953   1.4128
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.36913    0.05226  26.198  < 2e-16 ***
## wle_lesen    0.17045    0.05383   3.167  0.00154 **
## wle_hoeren   0.12008    0.05871   2.045  0.04081 *
## wle_mathe    0.33122    0.05930   5.585 2.33e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2464.3  on 2458  degrees of freedom
## Residual deviance: 2338.4  on 2455  degrees of freedom
##   (546 observations deleted due to missingness)
## AIC: 2346.4
##
## Number of Fisher Scoring iterations: 4
```

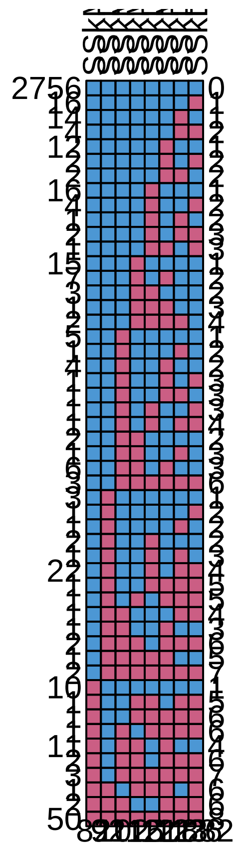
```r
exp(coef(glm(Emigr ~ wle_lesen+wle_hoeren+wle_mathe,
             data = datenLV, family = binomial))) - 1
```

```
## (Intercept)   wle_lesen  wle_hoeren   wle_mathe
##   2.9319093   0.1858348   0.1275879   0.3926618
```

Anmerkung: Hypothestentest, logistische Regression sind die zentralen Verfahren für die **deduktive Methode** der Itementwicklung

missing data

```r
## missing data patterns
mdpattern <- mice::md.pattern(x = datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SS
```



```r
## sorted by # missing
mdpattern[match(x = sort(x = as.numeric(rownames(mdpattern)), decreasing = TRUE), table = as.numeric(ro
```

```
##      SSkMa_a SSkDe_a SSkMa_d SSkMa_b SSkDe_b SSkMa_c SSkDe_d SSkDe_c
## 2756       1       1       1       1       1       1       1       1 0
## 50         0       0       0       0       0       0       0       0 8
## 22         1       0       1       1       0       1       0       0 4
## 16         1       1       1       1       1       1       1       0 1
## 16         1       1       1       1       1       1       1       0 1
## 15         1       1       1       0       1       1       1       1 1
## 14         1       1       1       1       1       1       0       1 1
## 12         1       1       1       1       1       0       1       1 1
## 11         0       1       0       0       1       0       1       1 4
## 10         0       1       1       1       1       1       1       1 1
## 7          1       1       1       0       1       0       1       1 2
## 6          1       1       0       0       1       0       1       1 3
## 5          1       1       0       1       1       1       1       1 1
## 4          1       1       1       1       1       1       0       0 2
## 4          1       1       1       1       1       1       0       0 2
## 4          1       1       1       1       1       1       0       0 2
## 3          1       1       1       0       0       1       1       1 2
## 3          1       1       1       0       0       1       1       1 2
```

```
## 3          1       1       1       0       0       1       1       1 2
## 3          1       1       1       0       0       1       1       1 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 2          1       1       1       1       1       0       1       0 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
## 1          1       1       1       1       0       1       0       1 2
```

```r
## generate dummy of missing variable to identify potential helper variables
datenLV$missing_SSkMa_a <- ifelse(test = is.na(datenLV$SSkMa_a), yes = 1, no = 0)
helpervars <- c("wle_lesen", "wle_hoeren", "SSkDe") # include normally many more
for(v in helpervars){
  tmp <- t.test(datenLV[[v]] ~ datenLV$missing_SSkMa_a)
  if(tmp$p.value < .05){
    print(v)
    print(tmp)
  }
}
```

```
## [1] "wle_lesen"
##
##  Welch Two Sample t-test
##
## data:  datenLV[[v]] by datenLV$missing_SSkMa_a
## t = 4.0297, df = 84.449, p-value = 0.0001217
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3177873 0.9369377
## sample estimates:
```

```
## mean in group 0 mean in group 1
##       0.1107931      -0.5165693
##
## [1] "wle_hoeren"
##
##  Welch Two Sample t-test
##
## data:  datenLV[[v]] by datenLV$missing_SSkMa_a
## t = 4.4992, df = 83.294, p-value = 2.193e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.4094216 1.0581596
## sample estimates:
## mean in group 0 mean in group 1
##       0.1294574      -0.6043332
```

```r
## overall missing for each single variable
round(x = sort(x = colSums(x = is.na(datenLV), na.rm = TRUE), decreasing = TRUE) / nrow(datenLV) * 100,
```

```
##          EHisei           Emigr     EHisced_akt           EDezh        SBezMs_b
##           20.70           18.17           16.14           15.57            6.62
##        SBezMs_c        SBezMs_d        SBezMs_a          SBezMs        tr_NotDe
##            5.76            5.72            5.46            4.83            4.36
##        tr_NotMa        SSkDe_c         SSkDe_d         SSkDe_b         SSkMa_c
##            4.23            4.19            4.09            4.03            4.03
##         SSkMa_b         SSkMa_d          SLesZt           SSkDe         SSkDe_a
##            3.83            3.39            3.23            3.16            3.06
##           SSkMa         SSkMa_a        SBuecher          tr_age        tr_Wdh_r
##            2.83            2.73            2.33            0.23            0.23
##       idsch_FDZ      idstud_FDZ          tr_sex       wle_lesen      wle_hoeren
##            0.00            0.00            0.00            0.00            0.00
##       wle_mathe   schoolEconDis schoolMiganteil missing_SSkMa_a
##            0.00            0.00            0.00            0.00
```

Es ist zentral fehlende Daten zu ersetzen bzw. modellbasiert zu schätzen. Die zwei modernsten Ansätze, um fehlende Daten zu ersetzen sind:

- multiple imputation, introductory book: https://stefvanbuuren.name/fimd/, Grund, Lüdtke, and Robitzsch (2018)
- full information maximum likelihood (aktuell in Mplus, aber nicht in R implementiert)

outlier analysis

Es wird unterschieden in uni- und multivariate Ausreißer, da *structural equation modelling / CFA* multivariate Verfahren sind (mehrere UVs und AVs), ist es notwendig die Daten auf multivariate Ausreißer zu kontrollieren. Dafür eignet sich die **Mahalanobis Distance**:

```r
## exemplify Mahalanobis Distance
sigma <- matrix(c(4,1,2,1,5,4,2,4,6), ncol = 3)
cov2cor(sigma)
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.2236068 0.4082483
## [2,] 0.2236068 1.0000000 0.7302967
## [3,] 0.4082483 0.7302967 1.0000000
```

```r
means <- c(0, 0, 0)
set.seed(42)
n <- 1000
x <- rmvnorm(n = n, mean = means, sigma = sigma)
d <- data.frame(x)
p4 <- plot_ly(d, x = ~ X1, y = ~ X2, z = ~ X3,
              marker = list(color = ~ X2,
                            showscale = TRUE)) %>%
  add_markers()

p4
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

```r
## identify multivariate outliers
d$mahal <- mahalanobis(d, colMeans(d), cov(d))
d$p_mahal <- pchisq(d$mahal, df=2, lower.tail=FALSE)
d[d$p_mahal < .001, ]
```

```
##            X1        X2        X3    mahal       p_mahal
## 274  5.759481 -4.929943 -1.450310 16.41275 0.0002729077
## 330  7.398060  3.344539  4.741212 14.50380 0.0007088271
## 980 -6.295530 -5.365858 -4.367558 14.17879 0.0008339001
```

```r
datenLV$mahal_SSkMa <- mahalanobis(datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_")]
datenLV$p_mahal_SSkMa <- pchisq(datenLV$mahal_SSkMa, df=3, lower.tail=FALSE)

## identify multivariate outliers
head(datenLV[datenLV$p_mahal_SSkMa < .001  & !is.na(datenLV$p_mahal_SSkMa), c("SSkMa_a", "SSkMa_b", "SS
```

```
##    SSkMa_a SSkMa_b SSkMa_c SSkMa_d mahal_SSkMa p_mahal_SSkMa
## 8        1       1       4       1    22.80886  4.426227e-05
## 9        1       3       1       4    29.61531  1.662648e-06
## 15       1       1       4       4    17.85799  4.705288e-04
## 25       4       4       1       4    22.55417  5.001386e-05
## 46       4       1       1       4    28.13860  3.396688e-06
## 58       2       4       1       1    17.17129  6.516648e-04
```

```
sum(datenLV$p_mahal_SSkMa < .001, na.rm = TRUE)
```

```
## [1] 112
```

```
datenLV$intravariability_SSkMa <- apply(datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSk
## identify insufficient item responding using variability of answering patterns
head(datenLV[datenLV$intravariability_SSkMa == 0  & !is.na(datenLV$intravariability_SSkMa), c("SSkMa_a"
```

```
##    SSkMa_a SSkMa_b SSkMa_c SSkMa_d mahal_SSkMa p_mahal_SSkMa
## 4        4       4       4       4   1.3060837      0.727689
## 5        3       3       3       3   0.2810342      0.963555
## 6        4       4       4       4   1.3060837      0.727689
## 7        4       4       4       4   1.3060837      0.727689
## 11       4       4       4       4   1.3060837      0.727689
## 13       3       3       3       3   0.2810342      0.963555
```

```
sum(datenLV$intravariability_SSkMa == 0, na.rm = TRUE)
```

```
## [1] 1049
```

[simulation study: standardized residuals, reliability]

Using the R-Package `simstudy` it is possible to generate all kinds of data:

I have generated a data set with 3 items (`y1-y3`) and a data set with 7 items (`m1-m7`) for different sample sizes. The variables `latentvar` and `errorvar` are unknown and for example important in the context of classical test theory as these correspond to the true and error variance):

```
##       varname   formula     variance   dist     link
##  1: latentvar        20          0.5 normal identity
##  2:  errorvar         4          0.5 normal identity
##  3:        y1 latentvar errorvar / 4 normal identity
##  4:        y2 latentvar errorvar / 4 normal identity
##  5:        y3 latentvar errorvar / 4 normal identity
##  6:        m1 latentvar errorvar / 4 normal identity
##  7:        m2 latentvar errorvar / 4 normal identity
##  8:        m3 latentvar errorvar / 4 normal identity
##  9:        m4 latentvar errorvar / 4 normal identity
## 10:        m5 latentvar errorvar / 4 normal identity
## 11:        m6 latentvar errorvar / 4 normal identity
## 12:        m7 latentvar errorvar / 4 normal identity
```

```
set.seed(111)
dt_50 <- genData(50, def); dt_50 <- as.data.frame(dt_50)
dt_200 <- genData(200, def); dt_200 <- as.data.frame(dt_200)
dt_500 <- genData(500, def); dt_500 <- as.data.frame(dt_500)
dt_100000 <- genData(100000, def); dt_100000 <- as.data.frame(dt_100000)
```

```r
round(x = cor(dt_50[, str_subset(string = colnames(dt_50), pattern = "m")]), digits = 2)
```

```
##      m1   m2   m3   m4   m5   m6   m7
## m1 1.00 0.45 0.39 0.34 0.35 0.43 0.24
## m2 0.45 1.00 0.52 0.34 0.56 0.40 0.49
## m3 0.39 0.52 1.00 0.25 0.44 0.39 0.57
## m4 0.34 0.34 0.25 1.00 0.19 0.30 0.35
## m5 0.35 0.56 0.44 0.19 1.00 0.20 0.37
## m6 0.43 0.40 0.39 0.30 0.20 1.00 0.49
## m7 0.24 0.49 0.57 0.35 0.37 0.49 1.00
```

```r
round(x = cor(dt_100000[, str_subset(string = colnames(dt_100000), pattern = "m")]), digits = 2)
```

```
##      m1   m2   m3   m4   m5   m6   m7
## m1 1.00 0.34 0.33 0.34 0.33 0.33 0.34
## m2 0.34 1.00 0.33 0.34 0.34 0.34 0.34
## m3 0.33 0.33 1.00 0.33 0.33 0.33 0.34
## m4 0.34 0.34 0.33 1.00 0.33 0.34 0.34
## m5 0.33 0.34 0.33 0.33 1.00 0.33 0.33
## m6 0.33 0.34 0.33 0.34 0.33 1.00 0.34
## m7 0.34 0.34 0.34 0.34 0.33 0.34 1.00
```

```r
round(x = cor(dt_50[, str_subset(string = colnames(dt_50), pattern = "m")]), digits = 2) - round(x = co
```

```
##       m1   m2    m3    m4    m5    m6    m7
## m1  0.00 0.11  0.06  0.00  0.02  0.10 -0.10
## m2  0.11 0.00  0.19  0.00  0.22  0.06  0.15
## m3  0.06 0.19  0.00 -0.08  0.11  0.06  0.23
## m4  0.00 0.00 -0.08  0.00 -0.14 -0.04  0.01
## m5  0.02 0.22  0.11 -0.14  0.00 -0.13  0.04
## m6  0.10 0.06  0.06 -0.04 -0.13  0.00  0.15
## m7 -0.10 0.15  0.23  0.01  0.04  0.15  0.00
```

```r
sd(dt_50$m1) / sqrt(x = length(dt_50$m1))
```

```
## [1] 0.18367
```

```r
sd(dt_100000$m1) / sqrt(x = length(dt_100000$m1))
```

```
## [1] 0.003876399
```

```r
psych::alpha(cor(dt_50[, str_subset(string = colnames(dt_50), pattern = "m")]))$total
```

```
##  raw_alpha std.alpha   G6(smc) average_r      S/N  median_r
##  0.8132445 0.8132445 0.8146988 0.3835094 4.354593 0.3878958
```

```r
psych::alpha(cor(dt_200[, str_subset(string = colnames(dt_200), pattern = "m")]))$total
```

```
##  raw_alpha std.alpha   G6(smc) average_r      S/N  median_r
##  0.7631399 0.7631399 0.7410928 0.3151959 3.221902 0.3114825
```

```r
psych::alpha(cor(dt_100000[, str_subset(string = colnames(dt_100000), pattern = "m")]))$total
```

```
##  raw_alpha std.alpha   G6(smc) average_r      S/N  median_r
##  0.7781921 0.7781921 0.7504623 0.3338666 3.508406 0.3353357
```

```r
psych::alpha(cor(dt_50[, str_subset(string = colnames(dt_50), pattern = "y")]))$total
```

```
##  raw_alpha std.alpha   G6(smc) average_r     S/N  median_r
##  0.6202657 0.6202657 0.5319161 0.3525301 1.63342 0.3163457
```

```r
psych::alpha(cor(dt_200[, str_subset(string = colnames(dt_200), pattern = "y")]))$total
```

```
##  raw_alpha std.alpha  G6(smc) average_r      S/N  median_r
##  0.5365776 0.5365776 0.4427281 0.2784747 1.157858 0.3205137
```

```r
psych::alpha(cor(dt_100000[, str_subset(string = colnames(dt_100000), pattern = "y")]))$total
```

```
##  raw_alpha std.alpha  G6(smc) average_r      S/N  median_r
##  0.5972012 0.5972012 0.4971001 0.3307499 1.482629 0.3324383
```

```r
psych::omega(m = dt_100000[, str_subset(string = colnames(dt_100000), pattern = "y")], nfactors = 1, pl
```

```
## Loading required namespace: GPArotation

## Omega_h for 1 factor is not meaningful, just omega_t

## Warning in schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
## Omega_h and Omega_asymptotic are not meaningful with one factor

## Warning in cov2cor(t(w) %*% r %*% w): diag(.) had 0 or NA entries; non-finite
## result is doubtful

## Omega
## Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
##     digits = digits, title = title, sl = sl, labels = labels,
##     plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
##     covar = covar)
## Alpha:                 0.6
## G.6:                   0.5
## Omega Hierarchical:    0.6
## Omega H asymptotic:    1
## Omega Total            0.6
##
## Schmid Leiman Factor loadings greater than  0.2
##       g  F1*   h2   u2 p2
## y1 0.58      0.34 0.66  1
## y2 0.57      0.33 0.67  1
## y3 0.57      0.33 0.67  1
##
## With eigenvalues of:
##    g  F1*
## 0.99 0.00
##
## general/max  Inf   max/min =   NaN
## mean percent general =  1    with sd =  0 and cv of  0
## Explained Common Variance of the general factor =  1
##
## The degrees of freedom are 0  and the fit is  0
## The number of observations was  100000  with Chi Square =  0  with prob <  NA
## The root mean square of the residuals is  0
## The df corrected root mean square of the residuals is  NA
##
## Compare this with the adequacy of just a general factor and no group factors
## The degrees of freedom for just the general factor are 0  and the fit is  0
## The number of observations was  100000  with Chi Square =  0  with prob <  NA
## The root mean square of the residuals is  0
## The df corrected root mean square of the residuals is  NA
```

```
##
## Measures of factor score adequacy
##                                                  g F1*
## Correlation of scores with factors            0.77   0
## Multiple R square of scores with factors      0.60   0
## Minimum correlation of factor score estimates 0.19  -1
##
##  Total, General and Subset omega for each subset
##                                                  g F1*
## Omega total for total scores and subscales    0.6 0.6
## Omega general for total scores and subscales  0.6 0.6
## Omega group for total scores and subscales    0.0 0.0
```

Zusammenfassung:

- die Kovarianzmatrix der Population ist unbekannt und wird durch unsere Daten geschätzt (wenn N gegen unendlich geht, können wir unseren Parameterschätzungen vertrauen -> Konzept des Standardfehlers des Mittelwerts)
- Reliabilität ist abhängig von der Anzahl der Items: unter Zunahme der Anzahl der Items bei gleicher mittlerer Interkorrelation steigt die Reliabilität
- Cronbachs Alpha sollte nur unter Bedingung eines tau-äquivalenten Messmodells und vorheriger Testungs auf Eindimensionalität verwendet werden (Sijtsma (2009)), sonst empfiehlt sich immer modell-basierte Reliabilitätsschätzer wie McDonald's Omega

developing a questionnaire scale

descreptive analysis using classical test theory

orientiert sich an Buchkapitel 7, 13 in Moosbrugger and Kelava (2020)

$y_i = \tau_i + \epsilon_i$, aus der Messfehlertheorie folgt die Definition der Reliabilität: $Rel(Y) = \frac{Var(T)}{Var(T)+Var(E)}$

$E(y_i) = E(\tau_i) + E(\epsilon_i)$

$E(y_i) = E(\tau_i) + 0$

über mehrere Items einer Skala lässt sich ein Punktschätzer für den wahren Wert τ_i wie folgt berechnen als Summenscore: $Y = \sum_{i=1}^{p} y_i$ oder besser interpretierbar als Personmittelwertmittelwert: $\bar{Y} = \frac{\sum_{i=1}^{p} y_i}{n}$! vorläufige Testwertermittlung (*Eindimensionalität, tau-äquivalenten Messmodells muss an sich gegeben sein*)

Schwierigkeitsindex

$P_i = \frac{\sum_{v=1}^{n} y_{vi}}{n * max(y_i)} * 100$

folgende Zahlen geben die Leichtigkeit des Items an:

```
datenLV[,str_subset(string = colnames(datenLV), pattern = "^SBezMs_")] <- datenLV[,str_subset(string = 
datenLV$failitem <- rbinom(n = nrow(datenLV), size = 3, prob = .95)
head(datenLV[, c(str_subset(string = colnames(datenLV), pattern = "^SBezMs_"), "failitem")])
```

```
##   SBezMs_a SBezMs_b SBezMs_c SBezMs_d failitem
## 1        3        2        3        3        2
## 2        2        2        3        3        3
## 3        3        0        3        2        3
## 4        0        3        0        0        3
## 5        3        2        0        0        3
## 6        3        3        3        3        3
```

```r
sum(datenLV$SBezMs_a, na.rm = TRUE) / (sum(!is.na(datenLV$SBezMs_a)) * max(datenLV$SBezMs_a, na.rm = TRU
```

```
## [1] 78.82201
```

```r
sum(datenLV$SBezMs_b , na.rm = TRUE) / (sum(!is.na(datenLV$SBezMs_b)) * max(datenLV$SBezMs_b, na.rm = T
```

```
## [1] 70.34925
```

```r
sum(datenLV$SBezMs_c , na.rm = TRUE) / (sum(!is.na(datenLV$SBezMs_c)) * max(datenLV$SBezMs_c, na.rm = T
```

```
## [1] 78.88418
```

```r
sum(datenLV$SBezMs_d, na.rm = TRUE) / (sum(!is.na(datenLV$SBezMs_d)) * max(datenLV$SBezMs_d, na.rm = TRU
```

```
## [1] 83.55101
```

```r
sum(datenLV$failitem, na.rm = TRUE) / (sum(!is.na(datenLV$failitem)) * max(datenLV$failitem, na.rm = TRU
```

```
## [1] 95.50749
```

```r
datenLV[,str_subset(string = colnames(datenLV), pattern = "^SBezMs_")] <- datenLV[,str_subset(string = 
```

Itemvarianz

$Var(y_i) = \frac{\sum_{v=1}^{n}(y_{vi} - \bar{y_i})^2}{n}$

```r
sum((datenLV$SBezMs_a - mean(datenLV$SBezMs_a, na.rm = TRUE))^2, na.rm = TRUE) / sum(!is.na(datenLV$SBez
```

```
## [1] 0.5393213
```

```r
sum((datenLV$SBezMs_b - mean(datenLV$SBezMs_b, na.rm = TRUE))^2, na.rm = TRUE) / sum(!is.na(datenLV$SBez
```

```
## [1] 0.8131689
```

```r
sum((datenLV$SBezMs_c - mean(datenLV$SBezMs_c, na.rm = TRUE))^2, na.rm = TRUE) / sum(!is.na(datenLV$SBez
```

```
## [1] 0.8543597
```

```r
sum((datenLV$SBezMs_d - mean(datenLV$SBezMs_d, na.rm = TRUE))^2, na.rm = TRUE) / sum(!is.na(datenLV$SBez
```

```
## [1] 0.6580054
```

```r
sum((datenLV$failitem - mean(datenLV$failitem, na.rm = TRUE))^2, na.rm = TRUE) / sum(!is.na(datenLV$fail
```

```
## [1] 0.1325844
```

Trennschärfe

part-whole korrigierte Trennschärfe $r_{it(i)}$: $r_{it(i)} = r_{(y_i, y(i))}$

```r
cor(datenLV$SBezMs_a, rowSums(datenLV[, c("SBezMs_b", "SBezMs_c", "SBezMs_d")], na.rm = TRUE), use = "co
```

```
## [1] 0.5407678
```

```r
cor(datenLV$SBezMs_b, rowSums(datenLV[, c("SBezMs_a", "SBezMs_c", "SBezMs_d")], na.rm = TRUE), use = "co
```

```
## [1] 0.4083451
```

```r
cor(datenLV$SBezMs_c, rowSums(datenLV[, c("SBezMs_a", "SBezMs_b", "SBezMs_d")], na.rm = TRUE), use = "co
```

```
## [1] 0.4591703
```

```r
cor(datenLV$SBezMs_d, rowSums(datenLV[, c("SBezMs_a", "SBezMs_b", "SBezMs_c")], na.rm = TRUE), use = "co
```

```
## [1] 0.3990633
```

```r
cor(datenLV$failitem, rowSums(datenLV[, c("SBezMs_a", "SBezMs_b", "SBezMs_c", "SBezMs_d")], na.rm = TRUE
```

```
## [1] 0.006453734
```

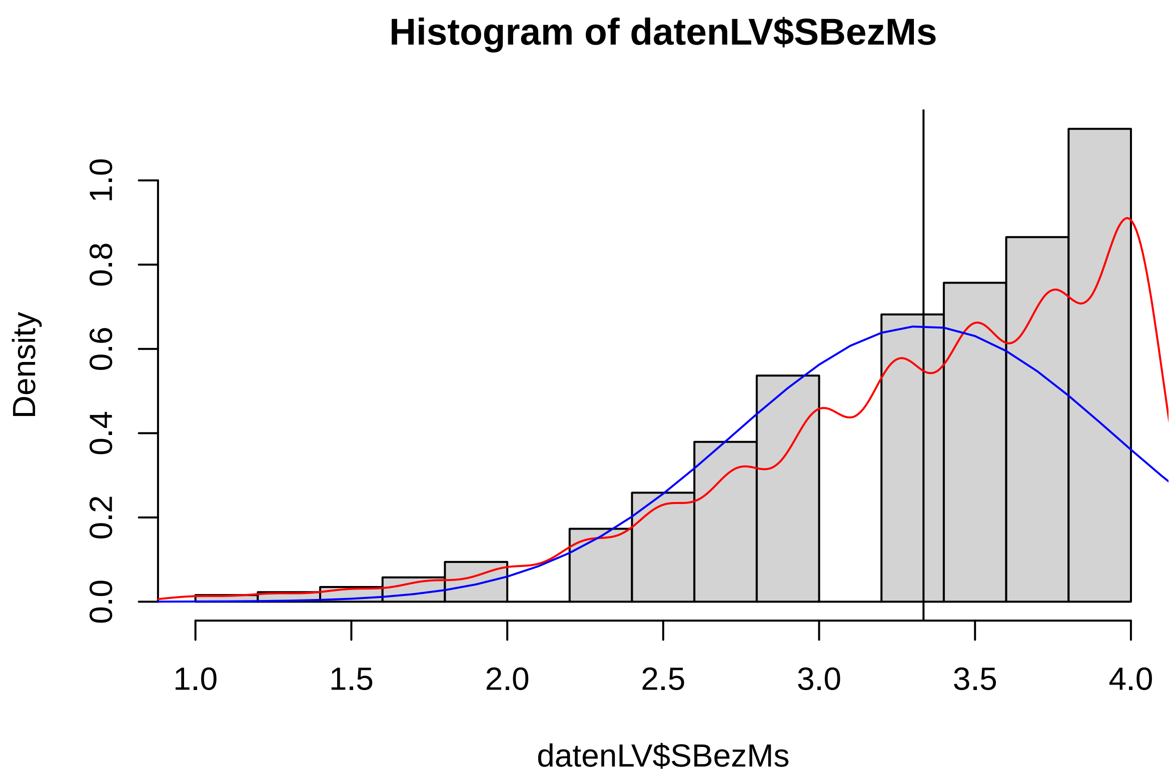
Testwertverteilung

orientiert sich an Buchkapitel 8 in Moosbrugger and Kelava (2020)

```r
## liegt bereits in Daten vor
cor(rowMeans(x = datenLV[,str_subset(string = colnames(datenLV), pattern = "^SBezMs_")]), datenLV$SBezMs
```

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

```r
hist(datenLV$SBezMs, freq = FALSE)
abline(v = mean(datenLV$SBezMs, na.rm = TRUE))
lines(density(datenLV$SBezMs[!is.na(datenLV$SBezMs)]), col="red") # empirical density
lines(seq(0, 5, by=.1), dnorm(seq(0, 5, by=.1),
      mean(datenLV$SBezMs, na.rm = TRUE), sd(datenLV$SBezMs, na.rm = TRUE)), col="blue") # normal densi
```




```r
sd(x = datenLV$SBezMs, na.rm = TRUE)
```

```
## [1] 0.6099304
```

```r
moments::skewness(x = datenLV$SBezMs, na.rm = TRUE)
```

```
## [1] -1.013992
```

```r
moments::kurtosis(x = datenLV$SBezMs, na.rm = TRUE) - 3 # = SPSS output
```

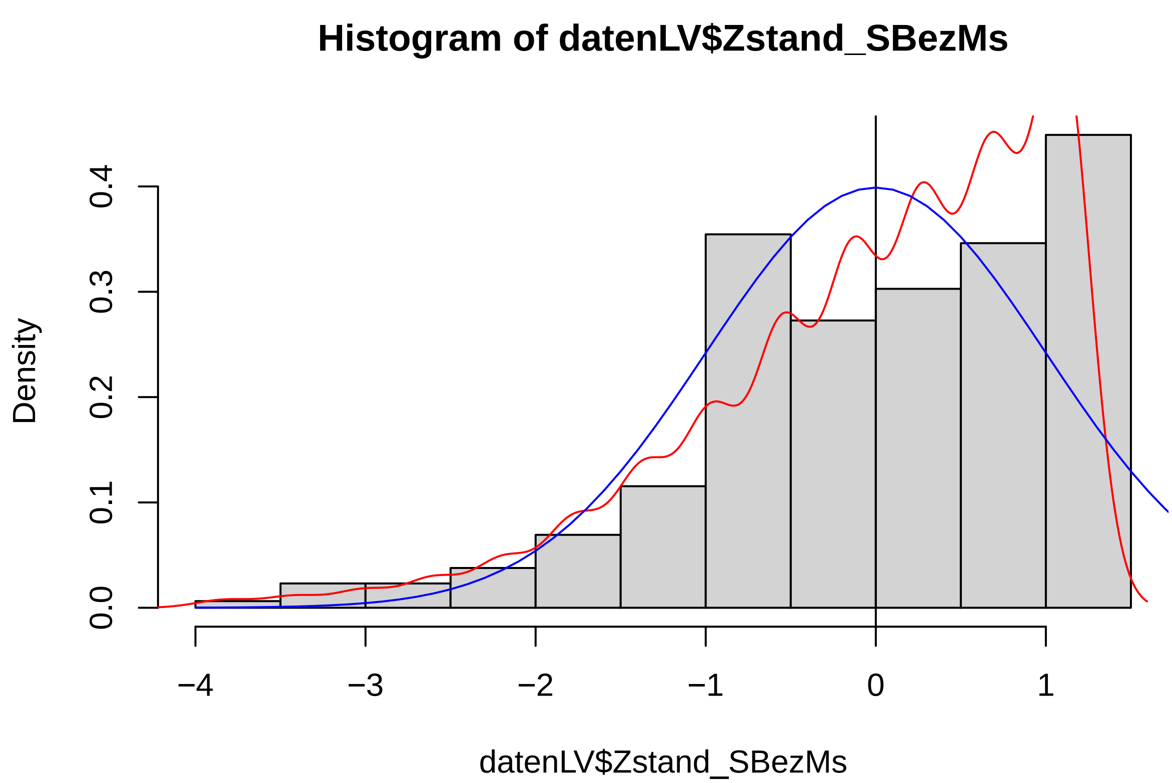
```
## [1] 0.7847189
```

```r
shapiro.test(x = datenLV$SBezMs)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datenLV$SBezMs
## W = 0.89501, p-value < 2.2e-16
```

für ein normorientierten Vergleich bietet sich eine z-Standardisierung $\frac{Y_v - \bar{Y}}{SD(Y)}$ an:

```r
datenLV$Zstand_SBezMs <- scale(x = datenLV$SBezMs, center = TRUE, scale = TRUE)
hist(datenLV$Zstand_SBezMs, freq = FALSE)
abline(v = mean(datenLV$Zstand_SBezMs, na.rm = TRUE))
lines(density(datenLV$Zstand_SBezMs[!is.na(datenLV$Zstand_SBezMs)]), col="red")
lines(seq(-4, 4, by=.1), dnorm(seq(-4, 4, by=.1),
      mean(datenLV$Zstand_SBezMs, na.rm = TRUE), sd(datenLV$Zstand_SBezMs, na.rm = TRUE)), col="blue")
```



exploratory factor analysis

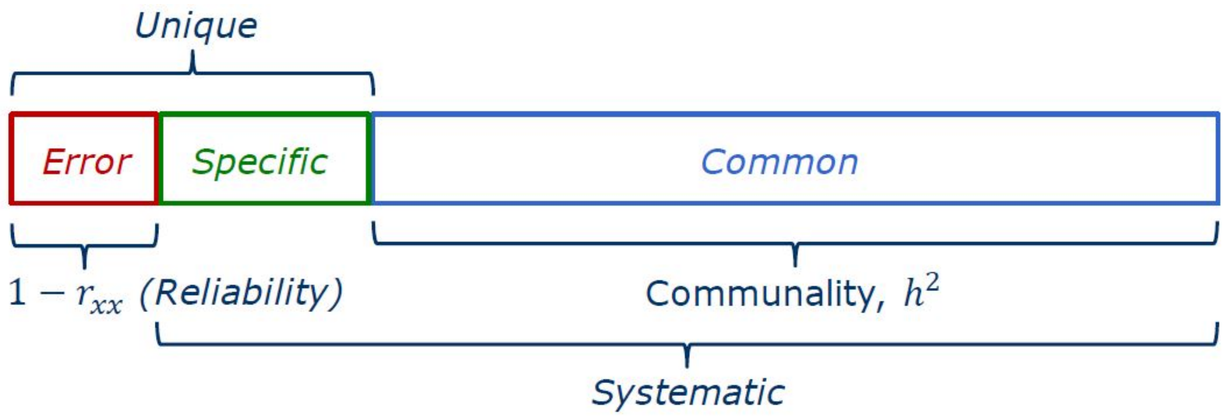
induktive Methode

an sich gehört zur KTT Testung auf Messinvarianz über die klassischen Testmodelle, jedoch muss für diese Eindimensionaltität gegeben sein, hierfür eignet sich eine sogenannte EFA

Verwendung des `psych` Paketes in R (siehe http://personality-project.org/r/psych/HowTo/factor.pdf), Alternativ eignet sich auch das Statistikprogramm JASP für EFA / CFA: (https://jasp-stats.org/)

einführende Artikel in EFA: Costello and Osborne (2005), Mvududu and Sink (2013) (Anmerkung: es gibt Mischformen zwischen EFA und CFA, wie beispielsweise ESEM: Marsh et al. (2014))

! wichtig es sollte keine principal component analysis gerechnet werden (Relikt der Vergangenheit, Grundprinzipien mit EFA gleich), da hier keine **Varianzzerlegung** stattfindet.



Ziele der explorativen Faktorenanalyse sind

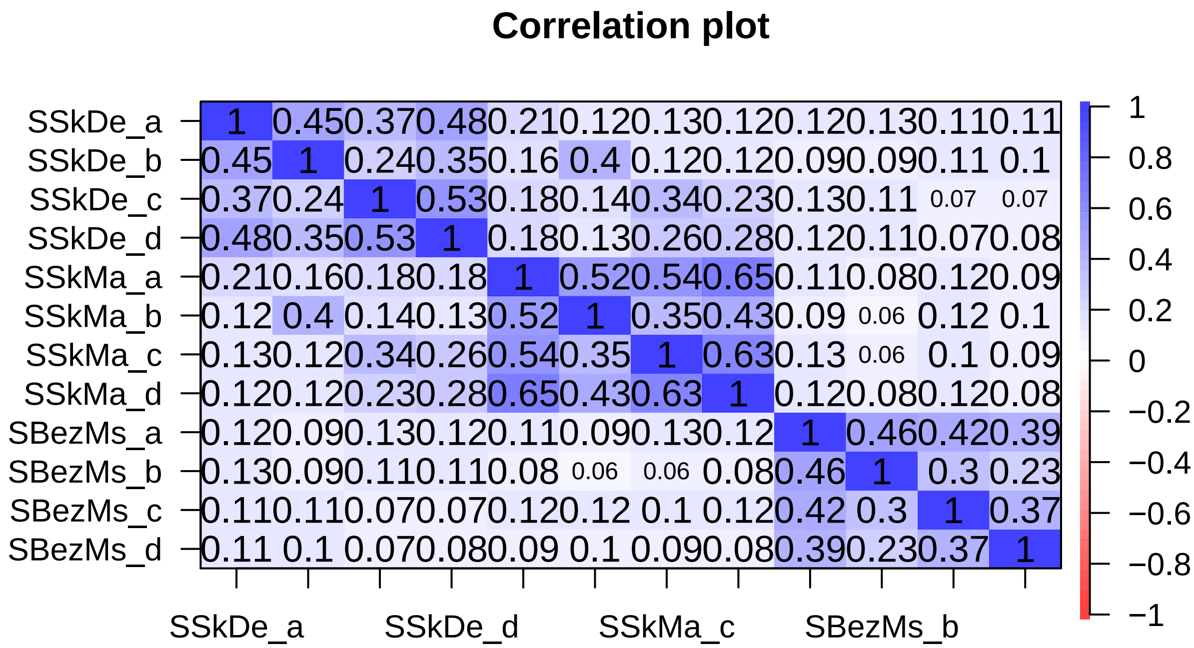
- die Reduktion der Dimension der Kovarianz- bzw. Korrelationsmatrix
- die Identifizierung von latenten Variablen (z.B. über die Hauptachsenanalyse (Principal Axes Analyses)) und
- die Ausdifferenzierung eines komplexen Merkmalsbereichs in homogene Teilbereiche, d.h. die Variablen werden so gruppiert, dass sie innerhalb der Gruppe möglichst hoch korreliert sind (homogen) und die Gruppen der Variablen zueinander möglichst heterogen sind. Hier wird das gleiche Ziel wie mit einer Clusteranalyse (Latente Klassenanalyse) verfolgt

EFA läuft in vier Schritten ab:

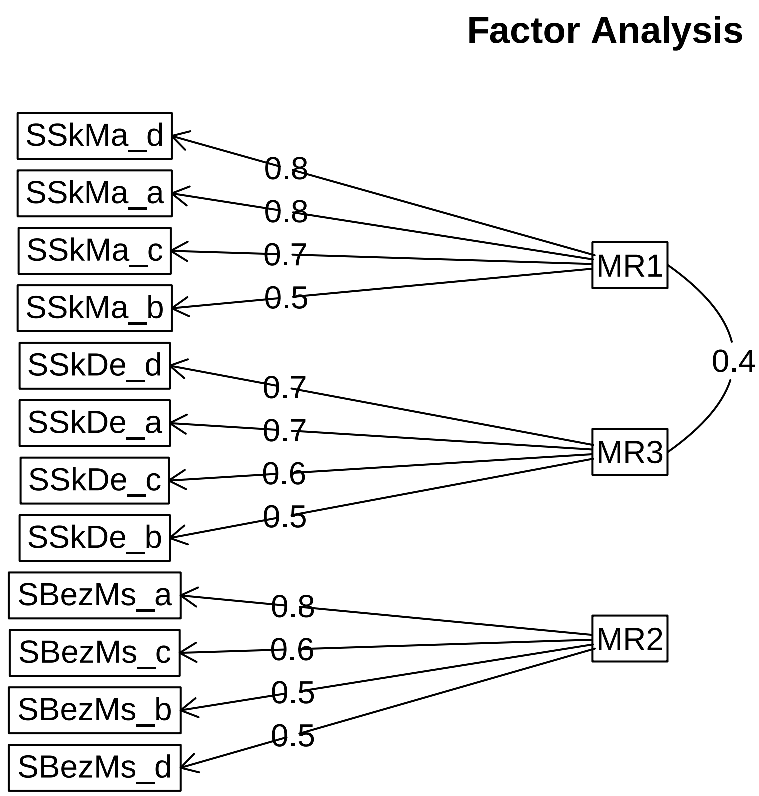
1. vorbetrachtende Tests
2. die Wahl der Extraktionsmethode
3. die Wahl eines Abbruchkriteriums und zuletzt
4. die Wahl der Rotationsmethode

zur eigenen Interpretation der Ergebnisse siehe Blog von Michael Clark: https://m-clark.github.io/posts/2020-04-10-psych-explained/

```
psych::corPlot(r = datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBezMs_")
```




```r
## not accounting for the non-normal / skewed data
efa1 = fa(r = datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBezMs_")], n
fa.diagram(efa1)
```




```
efa1
```

```
## Factor Analysis using method =  minres
## Call: fa(r = datenLV[, str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBezMs_")]
##     nfactors = 3, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            MR1   MR3   MR2   h2   u2 com
## SSkDe_a  -0.07  0.70  0.04 0.47 0.53   1
## SSkDe_b   0.04  0.51  0.03 0.28 0.72   1
## SSkDe_c   0.09  0.57 -0.01 0.37 0.63   1
## SSkDe_d   0.02  0.75 -0.03 0.56 0.44   1
## SSkMa_a   0.80 -0.02  0.01 0.63 0.37   1
## SSkMa_b   0.53  0.07  0.03 0.32 0.68   1
## SSkMa_c   0.69  0.06  0.00 0.51 0.49   1
## SSkMa_d   0.84 -0.03 -0.01 0.70 0.30   1
## SBezMs_a  0.00 -0.01  0.76 0.57 0.43   1
## SBezMs_b -0.04  0.05  0.53 0.29 0.71   1
## SBezMs_c  0.04 -0.02  0.59 0.35 0.65   1
## SBezMs_d  0.01  0.00  0.53 0.28 0.72   1
##
##                        MR1  MR3  MR2
## SS loadings           2.16 1.68 1.50
## Proportion Var        0.18 0.14 0.13
## Cumulative Var        0.18 0.32 0.44
## Proportion Explained  0.40 0.31 0.28
## Cumulative Proportion 0.40 0.72 1.00
##
```

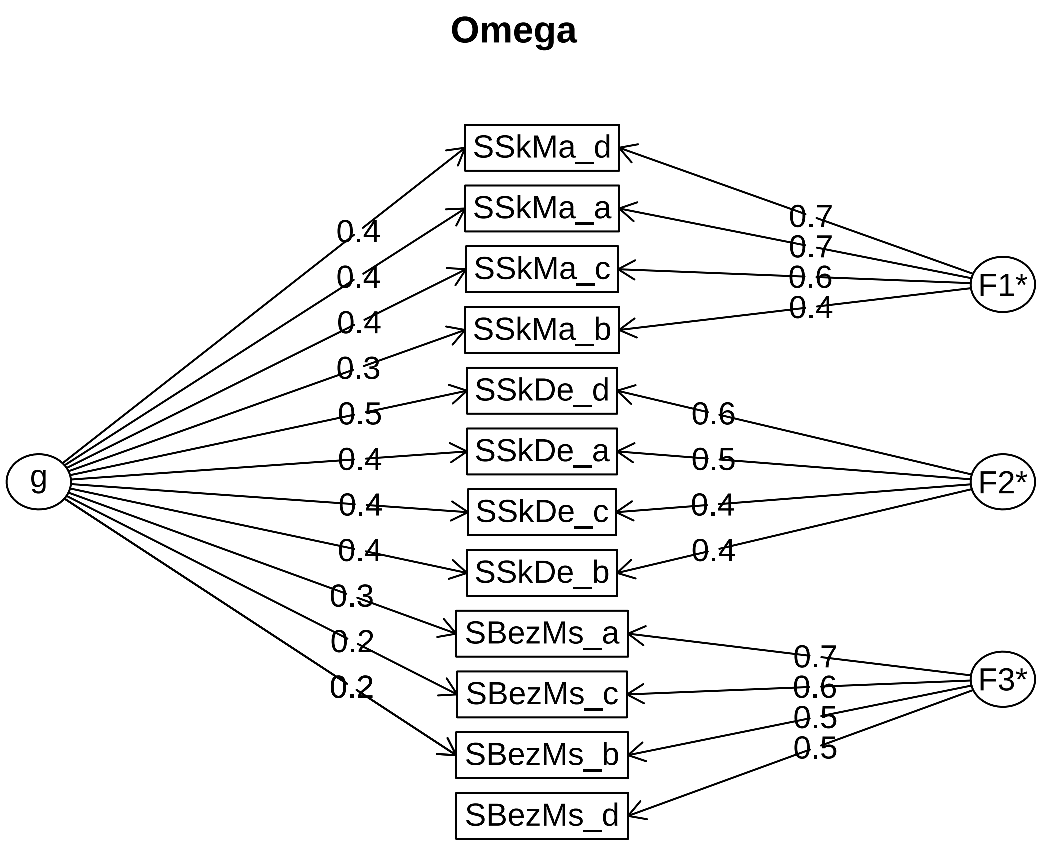
```
##  With factor correlations of
##      MR1  MR3  MR2
## MR1 1.00 0.36 0.20
## MR3 0.36 1.00 0.24
## MR2 0.20 0.24 1.00
##
## Mean item complexity =  1
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  66  and the objective function was  3.47 with Chi Squa
## The degrees of freedom for the model are 33  and the objective function was  0.43
##
## The root mean square of the residuals (RMSR) is  0.05
## The df corrected root mean square of the residuals is  0.07
##
## The harmonic number of observations is  2811 with the empirical chi square  868.15  with prob <  4.6
## The total number of observations was  3005  with Likelihood Chi Square =  1288.18  with prob <  1.3e
##
## Tucker Lewis Index of factoring reliability =  0.757
## RMSEA index =  0.113  and the 90 % confidence intervals are  0.107 0.118
## BIC =  1023.92
## Fit based upon off diagonal values = 0.97
## Measures of factor score adequacy
##                                                    MR1  MR3  MR2
## Correlation of (regression) scores with factors   0.92 0.88 0.86
## Multiple R square of scores with factors          0.85 0.77 0.73
## Minimum correlation of possible factor scores     0.70 0.53 0.47
```

```
### accounting partly for the non-normal / skewed data using choric correlations (limited information a
efa2choric <- fa(r = datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBezMs
efa2choric
```

```
## Factor Analysis using method =  wls
## Call: fa(r = datenLV[, str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBezMs_")]
##     nfactors = 3, rotate = "oblimin", scores = "Bartlett", max.iter = 500,
##     fm = "wls", cor = "poly")
## Standardized loadings (pattern matrix) based upon correlation matrix
##           WLS1  WLS3  WLS2   h2   u2 com
## SSkDe_a  -0.09  0.76  0.04 0.55 0.45 1.0
## SSkDe_b   0.04  0.57  0.03 0.36 0.64 1.0
## SSkDe_c   0.10  0.66 -0.01 0.49 0.51 1.1
## SSkDe_d   0.03  0.82 -0.03 0.67 0.33 1.0
## SSkMa_a   0.86 -0.03  0.01 0.72 0.28 1.0
## SSkMa_b   0.62  0.06  0.02 0.43 0.57 1.0
## SSkMa_c   0.77  0.07  0.00 0.64 0.36 1.0
## SSkMa_d   0.90 -0.02  0.00 0.80 0.20 1.0
## SBezMs_a  0.00 -0.01  0.83 0.68 0.32 1.0
## SBezMs_b -0.05  0.05  0.59 0.36 0.64 1.0
## SBezMs_c  0.04 -0.02  0.69 0.48 0.52 1.0
## SBezMs_d  0.00  0.01  0.63 0.40 0.60 1.0
##
##                       WLS1 WLS3 WLS2
## SS loadings           2.60 2.06 1.92
## Proportion Var        0.22 0.17 0.16
## Cumulative Var        0.22 0.39 0.55
```

```
## Proportion Explained  0.39 0.31 0.29
## Cumulative Proportion 0.39 0.71 1.00
##
##  With factor correlations of
##      WLS1 WLS3 WLS2
## WLS1 1.00 0.41 0.23
## WLS3 0.41 1.00 0.27
## WLS2 0.23 0.27 1.00
##
## Mean item complexity =  1
## Test of the hypothesis that 3 factors are sufficient.
##
## The degrees of freedom for the null model are  66  and the objective function was  5.63 with Chi Squa
## The degrees of freedom for the model are 33  and the objective function was  0.89
##
## The root mean square of the residuals (RMSR) is  0.06
## The df corrected root mean square of the residuals is  0.08
##
## The harmonic number of observations is  2811 with the empirical chi square  1148.38  with prob <  5e
## The total number of observations was  3005  with Likelihood Chi Square =  2678.78  with prob <  0
##
## Tucker Lewis Index of factoring reliability =  0.685
## RMSEA index =  0.163  and the 90 % confidence intervals are  0.158 0.169
## BIC =  2414.51
## Fit based upon off diagonal values = 0.97
## Measures of factor score adequacy
##                                                   WLS1 WLS3 WLS2
## Correlation of (regression) scores with factors   0.95 0.91 0.90
## Multiple R square of scores with factors          0.91 0.84 0.81
## Minimum correlation of possible factor scores     0.81 0.67 0.62
```

```
### model based reliability score
omega(datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBezMs_")])
```

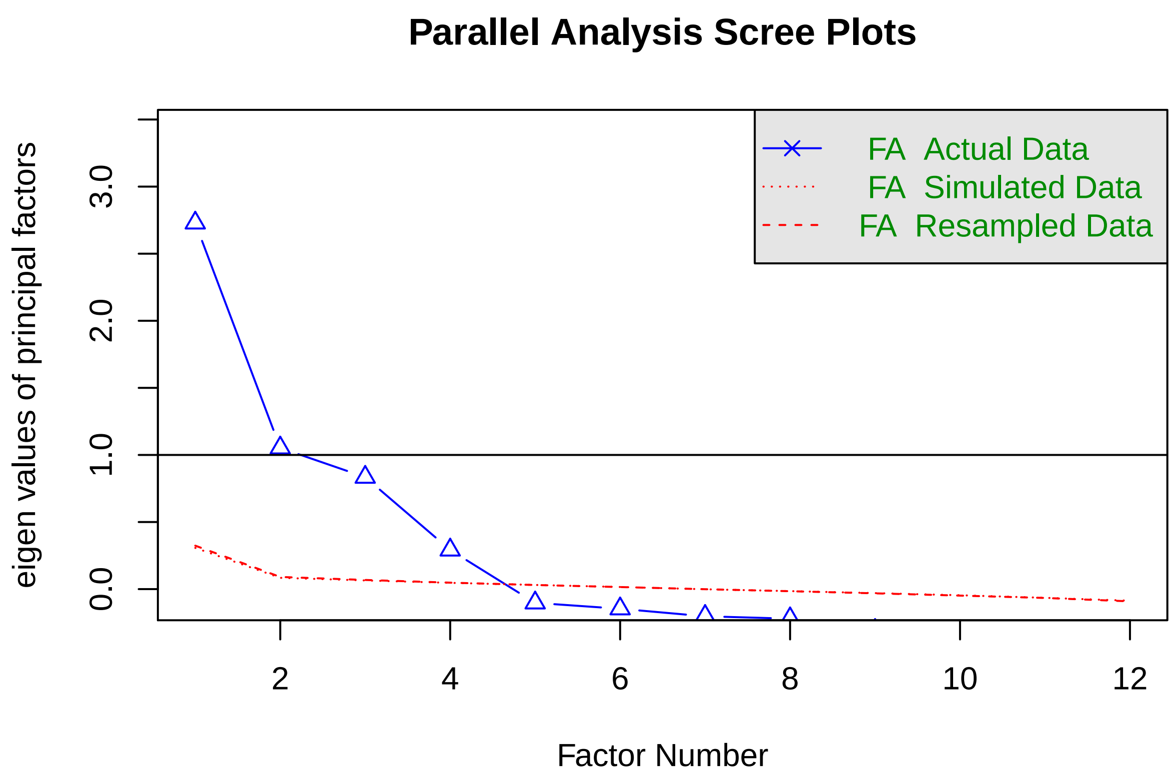



```
## Omega
## Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
##     digits = digits, title = title, sl = sl, labels = labels,
##     plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
##     covar = covar)
## Alpha:                 0.76
## G.6:                   0.81
## Omega Hierarchical:    0.45
## Omega H asymptotic:    0.54
## Omega Total            0.83
##
## Schmid Leiman Factor loadings greater than  0.2
##             g   F1*   F2*   F3*   h2   u2   p2
## SSkDe_a  0.43        0.53       0.47 0.53 0.40
## SSkDe_b  0.37        0.38       0.28 0.72 0.48
## SSkDe_c  0.42        0.43       0.37 0.63 0.48
## SSkDe_d  0.49        0.56       0.56 0.44 0.43
## SSkMa_a  0.43  0.67             0.63 0.37 0.29
## SSkMa_b  0.35  0.44             0.32 0.68 0.38
## SSkMa_c  0.42  0.57             0.51 0.49 0.35
## SSkMa_d  0.45  0.70             0.70 0.30 0.28
## SBezMs_a 0.27              0.71 0.57 0.43 0.13
## SBezMs_b 0.20              0.50 0.29 0.71 0.14
## SBezMs_c 0.22              0.55 0.35 0.65 0.14
## SBezMs_d 0.20              0.49 0.28 0.72 0.14
##
```

```
## With eigenvalues of:
##    g  F1*  F2*  F3*
## 1.63 1.48 0.93 1.30
##
## general/max  1.1   max/min =   1.59
## mean percent general =  0.3    with sd =  0.14 and cv of  0.45
## Explained Common Variance of the general factor =  0.31
##
## The degrees of freedom are 33  and the fit is  0.43
## The number of observations was  3005  with Chi Square =  1288.18  with prob <  1.3e-249
## The root mean square of the residuals is  0.05
## The df corrected root mean square of the residuals is  0.07
## RMSEA index =  0.113  and the 10 % confidence intervals are  0.107 0.118
## BIC =  1023.92
##
## Compare this with the adequacy of just a general factor and no group factors
## The degrees of freedom for just the general factor are 54  and the fit is  2.02
## The number of observations was  3005  with Chi Square =  6062.27  with prob <  0
## The root mean square of the residuals is  0.17
## The df corrected root mean square of the residuals is  0.19
##
## RMSEA index =  0.192  and the 10 % confidence intervals are  0.188 0.197
## BIC =  5629.84
##
## Measures of factor score adequacy
##                                                    g  F1*   F2*  F3*
## Correlation of scores with factors             0.70 0.81  0.69 0.81
## Multiple R square of scores with factors       0.49 0.65  0.48 0.66
## Minimum correlation of factor score estimates -0.03 0.30 -0.05 0.32
##
##  Total, General and Subset omega for each subset
##                                                  g  F1*  F2*  F3*
## Omega total for total scores and subscales    0.83 0.82 0.74 0.70
## Omega general for total scores and subscales  0.45 0.26 0.33 0.10
## Omega group for total scores and subscales    0.36 0.56 0.41 0.61
```

Wenn die Anzahl der zu bestimmenden Faktoren unklar ist bietet sich die Verwendung von Scree plots an:

```
efa3 <- fa.parallel(x = datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_|^SSkDe_|^SBe
```




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

```
efa3
```

```
## Call: fa.parallel(x = datenLV[, str_subset(string = colnames(datenLV),
##     pattern = "^SSkMa_|^SSkDe_|^SBezMs_")], fa = "fa", n.iter = 50)
## Parallel analysis suggests that the number of factors =  4  and the number of components =  NA
##
##  Eigen Values of
##
##  eigen values of factors
##  [1]  2.73  1.05  0.83  0.29 -0.10 -0.15 -0.20 -0.22 -0.31 -0.36 -0.42 -0.42
##
##  eigen values of simulated factors
##  [1]  0.31  0.08  0.06  0.05  0.03  0.02  0.00 -0.01 -0.03 -0.05 -0.06 -0.09
##
##  eigen values of components
##  [1] 3.43 1.89 1.60 1.04 0.78 0.62 0.61 0.49 0.47 0.45 0.33 0.28
##
##  eigen values of simulated components
## [1] NA
```

classical test models / measurment invariance

Im Folgenden wollen wir die Items zu dem Matheselbstkonzept genauer analysieren - ohne die Testung von tau-äquivalentem Modell, sowie Eindimensionalität berechnen wir vorläufig nur McDonald’s Omega:

```r
psych::omega(datenLV[,str_subset(string = colnames(datenLV), pattern = "^SSkMa_")], nfactors = 1)
```

```
## Omega_h for 1 factor is not meaningful, just omega_t

## Warning in schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
## Omega_h and Omega_asymptotic are not meaningful with one factor

## Omega
## Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
##     digits = digits, title = title, sl = sl, labels = labels,
##     plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
##     covar = covar)
## Alpha:                 0.81
## G.6:                   0.78
## Omega Hierarchical:    0.82
## Omega H asymptotic:    1
## Omega Total            0.82
##
## Schmid Leiman Factor loadings greater than  0.2
##            g  F1*   h2   u2 p2
## SSkMa_a 0.81      0.65 0.35  1
## SSkMa_b 0.56      0.31 0.69  1
## SSkMa_c 0.70      0.49 0.51  1
## SSkMa_d 0.83      0.69 0.31  1
##
## With eigenvalues of:
##   g F1*
## 2.1 0.0
##
## general/max  3.862933e+16   max/min =   1
## mean percent general =  1    with sd =  0 and cv of  0
## Explained Common Variance of the general factor =  1
##
## The degrees of freedom are 2  and the fit is  0.05
## The number of observations was  3005  with Chi Square =  143.45  with prob <  7.1e-32
## The root mean square of the residuals is  0.04
## The df corrected root mean square of the residuals is  0.07
## RMSEA index =  0.153  and the 10 % confidence intervals are  0.133 0.175
## BIC =  127.43
##
## Compare this with the adequacy of just a general factor and no group factors
## The degrees of freedom for just the general factor are 2  and the fit is  0.05
## The number of observations was  3005  with Chi Square =  143.45  with prob <  7.1e-32
## The root mean square of the residuals is  0.04
## The df corrected root mean square of the residuals is  0.07
##
## RMSEA index =  0.153  and the 10 % confidence intervals are  0.133 0.175
## BIC =  127.43
##
## Measures of factor score adequacy
##                                                  g F1*
## Correlation of scores with factors            0.92   0
## Multiple R square of scores with factors      0.85   0
## Minimum correlation of factor score estimates 0.69  -1
##
```

```
##  Total, General and Subset omega for each subset
##                                                  g  F1*
## Omega total for total scores and subscales    0.82 0.82
## Omega general for total scores and subscales  0.82 0.82
## Omega group for total scores and subscales    0.00 0.00
```

Um auf Messinvarianz zu testen, müssen wir das Messmodell über eine sogenannte Modellsyntax eingeben, um darauf folgend das R Paket lavaan verwenden zu können:

Achtung: CFAs werden geschätzt mittels maximum likelihood (ML), weiter unten in Abschnitt CFA / SEM für die Daten besser geeignete Schätzmethode (jedoch ist ML hier zielführend da hiermit über den likelihood ratio test ein Modellvergleich gerechnet werden kann):

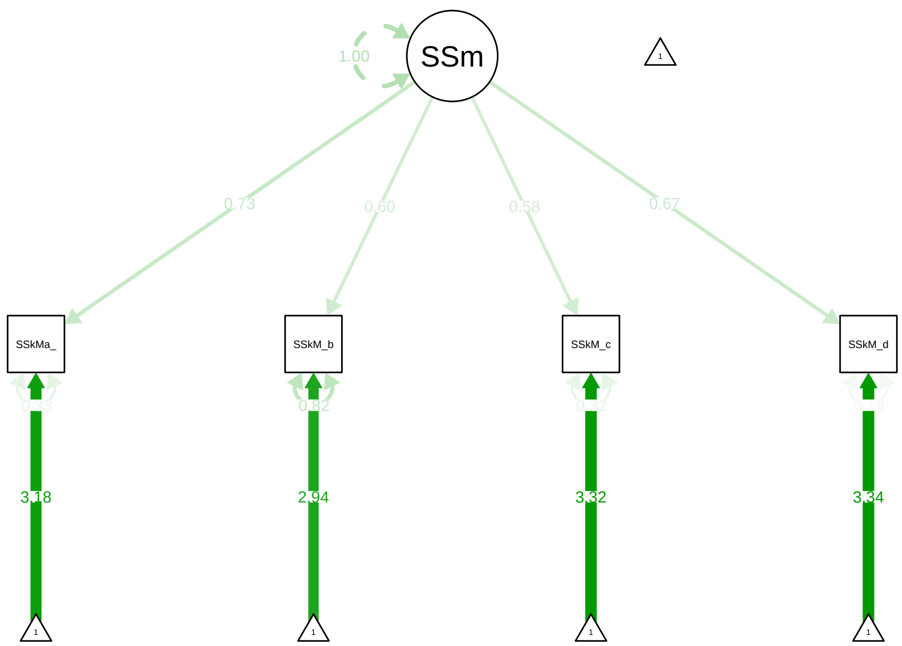
classical test models (Voraussetzung Reliabilitätsanalysen, Sparsamkeit des Modells; gleiches Prinzip wie Messinvarianz weiter unten)

```
cong.model <- '
SSmath =~ lam1*SSkMa_a + lam2*SSkMa_b + lam3*SSkMa_c + lam4*SSkMa_d

SSkMa_a ~~ var1*SSkMa_a
SSkMa_b ~~ var2*SSkMa_b
SSkMa_c ~~ var3*SSkMa_c
SSkMa_d ~~ var4*SSkMa_d

SSkMa_a ~ mean1*1
SSkMa_b ~ mean2*1
SSkMa_c ~ mean3*1
SSkMa_d ~ mean4*1
'

# identification: Fixed factor
cong.fit <-sem(cong.model, data = datenLV, std.lv = TRUE)
summary(cong.fit, standardized = TRUE, fit.measures=TRUE)
```

```
## lavaan 0.6-8 ended normally after 16 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        12
##
##                                                   Used       Total
##   Number of observations                          2843        3005
##
## Model Test User Model:
##
##   Test statistic                               124.430
##   Degrees of freedom                                 2
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              4111.594
##   Degrees of freedom                                 6
##   P-value                                        0.000
##
## User Model versus Baseline Model:
```

```
##
##   Comparative Fit Index (CFI)                    0.970
##   Tucker-Lewis Index (TLI)                       0.911
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -12967.449
##   Loglikelihood unrestricted model (H1)     -12905.234
##
##   Akaike (AIC)                               25958.898
##   Bayesian (BIC)                             26030.329
##   Sample-size adjusted Bayesian (BIC)        25992.201
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.147
##   90 Percent confidence interval - lower         0.125
##   90 Percent confidence interval - upper         0.169
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.028
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath =~
##     SSkMa_a (lam1)    0.733    0.016   45.845    0.000    0.733    0.786
##     SSkMa_b (lam2)    0.603    0.020   29.672    0.000    0.603    0.554
##     SSkMa_c (lam3)    0.582    0.014   40.830    0.000    0.582    0.717
##     SSkMa_d (lam4)    0.674    0.013   49.973    0.000    0.674    0.841
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a (men1)    3.177    0.017  181.586    0.000    3.177    3.406
##    .SSkMa_b (men2)    2.943    0.020  144.193    0.000    2.943    2.704
##    .SSkMa_c (men3)    3.319    0.015  217.876    0.000    3.319    4.086
##    .SSkMa_d (men4)    3.337    0.015  221.820    0.000    3.337    4.160
##     SSmath            0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a (var1)    0.332    0.013   25.253    0.000    0.332    0.382
##    .SSkMa_b (var2)    0.820    0.024   34.594    0.000    0.820    0.693
##    .SSkMa_c (var3)    0.320    0.011   29.946    0.000    0.320    0.486
##    .SSkMa_d (var4)    0.189    0.009   19.857    0.000    0.189    0.293
##     SSmath            1.000                               1.000    1.000
```

```r
semPlot::semPaths(object = cong.fit, what = "est")
```




```r
tauequi.model <- '
SSmath =~ lam1*SSkMa_a + lam2*SSkMa_b + lam3*SSkMa_c + lam4*SSkMa_d

SSkMa_a ~~ var1*SSkMa_a
SSkMa_b ~~ var2*SSkMa_b
SSkMa_c ~~ var3*SSkMa_c
SSkMa_d ~~ var4*SSkMa_d

SSkMa_a ~ mean1*1
SSkMa_b ~ mean2*1
SSkMa_c ~ mean3*1
SSkMa_d ~ mean4*1

# fix variance of SSmath factor
SSmath ~~ 1*SSmath

# constraints
lam1 == lam2
lam2 == lam3
lam3 == lam4
'

# identification: Fixed factor
tauequi.fit <-sem(tauequi.model, data = datenLV, std.lv = TRUE)
```

```r
summary(tauequi.fit, standardized = TRUE, fit.measures=TRUE)
```

```
## lavaan 0.6-8 ended normally after 12 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        12
##   Number of equality constraints                     3
##
##                                                   Used       Total
##   Number of observations                          2843        3005
##
## Model Test User Model:
##
##   Test statistic                               209.808
##   Degrees of freedom                                 5
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              4111.594
##   Degrees of freedom                                 6
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.950
##   Tucker-Lewis Index (TLI)                       0.940
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -13010.138
##   Loglikelihood unrestricted model (H1)     -12905.234
##
##   Akaike (AIC)                               26038.276
##   Bayesian (BIC)                             26091.849
##   Sample-size adjusted Bayesian (BIC)        26063.253
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.120
##   90 Percent confidence interval - lower         0.106
##   90 Percent confidence interval - upper         0.134
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.065
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
```

```
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath =~
##     SSkMa_a (lam1)    0.655    0.010   63.090    0.000    0.655    0.734
##     SSkMa_b (lam2)    0.655    0.010   63.090    0.000    0.655    0.586
##     SSkMa_c (lam3)    0.655    0.010   63.090    0.000    0.655    0.767
##     SSkMa_d (lam4)    0.655    0.010   63.090    0.000    0.655    0.830
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a (men1)    3.177    0.017  189.961    0.000    3.177    3.563
##    .SSkMa_b (men2)    2.943    0.021  140.437    0.000    2.943    2.634
##    .SSkMa_c (men3)    3.319    0.016  207.225    0.000    3.319    3.886
##    .SSkMa_d (men4)    3.337    0.015  225.422    0.000    3.337    4.228
##     SSmath            0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a (var1)    0.366    0.012   30.452    0.000    0.366    0.461
##    .SSkMa_b (var2)    0.820    0.024   34.536    0.000    0.820    0.657
##    .SSkMa_c (var3)    0.300    0.010   28.781    0.000    0.300    0.412
##    .SSkMa_d (var4)    0.194    0.008   23.975    0.000    0.194    0.312
##     SSmath            1.000                               1.000    1.000
##
## Constraints:
##                                                |Slack|
##     lam1 - (lam2)                                0.000
##     lam2 - (lam3)                                0.000
##     lam3 - (lam4)                                0.000
```

```
anova(cong.fit, tauequi.fit) # LRT
```

```
## Chi-Squared Difference Test
##
##             Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## cong.fit     2 25959 26030 124.43
## tauequi.fit  5 26038 26092 209.81     85.378       3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit.stats <- rbind(fitmeasures(cong.fit, fit.measures = c("chisq", "df", "rmsea", "tli", "cfi", "aic"))
fitmeasures(tauequi.fit, fit.measures = c("chisq", "df", "rmsea", "tli", "cfi", "aic")))
rownames(fit.stats) <- c("configural", "weak invariance")
fit.stats
```

```
##                    chisq df     rmsea       tli       cfi      aic
## configural      124.4304  2 0.1467375 0.9105389 0.9701796 25958.90
## weak invariance 209.8084  5 0.1200330 0.9401377 0.9501148 26038.28
```

```
parallel.model <- '
SSmath =~ lam1*SSkMa_a + lam2*SSkMa_b + lam3*SSkMa_c + lam4*SSkMa_d

SSkMa_a ~~ var1*SSkMa_a
SSkMa_b ~~ var2*SSkMa_b
SSkMa_c ~~ var3*SSkMa_c
```

```
SSkMa_d ~~ var4*SSkMa_d

SSkMa_a ~ mean1*1
SSkMa_b ~ mean2*1
SSkMa_c ~ mean3*1
SSkMa_d ~ mean4*1

# fix variance of SSmath factor
SSmath ~~ 1*SSmath

# constraints
lam1 == lam2
lam2 == lam3
lam3 == lam4

var1 == var2
var2 == var3
var3 == var4
'

# identification: Fixed factor
parallel.fit <-sem(parallel.model, data = datenLV, std.lv = TRUE)
summary(parallel.fit, standardized = TRUE, fit.measures=TRUE)
```

```
## lavaan 0.6-8 ended normally after 4 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        12
##   Number of equality constraints                     6
##
##                                                   Used       Total
##   Number of observations                          2843        3005
##
## Model Test User Model:
##
##   Test statistic                              1157.170
##   Degrees of freedom                                 8
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              4111.594
##   Degrees of freedom                                 6
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.720
##   Tucker-Lewis Index (TLI)                       0.790
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -13483.819
```

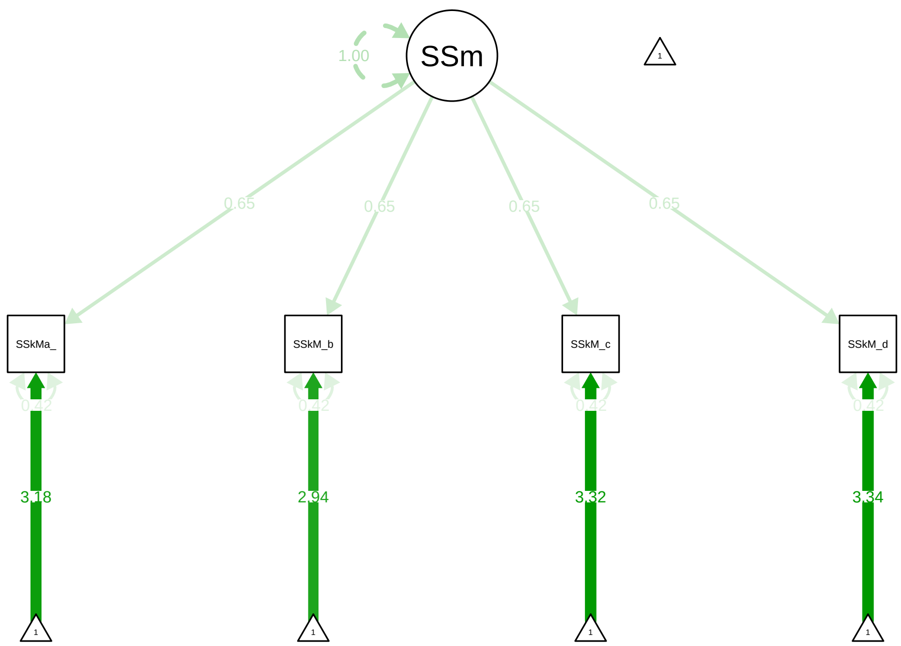
```
##   Loglikelihood unrestricted model (H1)     -12905.234
##
##   Akaike (AIC)                               26979.637
##   Bayesian (BIC)                             27015.353
##   Sample-size adjusted Bayesian (BIC)        26996.289
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.225
##   90 Percent confidence interval - lower         0.214
##   90 Percent confidence interval - upper         0.236
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.144
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath =~
##     SSkMa_a (lam1)    0.648    0.011   59.955    0.000    0.648    0.708
##     SSkMa_b (lam2)    0.648    0.011   59.955    0.000    0.648    0.708
##     SSkMa_c (lam3)    0.648    0.011   59.955    0.000    0.648    0.708
##     SSkMa_d (lam4)    0.648    0.011   59.955    0.000    0.648    0.708
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a (men1)    3.177    0.017  184.882    0.000    3.177    3.467
##    .SSkMa_b (men2)    2.943    0.017  171.290    0.000    2.943    3.213
##    .SSkMa_c (men3)    3.319    0.017  193.132    0.000    3.319    3.622
##    .SSkMa_d (men4)    3.337    0.017  194.196    0.000    3.337    3.642
##     SSmath            0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a (var1)    0.419    0.006   65.303    0.000    0.419    0.499
##    .SSkMa_b (var2)    0.419    0.006   65.303    0.000    0.419    0.499
##    .SSkMa_c (var3)    0.419    0.006   65.303    0.000    0.419    0.499
##    .SSkMa_d (var4)    0.419    0.006   65.303    0.000    0.419    0.499
##     SSmath            1.000                               1.000    1.000
##
## Constraints:
##                                                |Slack|
##     lam1 - (lam2)                                0.000
##     lam2 - (lam3)                                0.000
##     lam3 - (lam4)                                0.000
##     var1 - (var2)                                0.000
##     var2 - (var3)                                0.000
```

```
##     var3 - (var4)                                0.000
```

```r
anova(cong.fit, tauequi.fit, parallel.fit) # LRT
```

```
## Chi-Squared Difference Test
##
##              Df   AIC   BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## cong.fit      2 25959 26030  124.43
## tauequi.fit   5 26038 26092  209.81      85.38       3  < 2.2e-16 ***
## parallel.fit  8 26980 27015 1157.17     947.36       3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
semPlot::semPaths(object = parallel.fit, what = "est")
```



measurment invariance (longitudinal data, multi-group analysis)

zur eigenen Interpretation der Ergebnisse siehe: https://rstudio-pubs-static.s3.amazonaws.com/194879_192b64ad567743d392b559d650b95a3b.html

```r
CFAmodel <- ' SSmath  =~ SSkMa_a  + SSkMa_b  + SSkMa_c + SSkMa_d'
fit <- cfa(CFAmodel, data=datenLV) # ! ML
summary(fit, fit.measures=TRUE)
```

```
## lavaan 0.6-8 ended normally after 19 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         8
```

```
##
##                                                   Used       Total
##   Number of observations                          2843        3005
##
## Model Test User Model:
##
##   Test statistic                               124.430
##   Degrees of freedom                                 2
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              4111.594
##   Degrees of freedom                                 6
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.970
##   Tucker-Lewis Index (TLI)                       0.911
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -12967.449
##   Loglikelihood unrestricted model (H1)     -12905.234
##
##   Akaike (AIC)                               25950.898
##   Bayesian (BIC)                             25998.519
##   Sample-size adjusted Bayesian (BIC)        25973.100
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.147
##   90 Percent confidence interval - lower         0.125
##   90 Percent confidence interval - upper         0.169
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.033
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SSmath =~
##     SSkMa_a           1.000
##     SSkMa_b           0.823    0.029   28.041    0.000
##     SSkMa_c           0.794    0.022   36.756    0.000
##     SSkMa_d           0.920    0.023   40.728    0.000
```

```
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           0.332    0.013   25.253    0.000
##    .SSkMa_b           0.820    0.024   34.594    0.000
##    .SSkMa_c           0.320    0.011   29.946    0.000
##    .SSkMa_d           0.189    0.009   19.857    0.000
##     SSmath            0.538    0.023   22.923    0.000
```

```
table(datenLV$Emigr)
```

```
##
##     Mig keinMig
##     493    1966
```

```
configural <- cfa(CFAmodel, data=datenLV, group = "Emigr")
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: group vari
```

```
summary(configural, fit.measures=TRUE)
```

```
## lavaan 0.6-8 ended normally after 35 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        24
##
##   Number of observations per group:               Used       Total
##     keinMig                                       1878        1966
##     Mig                                            465         493
##
## Model Test User Model:
##
##   Test statistic                                98.189
##   Degrees of freedom                                 4
##   P-value (Chi-square)                           0.000
##   Test statistic for each group:
##     keinMig                                     72.077
##     Mig                                         26.112
##
## Model Test Baseline Model:
##
##   Test statistic                              3389.720
##   Degrees of freedom                                12
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.972
##   Tucker-Lewis Index (TLI)                       0.916
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -10511.368
##   Loglikelihood unrestricted model (H1)     -10462.274
##
```

```
##   Akaike (AIC)                               21070.736
##   Bayesian (BIC)                             21208.957
##   Sample-size adjusted Bayesian (BIC)        21132.704
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.142
##   90 Percent confidence interval - lower         0.118
##   90 Percent confidence interval - upper         0.167
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.027
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
##
## Group 1 [keinMig]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SSmath =~
##     SSkMa_a           1.000
##     SSkMa_b           0.865    0.037   23.226    0.000
##     SSkMa_c           0.814    0.027   29.652    0.000
##     SSkMa_d           0.931    0.028   32.753    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           3.238    0.021  155.739    0.000
##    .SSkMa_b           3.013    0.025  122.777    0.000
##    .SSkMa_c           3.356    0.018  184.052    0.000
##    .SSkMa_d           3.377    0.018  189.921    0.000
##     SSmath            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           0.324    0.015   21.236    0.000
##    .SSkMa_b           0.766    0.027   27.915    0.000
##    .SSkMa_c           0.301    0.012   24.207    0.000
##    .SSkMa_d           0.171    0.011   15.854    0.000
##     SSmath            0.488    0.027   18.289    0.000
##
##
## Group 2 [Mig]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SSmath =~
```

```
##     SSkMa_a           1.000
##     SSkMa_b           0.758    0.066   11.542    0.000
##     SSkMa_c           0.699    0.050   14.064    0.000
##     SSkMa_d           0.916    0.053   17.127    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           3.069    0.044   69.482    0.000
##    .SSkMa_b           2.798    0.051   55.224    0.000
##    .SSkMa_c           3.239    0.039   82.839    0.000
##    .SSkMa_d           3.241    0.040   80.897    0.000
##     SSmath            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           0.280    0.033    8.428    0.000
##    .SSkMa_b           0.833    0.059   14.087    0.000
##    .SSkMa_c           0.404    0.031   13.177    0.000
##    .SSkMa_d           0.219    0.027    8.049    0.000
##     SSmath            0.628    0.063    9.963    0.000
```

```
weak.invariance <- cfa(CFAmodel, data=datenLV, group = "Emigr", group.equal = "loadings")
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: group varia
```

```
summary(weak.invariance, fit.measures = TRUE)
```

```
## lavaan 0.6-8 ended normally after 27 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        24
##   Number of equality constraints                     3
##
##   Number of observations per group:               Used       Total
##     keinMig                                       1878        1966
##     Mig                                            465         493
##
## Model Test User Model:
##
##   Test statistic                               104.031
##   Degrees of freedom                                 7
##   P-value (Chi-square)                           0.000
##   Test statistic for each group:
##     keinMig                                     73.273
##     Mig                                         30.758
##
## Model Test Baseline Model:
##
##   Test statistic                              3389.720
##   Degrees of freedom                                12
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
```

```
##   Comparative Fit Index (CFI)                    0.971
##   Tucker-Lewis Index (TLI)                       0.951
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -10514.289
##   Loglikelihood unrestricted model (H1)     -10462.274
##
##   Akaike (AIC)                               21070.578
##   Bayesian (BIC)                             21191.521
##   Sample-size adjusted Bayesian (BIC)        21124.800
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.109
##   90 Percent confidence interval - lower         0.091
##   90 Percent confidence interval - upper         0.128
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.032
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
##
## Group 1 [keinMig]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SSmath =~
##     SSkMa_a           1.000
##     SSkMa_b (.p2.)    0.840    0.032   25.944    0.000
##     SSkMa_c (.p3.)    0.790    0.024   32.950    0.000
##     SSkMa_d (.p4.)    0.927    0.025   37.016    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           3.238    0.021  154.981    0.000
##    .SSkMa_b           3.013    0.024  123.393    0.000
##    .SSkMa_c           3.356    0.018  185.466    0.000
##    .SSkMa_d           3.377    0.018  189.351    0.000
##     SSmath            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           0.322    0.015   21.310    0.000
##    .SSkMa_b           0.768    0.027   28.123    0.000
##    .SSkMa_c           0.304    0.012   24.751    0.000
##    .SSkMa_d           0.169    0.011   16.002    0.000
```

```
##     SSmath            0.498    0.026   19.451    0.000
##
##
## Group 2 [Mig]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SSmath =~
##     SSkMa_a           1.000
##     SSkMa_b (.p2.)    0.840    0.032   25.944    0.000
##     SSkMa_c (.p3.)    0.790    0.024   32.950    0.000
##     SSkMa_d (.p4.)    0.927    0.025   37.016    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           3.069    0.043   70.625    0.000
##    .SSkMa_b           2.798    0.052   54.123    0.000
##    .SSkMa_c           3.239    0.040   80.088    0.000
##    .SSkMa_d           3.241    0.040   81.950    0.000
##     SSmath            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           0.292    0.029   10.086    0.000
##    .SSkMa_b           0.829    0.059   14.044    0.000
##    .SSkMa_c           0.395    0.030   12.983    0.000
##    .SSkMa_d           0.223    0.024    9.499    0.000
##     SSmath            0.586    0.049   11.977    0.000
```

```
anova(weak.invariance, configural) # LRT
```

```
## Chi-Squared Difference Test
##
##                 Df   AIC   BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## configural       4 21071 21209  98.189
## weak.invariance  7 21071 21192 104.031     5.8415       3     0.1196
```

```
fit.stats <- rbind(fitmeasures(configural, fit.measures = c("chisq", "df", "rmsea", "tli", "cfi", "aic"
fitmeasures(weak.invariance, fit.measures = c("chisq", "df", "rmsea", "tli", "cfi", "aic")))
rownames(fit.stats) <- c("configural", "weak invariance")
fit.stats
```

```
##                     chisq df     rmsea       tli       cfi      aic
## configural       98.18934  4 0.1417750 0.9163436 0.9721145 21070.74
## weak invariance 104.03081  7 0.1087764 0.9507542 0.9712733 21070.58
```

```
strong.invariance <- cfa(CFAmodel, data=datenLV, group = "Emigr", group.equal = c( "loadings", "interce
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: group varia
```

```
summary(strong.invariance, fit.measures = TRUE)
```

```
## lavaan 0.6-8 ended normally after 39 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        25
```

```
##   Number of equality constraints                     7
##
##   Number of observations per group:               Used       Total
##     keinMig                                       1878        1966
##     Mig                                            465         493
##
## Model Test User Model:
##
##   Test statistic                               107.336
##   Degrees of freedom                                10
##   P-value (Chi-square)                           0.000
##   Test statistic for each group:
##     keinMig                                     73.780
##     Mig                                         33.556
##
## Model Test Baseline Model:
##
##   Test statistic                              3389.720
##   Degrees of freedom                                12
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.971
##   Tucker-Lewis Index (TLI)                       0.965
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -10515.941
##   Loglikelihood unrestricted model (H1)     -10462.274
##
##   Akaike (AIC)                               21067.883
##   Bayesian (BIC)                             21171.548
##   Sample-size adjusted Bayesian (BIC)        21114.359
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.091
##   90 Percent confidence interval - lower         0.076
##   90 Percent confidence interval - upper         0.107
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.032
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
##
## Group 1 [keinMig]:
```

```
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SSmath =~
##     SSkMa_a           1.000
##     SSkMa_b (.p2.)    0.844    0.032   26.189    0.000
##     SSkMa_c (.p3.)    0.788    0.024   33.102    0.000
##     SSkMa_d (.p4.)    0.925    0.025   37.264    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a (.10.)    3.237    0.020  159.328    0.000
##    .SSkMa_b (.11.)    2.999    0.023  130.339    0.000
##    .SSkMa_c (.12.)    3.358    0.017  192.037    0.000
##    .SSkMa_d (.13.)    3.380    0.018  192.150    0.000
##     SSmath            0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           0.321    0.015   21.285    0.000
##    .SSkMa_b           0.767    0.027   28.090    0.000
##    .SSkMa_c           0.304    0.012   24.775    0.000
##    .SSkMa_d           0.170    0.011   16.109    0.000
##     SSmath            0.499    0.026   19.521    0.000
##
##
## Group 2 [Mig]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SSmath =~
##     SSkMa_a           1.000
##     SSkMa_b (.p2.)    0.844    0.032   26.189    0.000
##     SSkMa_c (.p3.)    0.788    0.024   33.102    0.000
##     SSkMa_d (.p4.)    0.925    0.025   37.264    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a (.10.)    3.237    0.020  159.328    0.000
##    .SSkMa_b (.11.)    2.999    0.023  130.339    0.000
##    .SSkMa_c (.12.)    3.358    0.017  192.037    0.000
##    .SSkMa_d (.13.)    3.380    0.018  192.150    0.000
##     SSmath           -0.164    0.042   -3.867    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .SSkMa_a           0.291    0.029   10.062    0.000
##    .SSkMa_b           0.832    0.059   14.035    0.000
##    .SSkMa_c           0.395    0.030   12.989    0.000
##    .SSkMa_d           0.224    0.024    9.543    0.000
##     SSmath            0.587    0.049   11.990    0.000
```

```r
anova(strong.invariance, weak.invariance, configural)
```

```
## Chi-Squared Difference Test
```

```
##
##                   Df   AIC   BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## configural         4 21071 21209  98.189
## weak.invariance    7 21071 21192 104.031     5.8415       3     0.1196
## strong.invariance 10 21068 21172 107.336     3.3049       3     0.3470
```

```r
strict.invariance <- cfa(CFAmodel, data=datenLV, group = "Emigr", group.equal = c( "loadings", "interce
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: group vari
```

```r
anova(strict.invariance, strong.invariance, weak.invariance, configural)
```

```
## Chi-Squared Difference Test
##
##                   Df   AIC   BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## configural         4 21071 21209  98.189
## weak.invariance    7 21071 21192 104.031     5.8415       3   0.119583
## strong.invariance 10 21068 21172 107.336     3.3049       3   0.346958
## strict.invariance 14 21077 21158 124.388    17.0527       4   0.001888 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

confirmatory factor analysis

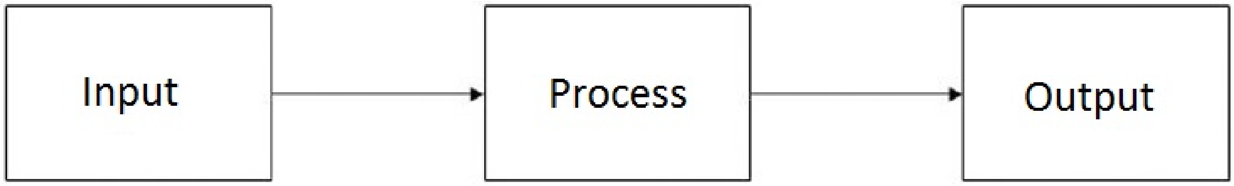
Verwendung des `lavaan` Paketes in R (siehe https://lavaan.ugent.be/), google group lavaan (https://groups.google.com/g/lavaan); es empfiehlt sich jedoch für komplexe Analysen Mplus zu verwenden (FIML, Bayesian SEM, ...)

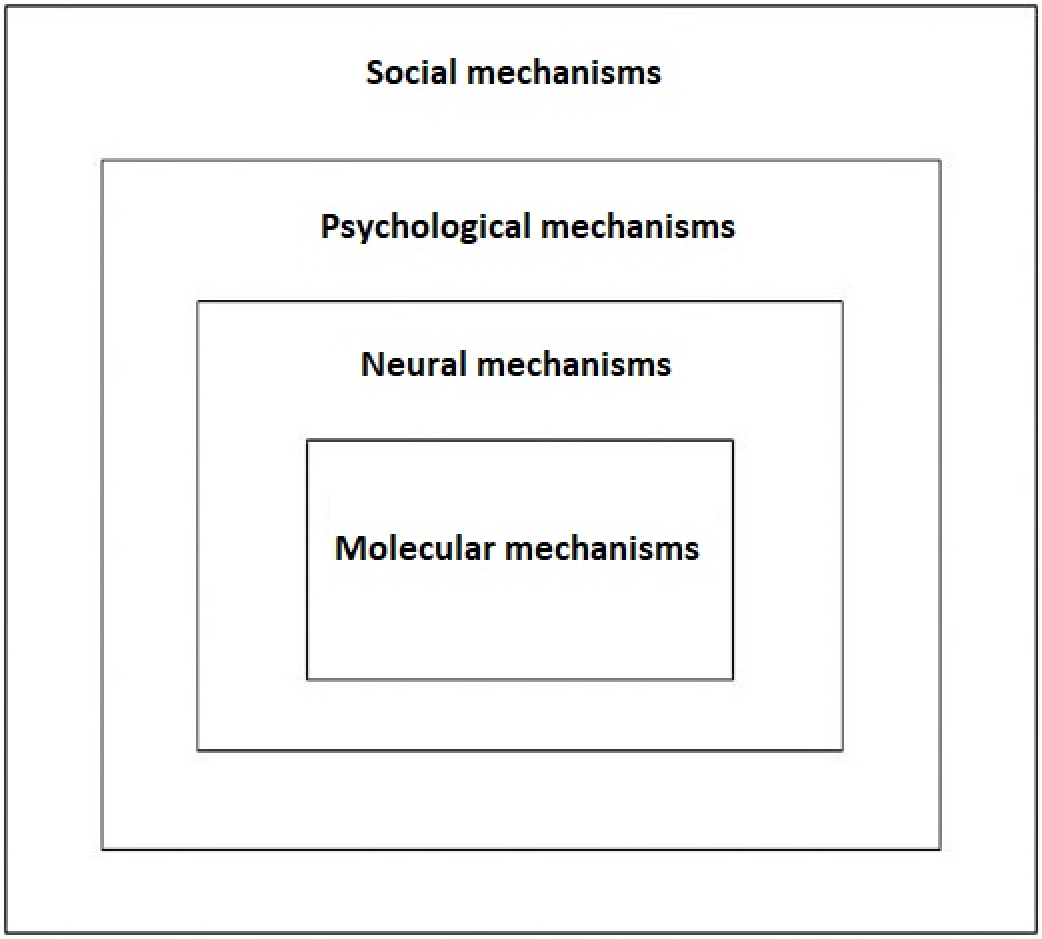
wichtigstes Grundlagenbuch zu SEM: Bollen (1989)

short theoretical: focus the structural modelling perspective

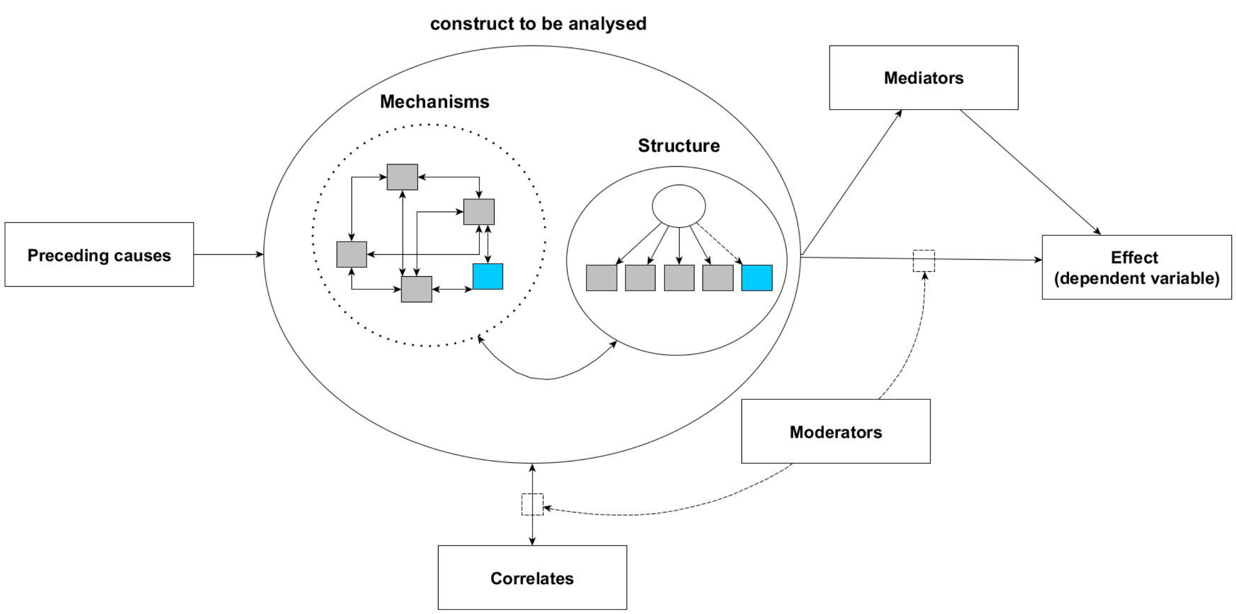
deduktive Methode

Context-Process-Input-Output model (bekannt in deutschsprachigen Raum durch Ditton (2000), entwickelt von Stufflebeam (1971); klare Ausführungen in Keller (2014)):





Dies lässt sich zusammenbauen zu einem nomologischen Netzwerk (= Testung Konstruktvalidität):



Welche möglicherweise Variablen interessant sind lässt sich aus einer graphischen theoretischen Ausarbeitung

(Pfaddiagramm) schrittweise aufbauen (Kapitel 7 “causal models” in Jaccard and Jacoby (2020)):

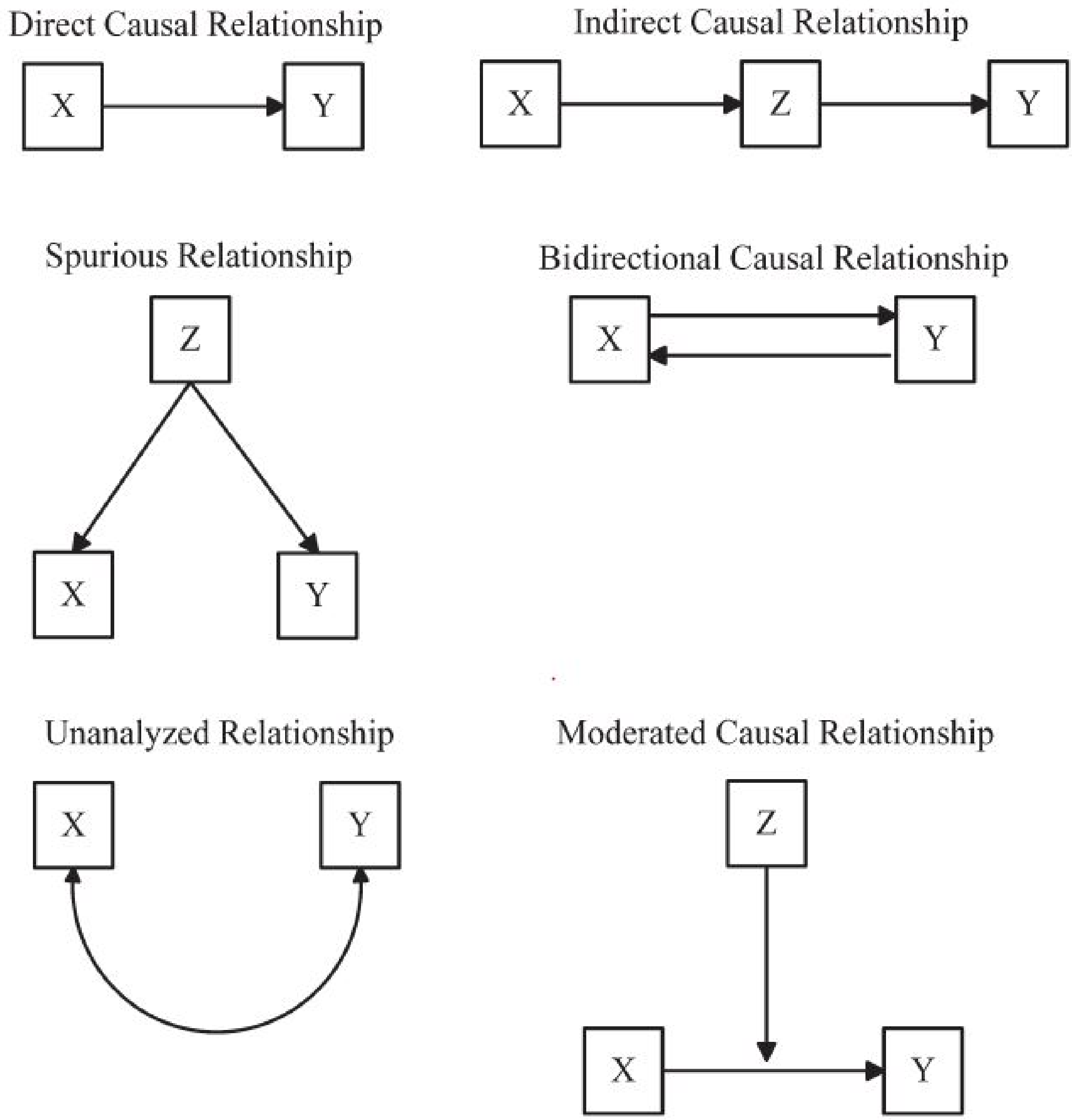


FIGURE 7.1. Relationships in causal models.

CFA (measurement model)

Abschnitt CFA / SEM orientiert sich an Kapitel 9-12 in Hair et al. (2019):

an sich sollten die einzelnen Messmodelle (CFAs) getrennt berechnet werden, hier wird jedoch aus Zeitdrücken direkt eine CFA erster Ordnung (*first order CFA*) für alle Messmodelle, die im Strukturgleichungsmodell verwendet werden gerechnet:

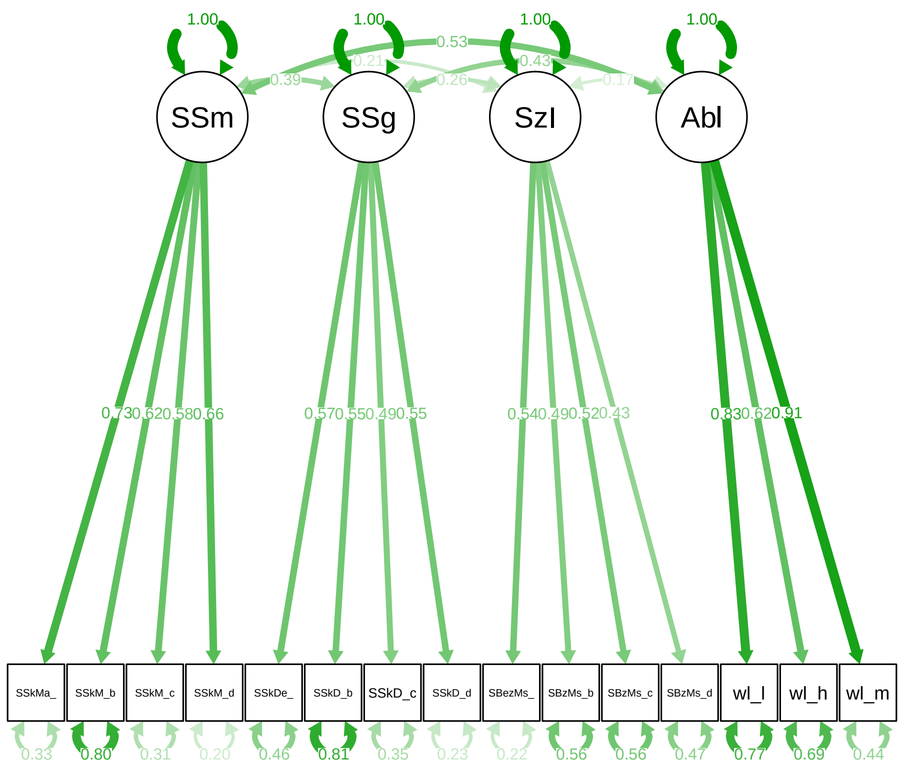
```r
firstorderCFA <- '
SSmath =~ SSkMa_a + SSkMa_b + SSkMa_c + SSkMa_d
SSgerman =~ SSkDe_a + SSkDe_b + SSkDe_c + SSkDe_d

SozInt =~ SBezMs_a + SBezMs_b + SBezMs_c + SBezMs_d

Abilities =~ wle_lesen + wle_hoeren + wle_mathe
'

# identification: Fixed factor
fit <-sem(firstorderCFA, data = datenLV, std.lv = TRUE)
summary(fit, standardized = TRUE, fit.measures=TRUE)
```

```
## lavaan 0.6-8 ended normally after 24 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        36
##
##                                                   Used       Total
##   Number of observations                          2566        3005
##
## Model Test User Model:
##
##   Test statistic                              1514.039
##   Degrees of freedom                                84
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                             11646.307
##   Degrees of freedom                               105
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.876
##   Tucker-Lewis Index (TLI)                       0.845
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -45769.003
##   Loglikelihood unrestricted model (H1)     -45011.984
##
##   Akaike (AIC)                               91610.007
##   Bayesian (BIC)                             91820.610
##   Sample-size adjusted Bayesian (BIC)        91706.228
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.081
##   90 Percent confidence interval - lower         0.078
##   90 Percent confidence interval - upper         0.085
##   P-value RMSEA <= 0.05                          0.000
```

```
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.054
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath =~
##     SSkMa_a           0.733    0.017   44.281    0.000    0.733    0.789
##     SSkMa_b           0.618    0.021   29.186    0.000    0.618    0.569
##     SSkMa_c           0.584    0.015   39.403    0.000    0.584    0.722
##     SSkMa_d           0.661    0.014   47.284    0.000    0.661    0.828
##   SSgerman =~
##     SSkDe_a           0.568    0.018   31.610    0.000    0.568    0.643
##     SSkDe_b           0.553    0.022   24.924    0.000    0.553    0.523
##     SSkDe_c           0.494    0.016   31.469    0.000    0.494    0.640
##     SSkDe_d           0.553    0.015   38.039    0.000    0.553    0.758
##   SozInt =~
##     SBezMs_a          0.544    0.016   34.545    0.000    0.544    0.757
##     SBezMs_b          0.495    0.019   25.382    0.000    0.495    0.553
##     SBezMs_c          0.524    0.020   26.359    0.000    0.524    0.573
##     SBezMs_d          0.429    0.018   24.247    0.000    0.429    0.530
##   Abilities =~
##     wle_lesen         0.830    0.024   34.203    0.000    0.830    0.687
##     wle_hoeren        0.618    0.021   29.300    0.000    0.618    0.597
##     wle_mathe         0.909    0.022   40.680    0.000    0.909    0.809
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath ~~
##     SSgerman          0.392    0.022   18.006    0.000    0.392    0.392
##     SozInt            0.212    0.024    8.711    0.000    0.212    0.212
##     Abilities         0.529    0.019   27.358    0.000    0.529    0.529
##   SSgerman ~~
##     SozInt            0.263    0.025   10.370    0.000    0.263    0.263
##     Abilities         0.427    0.023   18.929    0.000    0.427    0.427
##   SozInt ~~
##     Abilities         0.167    0.026    6.448    0.000    0.167    0.167
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a           0.326    0.013   24.737    0.000    0.326    0.378
##    .SSkMa_b           0.798    0.024   32.781    0.000    0.798    0.676
##    .SSkMa_c           0.313    0.011   28.688    0.000    0.313    0.478
##    .SSkMa_d           0.200    0.009   21.359    0.000    0.200    0.314
##    .SSkDe_a           0.457    0.016   27.929    0.000    0.457    0.587
##    .SSkDe_b           0.815    0.026   31.736    0.000    0.815    0.727
##    .SSkDe_c           0.350    0.012   28.040    0.000    0.350    0.590
```

```
##     .SSkDe_d           0.227    0.011    20.899    0.000    0.227    0.426
##     .SBezMs_a          0.221    0.013    17.345    0.000    0.221    0.427
##     .SBezMs_b          0.555    0.019    29.542    0.000    0.555    0.694
##     .SBezMs_c          0.560    0.019    28.764    0.000    0.560    0.671
##     .SBezMs_d          0.472    0.016    30.322    0.000    0.472    0.719
##     .wle_lesen         0.770    0.030    25.848    0.000    0.770    0.528
##     .wle_hoeren        0.690    0.023    30.039    0.000    0.690    0.643
##     .wle_mathe         0.437    0.027    16.471    0.000    0.437    0.346
##      SSmath            1.000                               1.000    1.000
##      SSgerman          1.000                               1.000    1.000
##      SozInt            1.000                               1.000    1.000
##      Abilities         1.000                               1.000    1.000
```

```r
semPlot::semPaths(object = fit, what = "est")
```



To account for the non-normal distribution of the questionnaire items and the small sample, the DWLS estimator was used and the X^2 statistic was mean and variance adjusted (e.g. chapter 11 in Hancock and Mueller (2013)):

! limited information approach; FIML, Bayesian SEM is possible in Mplus

```r
datenLV[,c("SSkMa_a",
        "SSkMa_b",
        "SSkMa_c",
        "SSkMa_d",
        "SSkDe_a",
        "SSkDe_b",
        "SSkDe_c",
```

```r
        "SSkDe_d",
        "SBezMs_a",
        "SBezMs_b",
        "SBezMs_c",
        "SBezMs_d")] <-
  lapply(datenLV[,c("SSkMa_a",
        "SSkMa_b",
        "SSkMa_c",
        "SSkMa_d",
        "SSkDe_a",
        "SSkDe_b",
        "SSkDe_c",
        "SSkDe_d",
        "SBezMs_a",
        "SBezMs_b",
        "SBezMs_c",
        "SBezMs_d")], ordered)

head(datenLV$SSkMa_a)
```

```
## [1] 3 3 3 4 3 4
## Levels: 1 < 2 < 3 < 4
```

```r
firstorderCFA <- '
SSmath =~ SSkMa_a + SSkMa_b + SSkMa_c + SSkMa_d
SSgerman =~ SSkDe_a + SSkDe_b + SSkDe_c + SSkDe_d

SozInt =~ SBezMs_a + SBezMs_b + SBezMs_c + SBezMs_d

Abilities =~ wle_lesen + wle_hoeren + wle_mathe
'

# identification: Marker variable method
fit <- sem(firstorderCFA, data = datenLV,
           ordered = c("SSkMa_a",
        "SSkMa_b",
        "SSkMa_c",
        "SSkMa_d",
        "SSkDe_a",
        "SSkDe_b",
        "SSkDe_c",
        "SSkDe_d",
        "SBezMs_a",
        "SBezMs_b",
        "SBezMs_c",
        "SBezMs_d"))

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

```
## lavaan 0.6-8 ended normally after 37 iterations
##
##   Estimator                                       DWLS
##   Optimization method                           NLMINB
##   Number of model parameters                        63
##
```

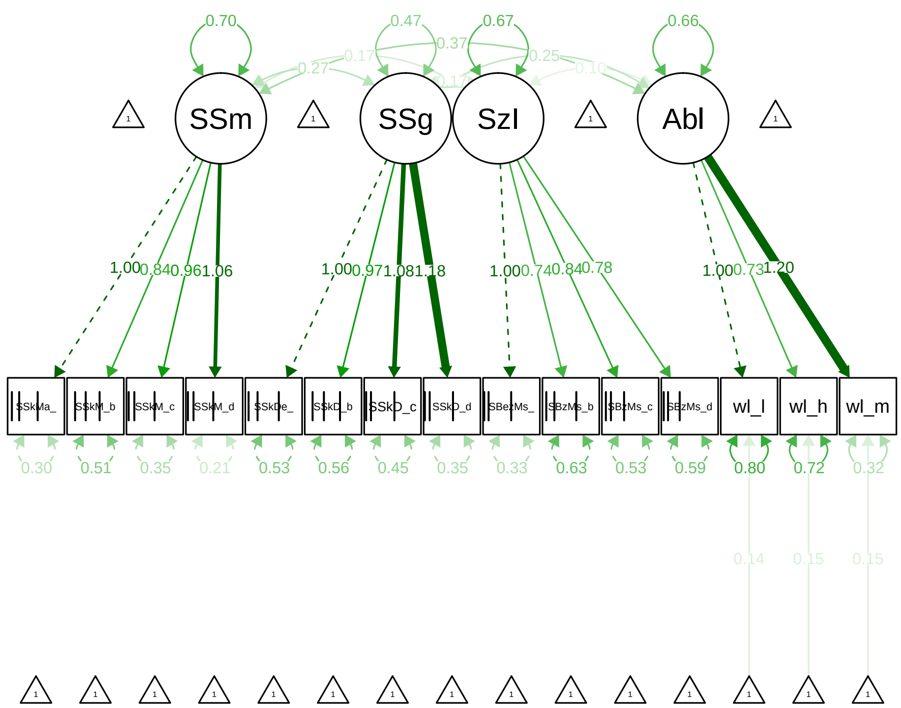
```
##                                                    Used       Total
##   Number of observations                          2566        3005
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                              1091.263    1391.590
##   Degrees of freedom                                84          84
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  0.795
##   Shift parameter                                           19.352
##        simple second-order correction
##
## Model Test Baseline Model:
##
##   Test statistic                             39677.932   22840.150
##   Degrees of freedom                               105         105
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.741
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.975       0.942
##   Tucker-Lewis Index (TLI)                       0.968       0.928
##
##   Robust Comparative Fit Index (CFI)                            NA
##   Robust Tucker-Lewis Index (TLI)                               NA
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.068       0.078
##   90 Percent confidence interval - lower         0.065       0.074
##   90 Percent confidence interval - upper         0.072       0.082
##   P-value RMSEA <= 0.05                          0.000       0.000
##
##   Robust RMSEA                                                  NA
##   90 Percent confidence interval - lower                        NA
##   90 Percent confidence interval - upper                        NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.057       0.057
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath =~
##     SSkMa_a           1.000                               0.837    0.837
##     SSkMa_b           0.837    0.018   46.522    0.000    0.701    0.701
##     SSkMa_c           0.965    0.015   66.163    0.000    0.807    0.807
```

```
##     SSkMa_d           1.059    0.015   70.858    0.000    0.886    0.886
##   SSgerman =~
##     SSkDe_a           1.000                               0.682    0.682
##     SSkDe_b           0.975    0.031   31.326    0.000    0.665    0.665
##     SSkDe_c           1.084    0.029   37.812    0.000    0.740    0.740
##     SSkDe_d           1.184    0.029   40.916    0.000    0.808    0.808
##   SozInt =~
##     SBezMs_a          1.000                               0.821    0.821
##     SBezMs_b          0.745    0.030   25.241    0.000    0.611    0.611
##     SBezMs_c          0.836    0.032   26.338    0.000    0.686    0.686
##     SBezMs_d          0.781    0.031   24.827    0.000    0.641    0.641
##   Abilities =~
##     wle_lesen         1.000                               0.811    0.672
##     wle_hoeren        0.731    0.035   20.902    0.000    0.593    0.573
##     wle_mathe         1.196    0.048   24.758    0.000    0.970    0.863
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath ~~
##     SSgerman          0.273    0.014   19.965    0.000    0.478    0.478
##     SozInt            0.166    0.017    9.494    0.000    0.241    0.241
##     Abilities         0.367    0.018   19.885    0.000    0.541    0.541
##   SSgerman ~~
##     SozInt            0.165    0.015   10.840    0.000    0.295    0.295
##     Abilities         0.253    0.016   16.022    0.000    0.457    0.457
##   SozInt ~~
##     Abilities         0.103    0.018    5.751    0.000    0.155    0.155
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a           0.000                               0.000    0.000
##    .SSkMa_b           0.000                               0.000    0.000
##    .SSkMa_c           0.000                               0.000    0.000
##    .SSkMa_d           0.000                               0.000    0.000
##    .SSkDe_a           0.000                               0.000    0.000
##    .SSkDe_b           0.000                               0.000    0.000
##    .SSkDe_c           0.000                               0.000    0.000
##    .SSkDe_d           0.000                               0.000    0.000
##    .SBezMs_a          0.000                               0.000    0.000
##    .SBezMs_b          0.000                               0.000    0.000
##    .SBezMs_c          0.000                               0.000    0.000
##    .SBezMs_d          0.000                               0.000    0.000
##    .wle_lesen         0.144    0.024    6.032    0.000    0.144    0.119
##    .wle_hoeren        0.148    0.021    7.186    0.000    0.148    0.143
##    .wle_mathe         0.151    0.022    6.788    0.000    0.151    0.134
##     SSmath            0.000                               0.000    0.000
##     SSgerman          0.000                               0.000    0.000
##     SozInt            0.000                               0.000    0.000
##     Abilities         0.000                               0.000    0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     SSkMa_a|t1       -1.438    0.037  -39.175    0.000   -1.438   -1.438
##     SSkMa_a|t2       -0.828    0.028  -29.478    0.000   -0.828   -0.828
```

```
##     SSkMa_a|t3        0.074    0.025    3.000    0.003    0.074    0.074
##     SSkMa_b|t1       -1.072    0.031  -34.946    0.000   -1.072   -1.072
##     SSkMa_b|t2       -0.442    0.026  -17.229    0.000   -0.442   -0.442
##     SSkMa_b|t3        0.185    0.025    7.417    0.000    0.185    0.185
##     SSkMa_c|t1       -1.758    0.045  -38.947    0.000   -1.758   -1.758
##     SSkMa_c|t2       -1.088    0.031  -35.229    0.000   -1.088   -1.088
##     SSkMa_c|t3       -0.014    0.025   -0.553    0.581   -0.014   -0.014
##     SSkMa_d|t1       -1.801    0.047  -38.664    0.000   -1.801   -1.801
##     SSkMa_d|t2       -1.109    0.031  -35.599    0.000   -1.109   -1.109
##     SSkMa_d|t3       -0.044    0.025   -1.776    0.076   -0.044   -0.044
##     SSkDe_a|t1       -1.575    0.040  -39.504    0.000   -1.575   -1.575
##     SSkDe_a|t2       -0.787    0.028  -28.379    0.000   -0.787   -0.787
##     SSkDe_a|t3        0.228    0.025    9.111    0.000    0.228    0.228
##     SSkDe_b|t1       -1.090    0.031  -35.260    0.000   -1.090   -1.090
##     SSkDe_b|t2       -0.420    0.026  -16.448    0.000   -0.420   -0.420
##     SSkDe_b|t3        0.308    0.025   12.218    0.000    0.308    0.308
##     SSkDe_c|t1       -1.816    0.047  -38.552    0.000   -1.816   -1.816
##     SSkDe_c|t2       -1.175    0.032  -36.630    0.000   -1.175   -1.175
##     SSkDe_c|t3        0.081    0.025    3.276    0.001    0.081    0.081
##     SSkDe_d|t1       -2.003    0.055  -36.643    0.000   -2.003   -2.003
##     SSkDe_d|t2       -1.244    0.033  -37.544    0.000   -1.244   -1.244
##     SSkDe_d|t3        0.043    0.025    1.737    0.082    0.043    0.043
##     SBezMs_a|t1      -2.033    0.056  -36.257    0.000   -2.033   -2.033
##     SBezMs_a|t2      -1.294    0.034  -38.099    0.000   -1.294   -1.294
##     SBezMs_a|t3      -0.012    0.025   -0.474    0.636   -0.012   -0.012
##     SBezMs_b|t1      -1.495    0.038  -39.394    0.000   -1.495   -1.495
##     SBezMs_b|t2      -0.808    0.028  -28.930    0.000   -0.808   -0.808
##     SBezMs_b|t3       0.243    0.025    9.702    0.000    0.243    0.243
##     SBezMs_c|t1      -1.475    0.037  -39.329    0.000   -1.475   -1.475
##     SBezMs_c|t2      -0.999    0.030  -33.512    0.000   -0.999   -0.999
##     SBezMs_c|t3      -0.278    0.025  -11.078    0.000   -0.278   -0.278
##     SBezMs_d|t1      -1.723    0.044  -39.140    0.000   -1.723   -1.723
##     SBezMs_d|t2      -1.183    0.032  -36.742    0.000   -1.183   -1.183
##     SBezMs_d|t3      -0.456    0.026  -17.735    0.000   -0.456   -0.456
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a           0.300                               0.300    0.300
##    .SSkMa_b           0.509                               0.509    0.509
##    .SSkMa_c           0.348                               0.348    0.348
##    .SSkMa_d           0.214                               0.214    0.214
##    .SSkDe_a           0.535                               0.535    0.535
##    .SSkDe_b           0.558                               0.558    0.558
##    .SSkDe_c           0.453                               0.453    0.453
##    .SSkDe_d           0.348                               0.348    0.348
##    .SBezMs_a          0.326                               0.326    0.326
##    .SBezMs_b          0.626                               0.626    0.626
##    .SBezMs_c          0.530                               0.530    0.530
##    .SBezMs_d          0.588                               0.588    0.588
##    .wle_lesen         0.801    0.032   25.318    0.000    0.801    0.549
##    .wle_hoeren        0.720    0.025   28.933    0.000    0.720    0.672
##    .wle_mathe         0.323    0.033    9.688    0.000    0.323    0.255
##     SSmath            0.700    0.015   47.431    0.000    1.000    1.000
##     SSgerman          0.465    0.019   24.768    0.000    1.000    1.000
```

```
##     SozInt            0.674    0.028   24.229    0.000    1.000    1.000
##     Abilities         0.658    0.039   16.728    0.000    1.000    1.000
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     SSkMa_a           1.000                               1.000    1.000
##     SSkMa_b           1.000                               1.000    1.000
##     SSkMa_c           1.000                               1.000    1.000
##     SSkMa_d           1.000                               1.000    1.000
##     SSkDe_a           1.000                               1.000    1.000
##     SSkDe_b           1.000                               1.000    1.000
##     SSkDe_c           1.000                               1.000    1.000
##     SSkDe_d           1.000                               1.000    1.000
##     SBezMs_a          1.000                               1.000    1.000
##     SBezMs_b          1.000                               1.000    1.000
##     SBezMs_c          1.000                               1.000    1.000
##     SBezMs_d          1.000                               1.000    1.000
```

```
semPlot::semPaths(object = fit, what = "est")
```



SEM (measurement model + structural model)

```
SEMmodel <- '
  # measurement models
SSmath =~ SSkMa_a + SSkMa_b + SSkMa_c + SSkMa_d
SSgerman =~ SSkDe_a + SSkDe_b + SSkDe_c + SSkDe_d
```

```r
SozInt =~ SBezMs_a + SBezMs_b + SBezMs_c + SBezMs_d

Abilities =~ wle_lesen + wle_hoeren + wle_mathe

  # regressions (+2 dummies)
  Abilities ~ SSmath + SSgerman + SozInt + Emigr + tr_sex + EHisei
'

# identification: Marker variable method
fit <- sem(SEMmodel, data = datenLV,
           ordered = c("SSkMa_a",
        "SSkMa_b",
        "SSkMa_c",
        "SSkMa_d",
        "SSkDe_a",
        "SSkDe_b",
        "SSkDe_c",
        "SSkDe_d",
        "SBezMs_a",
        "SBezMs_b",
        "SBezMs_c",
        "SBezMs_d"))

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

```
## lavaan 0.6-8 ended normally after 39 iterations
##
##   Estimator                                         DWLS
##   Optimization method                             NLMINB
##   Number of model parameters                          66
##
##                                                     Used       Total
##   Number of observations                            1973        3005
##
## Model Test User Model:
##                                                 Standard      Robust
##   Test Statistic                                1505.862    1548.544
##   Degrees of freedom                                 126         126
##   P-value (Chi-square)                             0.000       0.000
##   Scaling correction factor                                    0.994
##   Shift parameter                                             33.596
##        simple second-order correction
##
## Model Test Baseline Model:
##
##   Test statistic                               27860.196   16284.077
##   Degrees of freedom                                 105         105
##   P-value                                          0.000       0.000
##   Scaling correction factor                                    1.715
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                      0.950       0.912
```

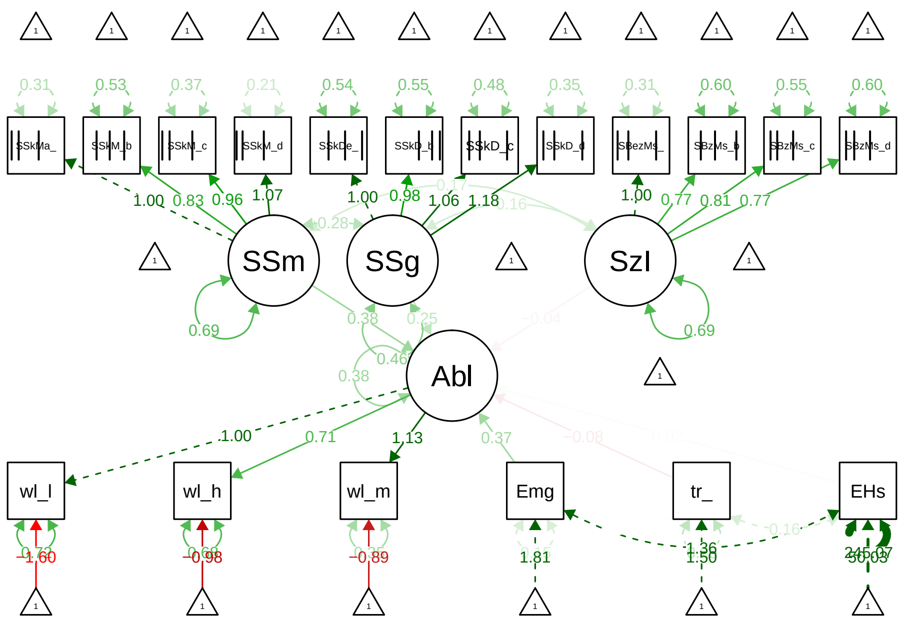
```
##   Tucker-Lewis Index (TLI)                       0.959       0.927
##
##   Robust Comparative Fit Index (CFI)                            NA
##   Robust Tucker-Lewis Index (TLI)                               NA
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.075       0.076
##   90 Percent confidence interval - lower         0.071       0.072
##   90 Percent confidence interval - upper         0.078       0.079
##   P-value RMSEA <= 0.05                          0.000       0.000
##
##   Robust RMSEA                                                  NA
##   90 Percent confidence interval - lower                        NA
##   90 Percent confidence interval - upper                        NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.055       0.055
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath =~
##     SSkMa_a           1.000                               0.828    0.828
##     SSkMa_b           0.830    0.021   38.856    0.000    0.688    0.688
##     SSkMa_c           0.958    0.018   54.715    0.000    0.794    0.794
##     SSkMa_d           1.072    0.018   59.008    0.000    0.887    0.887
##   SSgerman =~
##     SSkDe_a           1.000                               0.681    0.681
##     SSkDe_b           0.983    0.035   27.891    0.000    0.669    0.669
##     SSkDe_c           1.063    0.033   32.359    0.000    0.724    0.724
##     SSkDe_d           1.181    0.034   35.089    0.000    0.804    0.804
##   SozInt =~
##     SBezMs_a          1.000                               0.828    0.828
##     SBezMs_b          0.768    0.034   22.361    0.000    0.636    0.636
##     SBezMs_c          0.808    0.036   22.713    0.000    0.669    0.669
##     SBezMs_d          0.766    0.035   21.863    0.000    0.634    0.634
##   Abilities =~
##     wle_lesen         1.000                               0.814    0.693
##     wle_hoeren        0.706    0.039   17.878    0.000    0.574    0.572
##     wle_mathe         1.130    0.051   21.943    0.000    0.919    0.841
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Abilities ~
##     SSmath            0.383    0.031   12.450    0.000    0.389    0.389
##     SSgerman          0.248    0.038    6.571    0.000    0.207    0.207
##     SozInt           -0.039    0.027   -1.449    0.147   -0.040   -0.040
```

```
##     Emigr             0.369    0.051    7.206    0.000    0.454    0.178
##     tr_sex           -0.076    0.039   -1.937    0.053   -0.093   -0.046
##     EHisei            0.017    0.001   12.552    0.000    0.021    0.327
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SSmath ~~
##     SSgerman          0.279    0.016   17.807    0.000    0.495    0.495
##     SozInt            0.168    0.020    8.468    0.000    0.245    0.245
##   SSgerman ~~
##     SozInt            0.164    0.018    9.265    0.000    0.290    0.290
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a           0.000                               0.000    0.000
##    .SSkMa_b           0.000                               0.000    0.000
##    .SSkMa_c           0.000                               0.000    0.000
##    .SSkMa_d           0.000                               0.000    0.000
##    .SSkDe_a           0.000                               0.000    0.000
##    .SSkDe_b           0.000                               0.000    0.000
##    .SSkDe_c           0.000                               0.000    0.000
##    .SSkDe_d           0.000                               0.000    0.000
##    .SBezMs_a          0.000                               0.000    0.000
##    .SBezMs_b          0.000                               0.000    0.000
##    .SBezMs_c          0.000                               0.000    0.000
##    .SBezMs_d          0.000                               0.000    0.000
##    .wle_lesen        -1.602    0.152  -10.511    0.000   -1.602   -1.364
##    .wle_hoeren       -0.980    0.123   -7.956    0.000   -0.980   -0.976
##    .wle_mathe        -0.892    0.132   -6.777    0.000   -0.892   -0.816
##     SSmath            0.000                               0.000    0.000
##     SSgerman          0.000                               0.000    0.000
##     SozInt            0.000                               0.000    0.000
##    .Abilities         0.000                               0.000    0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     SSkMa_a|t1       -1.467    0.152   -9.631    0.000   -1.467   -1.467
##     SSkMa_a|t2       -0.859    0.151   -5.700    0.000   -0.859   -0.859
##     SSkMa_a|t3        0.121    0.150    0.805    0.421    0.121    0.121
##     SSkMa_b|t1       -0.723    0.148   -4.881    0.000   -0.723   -0.723
##     SSkMa_b|t2       -0.084    0.147   -0.572    0.567   -0.084   -0.084
##     SSkMa_b|t3        0.586    0.148    3.961    0.000    0.586    0.586
##     SSkMa_c|t1       -1.553    0.154  -10.081    0.000   -1.553   -1.553
##     SSkMa_c|t2       -0.882    0.151   -5.852    0.000   -0.882   -0.882
##     SSkMa_c|t3        0.225    0.151    1.492    0.136    0.225    0.225
##     SSkMa_d|t1       -1.786    0.157  -11.392    0.000   -1.786   -1.786
##     SSkMa_d|t2       -1.070    0.153   -7.005    0.000   -1.070   -1.070
##     SSkMa_d|t3        0.041    0.152    0.267    0.789    0.041    0.041
##     SSkDe_a|t1       -0.456    0.150   -3.050    0.002   -0.456   -0.456
##     SSkDe_a|t2        0.292    0.148    1.975    0.048    0.292    0.292
##     SSkDe_a|t3        1.367    0.150    9.100    0.000    1.367    1.367
##     SSkDe_b|t1        0.211    0.144    1.461    0.144    0.211    0.211
##     SSkDe_b|t2        0.899    0.144    6.249    0.000    0.899    0.899
##     SSkDe_b|t3        1.672    0.146   11.481    0.000    1.672    1.672
```

```
##     SSkDe_c|t1       -1.090    0.158   -6.886    0.000   -1.090   -1.090
##     SSkDe_c|t2       -0.456    0.152   -3.008    0.003   -0.456   -0.456
##     SSkDe_c|t3        0.857    0.153    5.611    0.000    0.857    0.857
##     SSkDe_d|t1       -1.218    0.161   -7.542    0.000   -1.218   -1.218
##     SSkDe_d|t2       -0.515    0.152   -3.382    0.001   -0.515   -0.515
##     SSkDe_d|t3        0.831    0.153    5.418    0.000    0.831    0.831
##     SBezMs_a|t1      -1.361    0.160   -8.500    0.000   -1.361   -1.361
##     SBezMs_a|t2      -0.596    0.152   -3.917    0.000   -0.596   -0.596
##     SBezMs_a|t3       0.722    0.154    4.697    0.000    0.722    0.722
##     SBezMs_b|t1      -0.772    0.144   -5.354    0.000   -0.772   -0.772
##     SBezMs_b|t2      -0.076    0.143   -0.529    0.597   -0.076   -0.076
##     SBezMs_b|t3       1.019    0.146    6.996    0.000    1.019    1.019
##     SBezMs_c|t1      -1.196    0.154   -7.759    0.000   -1.196   -1.196
##     SBezMs_c|t2      -0.728    0.153   -4.752    0.000   -0.728   -0.728
##     SBezMs_c|t3       0.034    0.154    0.223    0.824    0.034    0.034
##     SBezMs_d|t1      -1.203    0.164   -7.316    0.000   -1.203   -1.203
##     SBezMs_d|t2      -0.650    0.160   -4.056    0.000   -0.650   -0.650
##     SBezMs_d|t3       0.091    0.161    0.568    0.570    0.091    0.091
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SSkMa_a           0.314                               0.314    0.314
##    .SSkMa_b           0.527                               0.527    0.527
##    .SSkMa_c           0.370                               0.370    0.370
##    .SSkMa_d           0.213                               0.213    0.213
##    .SSkDe_a           0.536                               0.536    0.536
##    .SSkDe_b           0.552                               0.552    0.552
##    .SSkDe_c           0.476                               0.476    0.476
##    .SSkDe_d           0.354                               0.354    0.354
##    .SBezMs_a          0.315                               0.315    0.315
##    .SBezMs_b          0.596                               0.596    0.596
##    .SBezMs_c          0.553                               0.553    0.553
##    .SBezMs_d          0.598                               0.598    0.598
##    .wle_lesen         0.718    0.031   22.816    0.000    0.718    0.520
##    .wle_hoeren        0.679    0.026   25.904    0.000    0.679    0.673
##    .wle_mathe         0.350    0.031   11.178    0.000    0.350    0.293
##     SSmath            0.686    0.017   39.302    0.000    1.000    1.000
##     SSgerman          0.464    0.022   21.511    0.000    1.000    1.000
##     SozInt            0.685    0.032   21.401    0.000    1.000    1.000
##    .Abilities         0.378    0.028   13.341    0.000    0.571    0.571
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     SSkMa_a           1.000                               1.000    1.000
##     SSkMa_b           1.000                               1.000    1.000
##     SSkMa_c           1.000                               1.000    1.000
##     SSkMa_d           1.000                               1.000    1.000
##     SSkDe_a           1.000                               1.000    1.000
##     SSkDe_b           1.000                               1.000    1.000
##     SSkDe_c           1.000                               1.000    1.000
##     SSkDe_d           1.000                               1.000    1.000
##     SBezMs_a          1.000                               1.000    1.000
##     SBezMs_b          1.000                               1.000    1.000
##     SBezMs_c          1.000                               1.000    1.000
```

```
##     SBezMs_d          1.000                                     1.000    1.000
```

```
semPlot::semPaths(object = fit, what = "est")
```



short: item response theory

wörtliche Anmerkungen, wenn Zeit übrig

short: longitudinal data / multi-group analysis

wörtliche Anmerkungen, wenn Zeit übrig

References

Bollen, Kenneth. 1989. *Structural Equations with Latent Variables*. John Wiley.

Costello, Anna, and Jason Osborne. 2005. "Best Practices in Exploratory Factor Analysis: Four Recommendations for Getting the Most from Your Analysis." *Practical Assessment, Research & Evaluation* 10 (7): 1–9.

Ditton, Hartmut. 2000. "Qualitätskontrolle Und Qualitätssicherung in Schule Und Unterricht. Ein Überblick Zum Stand Der Empirischen Forschung." *Zeitschrift Für Pädagogik*, no. 41.

Grund, Simon, Oliver Lüdtke, and Alexander Robitzsch. 2018. "Multiple Imputation of Missing Data for Multilevel Models: Simulations and Recommendations." *Organizational Research Methods* 21 (1): 111–49.

Hair, Joseph F, William C Black, Barry J Babin, and Rolph E Anderson. 2019. *Multivariate Data Analysis*. Annabel Ainscow.

Hancock, Gregory R, and Ralph Mueller. 2013. *Structural Equation Modeling: A Second Course*. Iap.

Jaccard, James, and Jacob Jacoby. 2020. *Theory Construction and Model-Building Skills: A Practical Guide for Social Scientists*. Guilford Publications.

Keller, Florian. 2014. *Strukturelle Faktoren Des Bildungserfolgs: Wie Das Bildungssystem Den Übertritt Ins Berufsleben Bestimmt*. Springer-Verlag.

Marsh, Herbert W, Alexandre JS Morin, Philip D Parker, and Gurvinder Kaur. 2014. "Exploratory Structural Equation Modeling: An Integration of the Best Features of Exploratory and Confirmatory Factor Analysis." *Annual Review of Clinical Psychology* 10: 85–110.

Moosbrugger, Helfried, and Augustin Kelava. 2020. *Testtheorie Und Fragebogenkonstruktion*. Springer.

Mvududu, Nyaradzo, and Christopher Sink. 2013. "Factor Analysis in Counseling Research and Practice." *Counseling Outcome Research and Evaluation* 4 (2): 75–98.

Sijtsma, Klaas. 2009. "On the Use, the Misuse, and the Very Limited Usefulness of Cronbach's Alpha." *Psychometrika* 74 (1): 107–20.

Stufflebeam, Daniel. 1971. "The Relevance of the Cipp Evaluation Model for Educational Accountability."