

Multivariate Verfahren

Einheit 8: Multivariate (latente) Modellierung (2)

Wintersemester 2025 | Prof. Dr. Stephan Goerigk

Multivariate (latente) Modellierung

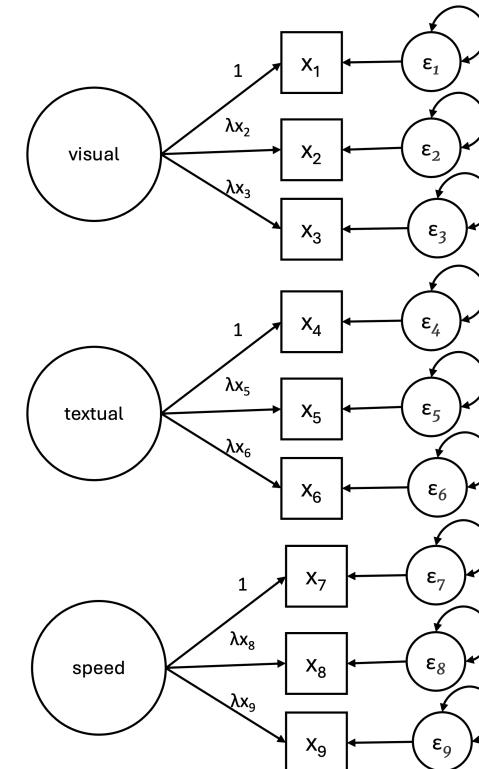
Strukturgleichungsmodelle in R - CFA Beispiel

- Das Modell (rechts) stellt die CFA für einen Kognitionstest dar (Holzinger & Swineford, 1939).
- Ein visueller Faktor, gemessen durch die 3 Variablen: x_1, x_2 und x_3 .
- Ein textueller Faktor, gemessen durch die 3 Variablen: x_4, x_5 und x_6 .
- Ein Geschwindigkeitsfaktor, gemessen durch die 3 Variablen: x_7, x_8 und x_9 .

Modell in lavaan:

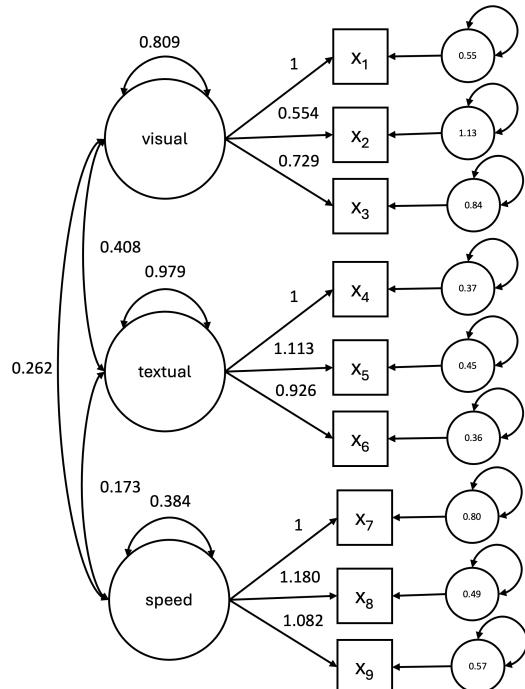
```
model <- 'visual  =~ x1 + x2 + x3
          textual =~ x4 + x5 + x6
          speed   =~ x7 + x8 + x9'

fit <- cfa(model, data = HolzingerSwineford1939)
```



Multivariate (latente) Modellierung

Strukturgleichungsmodelle in R - CFA Beispiel



parameterEstimates(fit)

Latent Variables:				
	Estimate	Std.Err	z-value	P(> z)
visual =~				
x1	1.000			
x2	0.554	0.100	5.554	0.000
x3	0.729	0.109	6.685	0.000
textual =~				
x4	1.000			
x5	1.113	0.065	17.014	0.000
x6	0.926	0.055	16.703	0.000
speed =~				
x7	1.000			
x8	1.180	0.165	7.152	0.000
x9	1.082	0.151	7.155	0.000
Covariances:				
	Estimate	Std.Err	z-value	P(> z)
visual ~~				
textual	0.408	0.074	5.552	0.000
speed	0.262	0.056	4.660	0.000
textual ~~				
speed	0.173	0.049	3.518	0.000
Variances:				
	Estimate	Std.Err	z-value	P(> z)
.x1	0.549	0.114	4.833	0.000
.x2	1.134	0.102	11.146	0.000
.x3	0.844	0.091	9.317	0.000
.x4	0.371	0.048	7.779	0.000
.x5	0.446	0.058	7.642	0.000
.x6	0.356	0.043	8.277	0.000
.x7	0.799	0.081	9.823	0.000
.x8	0.488	0.074	6.573	0.000
.x9	0.566	0.071	8.003	0.000
visual	0.809	0.145	5.564	0.000
textual	0.979	0.112	8.737	0.000
speed	0.384	0.086	4.451	0.000

Multivariate (latente) Modellierung

Strukturgleichungsmodelle in R - CFA Beispiel

Die Ausgabe besteht aus drei Teilen. Die ersten neun Zeilen werden als Header bezeichnet.

Der Header enthält die folgenden Informationen:

- die verwendete lavaan-Version
- ob die Optimierung normal beendet wurde oder nicht und wie viele Iterationen dafür benötigt wurden
- den Schätzer, der verwendet wurde (hier: ML, für Maximum-Likelihood)
- den Optimizer, der verwendet wurde, um die am besten passenden Parameterwerte für diesen Schätzer zu finden (hier: NLMINB)
- die Anzahl der Modellparameter (hier: 21)
- die Anzahl der Beobachtungen, die effektiv in der Analyse verwendet wurden (hier: 301)
- einen Abschnitt namens Model Test User Model, der einen Teststatistik-Wert, Freiheitsgrade und einen p-Wert für das vom Nutzer spezifizierte Modell bereitstellt.

fit

```
## lavaan 0.6-19 ended normally after 35 iterations
## 
##   Estimator                           ML
##   Optimization method            NLMINB
##   Number of model parameters          21
## 
##   Number of observations            301
## 
##   Model Test User Model:
## 
##   Test statistic                   85.306
##   Degrees of freedom                  24
##   P-value (Chi-square)            0.000
```

Multivariate (latente) Modellierung

Strukturgleichungsmodelle in R - CFA Beispiel

Anschließend werden alle freien (und festen) Parameter, die im Modell enthalten sind, tabellarisch dargestellt. Typischerweise werden zunächst die latenten Variablen angezeigt, gefolgt von Kovarianzen und (residualen) Varianzen.

- Die erste Spalte (**Estimate**) enthält den (geschätzten oder festgelegten) Parameterwert für jeden Modellparameter.
- Die zweite Spalte (**Std. err**) zeigt den Standardfehler für jeden geschätzten Parameter.
- Die dritte Spalte (**Z-value**) enthält die Wald-Statistik (die einfach durch Division des Parameterwerts durch seinen Standardfehler berechnet wird).
- Die letzte Spalte (**P(>|z|)**) enthält den p-Wert, um die Nullhypothese zu testen, dass der Parameterwert in der Population gleich null ist.

parameterEstimates(fit)				
Latent Variables:				
visual =~	Estimate	Std.Err	z-value	P(> z)
x1	1.000			
x2	0.554	0.100	5.554	0.000
x3	0.729	0.109	6.685	0.000
textual =~				
x4	1.000			
x5	1.113	0.065	17.014	0.000
x6	0.926	0.055	16.703	0.000
speed =~				
x7	1.000			
x8	1.180	0.165	7.152	0.000
x9	1.082	0.151	7.155	0.000
Covariances:				
visual ~~	Estimate	Std.Err	z-value	P(> z)
textual	0.408	0.074	5.552	0.000
speed	0.262	0.056	4.660	0.000
textual ~~				
speed	0.173	0.049	3.518	0.000
Variances:				
.x1	Estimate	Std.Err	z-value	P(> z)
	0.549	0.114	4.833	0.000
.x2	1.134	0.102	11.146	0.000
.x3	0.844	0.091	9.317	0.000
.x4	0.371	0.048	7.779	0.000
.x5	0.446	0.058	7.642	0.000
.x6	0.356	0.043	8.277	0.000
.x7	0.799	0.081	9.823	0.000
.x8	0.488	0.074	6.573	0.000
.x9	0.566	0.071	8.003	0.000
visual	0.809	0.145	5.564	0.000
textual	0.979	0.112	8.737	0.000
speed	0.384	0.086	4.451	0.000

Multivariate (latente) Modellierung

Strukturgleichungsmodelle in R - CFA Beispiel

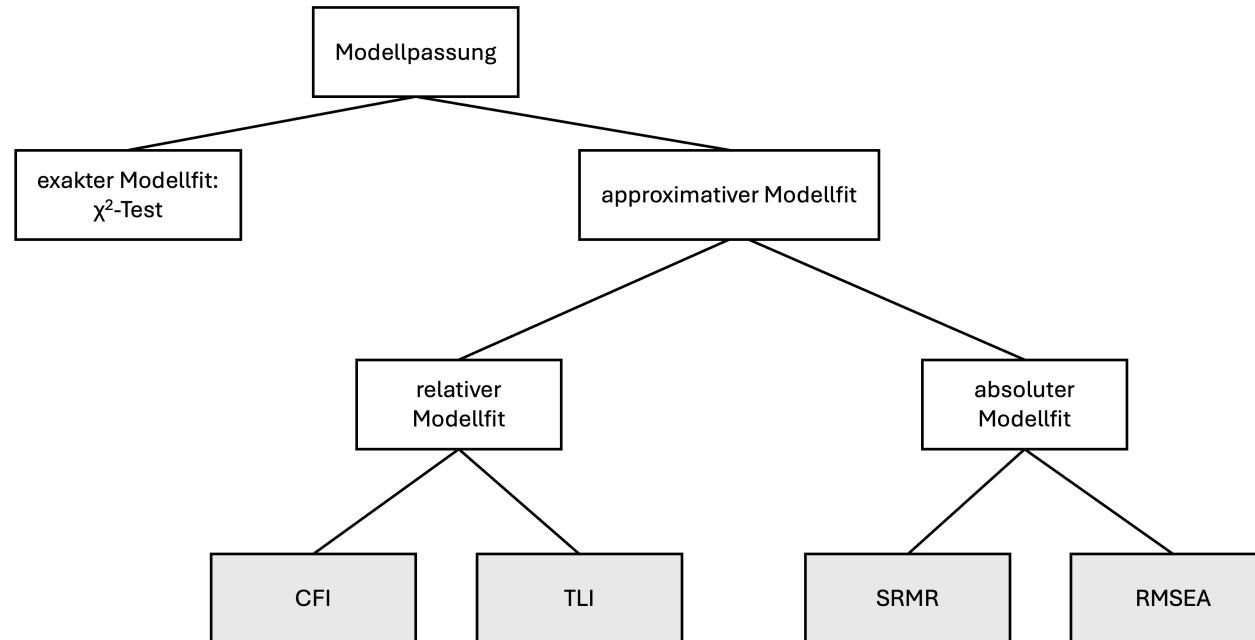
Abschnitt *Variances*:

- Vor den Namen der beobachteten Variablen steht ein Punkt.
 - Dies liegt daran, dass es sich um abhängige (oder endogene) Variablen handelt, die von den latenten Variablen vorhergesagt werden.
 - Der in der Ausgabe angezeigte Wert für die Varianz ist daher eine Schätzung der Residualvarianz, also der übrig gebliebenen Varianz, die nicht durch die Prädiktor(en) erklärt wird.
- Dagegen steht kein Punkt vor den Namen der latenten Variablen, da sie in diesem Modell exogene Variablen sind (es zeigen keine einseitigen Pfeile auf sie).
- Die angezeigten Werte für die Varianzen stellen hier die geschätzten Gesamtvarianzen der latenten Variablen dar.

parameterEstimates(fit)

Latent Variables:				
	Estimate	Std.Err	z-value	P(> z)
visual =~				
x1	1.000			
x2	0.554	0.100	5.554	0.000
x3	0.729	0.109	6.685	0.000
textual =~				
x4	1.000			
x5	1.113	0.065	17.014	0.000
x6	0.926	0.055	16.703	0.000
speed =~				
x7	1.000			
x8	1.180	0.165	7.152	0.000
x9	1.082	0.151	7.155	0.000
Covariances:				
	Estimate	Std.Err	z-value	P(> z)
visual ~~				
textual	0.408	0.074	5.552	0.000
speed	0.262	0.056	4.660	0.000
textual ~~				
speed	0.173	0.049	3.518	0.000
Variances:				
	Estimate	Std.Err	z-value	P(> z)
.x1	0.549	0.114	4.833	0.000
.x2	1.134	0.102	11.146	0.000
.x3	0.844	0.091	9.317	0.000
.x4	0.371	0.048	7.779	0.000
.x5	0.446	0.058	7.642	0.000
.x6	0.356	0.043	8.277	0.000
.x7	0.799	0.081	9.823	0.000
.x8	0.488	0.074	6.573	0.000
.x9	0.566	0.071	8.003	0.000
visual	0.809	0.145	5.564	0.000
textual	0.979	0.112	8.737	0.000
speed	0.384	0.086	4.451	0.000

Modellpassung und Fit indices



Modellpassung und Fit indices

`lavaan` liefert eine Vielzahl von Statistiken zur Modellpassung. Wir konzentrieren uns jedoch auf die 5 gängigsten:

- Modell-Chi-Quadrat: Dies ist die Chi-Quadrat-Statistik, die aus der Maximum-Likelihood-Schätzung gewonnen wird (in `lavaan` als Test Statistic for the Model Test User Model bezeichnet).
- CFI (Comparative Fit Index): Werte liegen zwischen 0 und 1. Werte > 0.90 (konservativ: 0.95) deuten auf eine gute Modellanpassung hin.
- TLI (Tucker-Lewis-Index): Auch dieser liegt zwischen 0 und 1. Werte > 1 werden auf 1 gerundet. Werte über 0.90 weisen auf eine gute Modellanpassung hin. Wenn CFI und TLI beide unter 1 liegen, ist der CFI stets größer als der TLI.
- SRMR (Standardized-Root-Mean-Residual)
- RMSEA (Root Mean Square Error of Approximation):
 - In `lavaan` wird zusätzlich ein p-Wert für die „Close Fit“-Hypothese angegeben, der testet, ob $RMSEA < 0.05$ ist.
 - Wird das Modell abgelehnt, bedeutet dies, dass Ihr Modell keine „nahe“ Anpassung an die Daten zeigt.

→ Diese Fit-Maße gehören zu den wichtigsten Indikatoren für die Bewertung von Strukturgleichungsmodellen.

Modellpassung und Fit indices

- In `lavaan` können Maße zur Modellpassung angezeigt werden
- Dafür setzen wir im `summary()` Befehl, `fit.measures = TRUE` ein.
- Die wichtigsten Indizes sind standardmäßig enthalten
- Für zusätzliche Maße kann `fitmeasures(fit)` genutzt werden

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

Model Test User Model:		
Test statistic	85.306	
Degrees of freedom	24	
P-value (Chi-square)	0.000	
Model Test Baseline Model:		
Test statistic	918.852	
Degrees of freedom	36	
P-value	0.000	
User Model versus Baseline Model:		
Comparative Fit Index (CFI)	0.931	
Tucker-Lewis Index (TLI)	0.896	
Loglikelihood and Information Criteria:		
Loglikelihood user model (H0)	-3737.745	
Loglikelihood unrestricted model (H1)	-3695.092	
Akaike (AIC)	7517.490	
Bayesian (BIC)	7595.339	
Sample-size adjusted Bayesian (SABIC)	7528.739	
Root Mean Square Error of Approximation:		
RMSEA	0.092	
90 Percent confidence interval - lower	0.071	
90 Percent confidence interval - upper	0.114	
P-value H_0: RMSEA <= 0.050	0.001	
P-value H_0: RMSEA >= 0.080	0.840	
Standardized Root Mean Square Residual:		
SRMR	0.065	

Modellpassung und Fit indices

χ^2 -Test

Das Modell-Chi-Quadrat wird definiert als entweder NF_{ML} oder $(N - 1)F_{ML}$ (je nach statistischem Paket)

$$(N - 1)F_{ML}[S, \Sigma(\hat{\theta})] \sim \chi^2(df) \quad \text{mit } df = p - q$$

- p : Anzahl bekannter Parameter
- q : Anzahl zu schätzender Parameter
- N : Anzahl der Versuchspersonen
- F_{ML} ist die Fit-Funktion aus der Maximum-Likelihood-Methode ist, die zur Schätzung der Parameter im Modell verwendet wird.

Das Hypothesenpaar des χ^2 -Tests lautet

- $\mathbf{H_0}: \Sigma = S$
- $\mathbf{H_1}: \Sigma \neq S$
- Unter $\mathbf{H_0}$ ist dieser Wert χ^2 -verteilt mit $df = p - q$.

Modellpassung und Fit indices

χ^2 -Test

- Je größer der χ^2 -Wert, desto größer ist der Unterschied zwischen der durch das Modell implizierten Kovarianzmatrix und der beobachteten Kovarianzmatrix.
- Ergibt der Signifikanztest für diesen χ^2 -Wert ein $p < 0.05$, wird die H_0 abgelehnt.
 - Das bedeutet, die modell-implizierte Kovarianzmatrix weicht signifikant von der empirischen Kovarianzmatrix ab.
- Wird der Test **nicht signifikant**, spricht dies für einen guten Modellfit.
 - Dadurch steigt die Wahrscheinlichkeit, dass das Modell abgelehnt wird.

Modellpassung und Fit indices

χ^2 -Test und Stichprobengröße/Power

- Test ist sehr abhängig von der Stichprobengröße
- Große Stichproben:
 - In der Literatur ist gut dokumentiert, dass der χ^2 -Test in der Modellprüfung bei großen Stichproben überempfindlich reagiert.
 - Für Modelle mit 75 bis 200 Fällen ein vernünftiges Maß für die Modellanpassung
 - Bei 400 oder mehr Fällen fällt er nahezu immer signifikant aus (Kenny, 2003).
- Kleine Stichproben:
 - Auch große Abweichungen von einem perfekten Modell führen nicht zwangsläufig zur Ablehnung des Modells (underpowered).

Modellpassung und Fit indices

χ^2 -Test und Stichprobengröße/Power

- Der χ^2 -Test ist empfindlich gegenüber großen Stichprobenumfängen, aber bedeutet das, dass wir uns auf kleine Stichproben beschränken sollten?
 - Nein – größere Stichproben sind vorzuziehen.
 - SEM sind für große Stichproben ausgelegt - Theorie basiert auf der Annahme, dass die Stichprobe so groß wie möglich ist.

Benötigte Stichprobengröße im SEM:

- Die $n : q$ Regel- (Kline, 2016)
 - Stichprobengröße sollte von der Anzahl der Parameter in Ihrem Modell abhängen
 - empfohlenes Verhältnis von 20:1
 - Das bedeutet, dass bei 10 Parametern die Stichprobengröße mindestens $n = 200$ betragen sollte.
 - Laut Kline ist eine Stichprobe von weniger als 100 Fällen nahezu immer unzureichend.

Modellpassung und Fit indices

Exakter vs. approximativer Modellfit

Historisch war der χ^2 -Test die einzige Methode zur Prüfung der Modellanpassung.

- In der Praxis wurde die H_0 aufgrund der Empfindlichkeit des χ^2 -Tests häufig abgelehnt
- Daher wurden approximative Fit-Indizes entwickelt → basieren nicht auf Verwerfung der H_0
- Diese approximativen Fit-Indizes lassen sich weiter unterteilen in:
 - Absolute Fit-Indizes
 - Relative Fit-Indizes (aka inkrementelle Fit-Indizes)

Modellpassung und Fit indices

Approximative Fit-Indizes

- Relative Fit-Indizes (z.B. CFI, TLI)
 - Vergleichen Abweichung des genutzten Modells vom schlechtest passenden Modell (auch Baseline-Modell genannt) vs. Abweichung des saturierten Modells (auch bestes passendes Modell) vom Nullmodell.
 - Nullmodell = alle Variablen unkorreliert (außer den Fehlertermen keine latenten Variablen und keine Zusammenhänge zwischen den Variablen)
 - Verhältnis sollte 1 betragen, wenn die Abweichung des genutzten Modells der Abweichung des saturierten Modells entspricht.
 - Je größer jedoch die Diskrepanz zwischen den Abweichungen ist, desto näher liegt das Verhältnis bei 0
- Absolute Fit-Indizes (z.B. RSMEA, SRMR)
 - Vergleichen genutztes Modell mit den beobachteten Daten.

Modellpassung und Fit indices

Relative Fit-Indizes - CFI (Comparative Fit Index)

- CFI vergleicht genutztes Modells mit restriktiveren Nullmodell

$$CFI = 1 - \frac{\chi_M^2 - df_M}{\chi_N^2 - df_N}$$

- χ_M^2 : Teststatistik des genutzten Modells
- χ_N^2 : Teststatistik des Nullmodells
- Der Wertebereich liegt zwischen 0 und 1 (1 = optimaler Fit)
- Vorteil: weniger sensitiv gegenüber der Stichprobengröße (N)

Modellpassung und Fit indices

Relative Fit-Indizes - CFI (Comparative Fit Index)

- Berechnung des CFI im Beispiel:

$$CFI = 1 - \frac{\chi_M^2 - df_M}{\chi_N^2 - df_N}$$

$$CFI = 1 - \frac{85.306 - 24}{918.852 - 36} = 0.931$$

- Typische Empfehlung:

- CFI sollte > 0.95 sein (Hu & Bentler, 1998)

Model Test User Model:

Test statistic	85.306
Degrees of freedom	24
P-value (Chi-square)	0.000

Model Test Baseline Model:

Test statistic	918.852
Degrees of freedom	36
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.931
Tucker-Lewis Index (TLI)	0.896

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-3737.745
Loglikelihood unrestricted model (H1)	-3695.092
Akaike (AIC)	7517.490
Bayesian (BIC)	7595.339
Sample-size adjusted Bayesian (SABIC)	7528.739

Root Mean Square Error of Approximation:

RMSEA	0.092
90 Percent confidence interval - lower	0.071
90 Percent confidence interval - upper	0.114
P-value H_0: RMSEA <= 0.050	0.001
P-value H_0: RMSEA >= 0.080	0.840

Standardized Root Mean Square Residual:

SRMR	0.065
------	-------

Modellpassung und Fit indices

Relative Fit-Indizes - TLI (Tucker Lewis Index)

- TLI wird häufig zusammen mit dem CFI ausgegeben
- TLI nutzt relatives χ^2 (weniger empfindlich gegenüber der Stichprobengröße)
- Ein perfekt angepasstes Modell ergibt einen TLI, der genau 1 beträgt.
- Da der TLI und der CFI hoch korreliert sind, sollte nur einer der beiden berichtet werden.

$$TLI = \frac{\frac{\chi^2_N}{df_N} - \frac{\chi^2_M}{df_M}}{\frac{\chi^2_N}{df_N} - 1}$$

Modellpassung und Fit indices

Absolute Fit-Indizes - RMSEA (Root-Mean-Square-Error of Approximation)

$$RMSEA = \sqrt{\frac{\chi^2 - df}{N \cdot df}}$$

- RMSEA berücksichtigt sowohl Stichprobegröße als auch Modellkomplexität
 - Je komplexer ein Modell, desto weniger Freiheitsgrade
 - Nenner wird kleiner → RMSEA wird größer
- Wertebereich zwischen 0 und 1 (optimaler Wert = 0)
- Nachteil: sensibel gegenüber fehlspezifizierten Ladungen und Pfaden.

Modellpassung und Fit indices

Absolute Fit-Indizes - RMSEA (Root-Mean-Square-Error of Approximation)

- Berechnung des RMSEA im Beispiel:

$$RMSEA = \sqrt{\frac{\chi^2 - df}{N \cdot df}}$$

$$RMSEA = \sqrt{\frac{85.306 - 24}{301 \cdot 24}} = 0.092$$

- Typische Empfehlung (Hu & Bentler, 1998):
 - RMSEA ≤ 0.06 bei $N > 250$.
 - RMSEA ≤ 0.08 bei $N < 250$.

Model Test User Model:		
Test statistic	85.306	
Degrees of freedom	24	
P-value (Chi-square)	0.000	
Model Test Baseline Model:		
Test statistic	918.852	
Degrees of freedom	36	
P-value	0.000	
User Model versus Baseline Model:		
Comparative Fit Index (CFI)	0.931	
Tucker-Lewis Index (TLI)	0.896	
Loglikelihood and Information Criteria:		
Loglikelihood user model (H0)	-3737.745	
Loglikelihood unrestricted model (H1)	-3695.092	
Akaike (AIC)	7517.490	
Bayesian (BIC)	7595.339	
Sample-size adjusted Bayesian (SABIC)	7528.739	
Root Mean Square Error of Approximation:		
RMSEA	0.092	
90 Percent confidence interval - lower	0.071	
90 Percent confidence interval - upper	0.114	
P-value H_0: RMSEA <= 0.050	0.001	
P-value H_0: RMSEA >= 0.080	0.840	
Standardized Root Mean Square Residual:		
SRMR	0.065	

Modellpassung und Fit indices

Absolute Fit-Indizes - SRMR (Standardized-Root-Mean-Residual)

$$SRMR = \sqrt{\sum_j \sum_{k < j} \frac{r_{jk}^2}{e}}$$

$$r_{jk} = \frac{S_{jk}}{S_j \cdot S_k} - \frac{\hat{\sigma}_{jk}}{\hat{\sigma}_j \cdot \hat{\sigma}_k}$$

$$e = \frac{p \cdot (p + 1)}{2}$$

- Wertebereich zwischen 0 und 1 (optimale Wert = 0)
- SRMR gibt durchschnittliche Abweichung der beobachteten (S) von der modellimplizierten Korrelationsmatrix (Σ) an.
- Formel:
 - p : Anzahl der manifesten Variablen
 - S : modellimplizierte SDs/Kovarianz
 - σ : aus Stichprobe geschätzte SDs/Kovarianz
- SRMR berücksichtigt weder Modellkomplexität noch die Stichprobengröße

Modellpassung und Fit indices

Zusammenfassung der Cutoff-Empfehlungen:

- Cutoff-Werte für ML- Schätzalgorithmus (Hu & Bentler, 1999)
 - RMSEA < 0.06 bei $N > 250$; < 0.08 bei $N \leq 250$
 - SRMR < 0.11
 - CFI > 0.95
- Kritik an Cut-Off Werten
 - Cutoffs in der Praxis oft nicht erreichbar (Marsh, 2004)
 - Hu & Bentler Empfehlungen in Daten mit sehr hohen Ladungen demonstriert

Modellpassung und Fit indices

Voraussetzungen für Modelltests

- Linearität
- Keine (Multi-) Kollinearität
 - mehrere Items korrelieren sehr hoch miteinander (Daumenregel: $r > .85$ problematisch)
- Ausreichende Stichprobengröße
- Angemessene Anzahl von Indikatoren pro latenter Variable
 - Daumenregel: mindestens vier Items pro latenter Variable
 - Mehr Items pro Faktor führen zu besseren Schätzungen (Marsh, Hau, Balla & Grayson, 1998)
- Zutreffen angenommener Verteilungen

Multivariate (latente) Modellierung

Signifikanztest für einzelne Modellparameter

- Einzelne Modellparameter: z.B. Ladungen, Regressionen, Kovarianzen
- **lavaan** gibt Hypothesentest für Einzelparameter aus
 - H_0 : Parameter = 0
 - H_1 : Parameter $\neq 0$
- Teststatistik ist ein z-verteilter **critical ratio** (ähnlich Wald-Test bei Regression):

$$C. R. = \frac{\text{Estimate}}{\text{SE}}$$

- Wenn $P(>|z|) < .05$ wird H_0 verworfen.

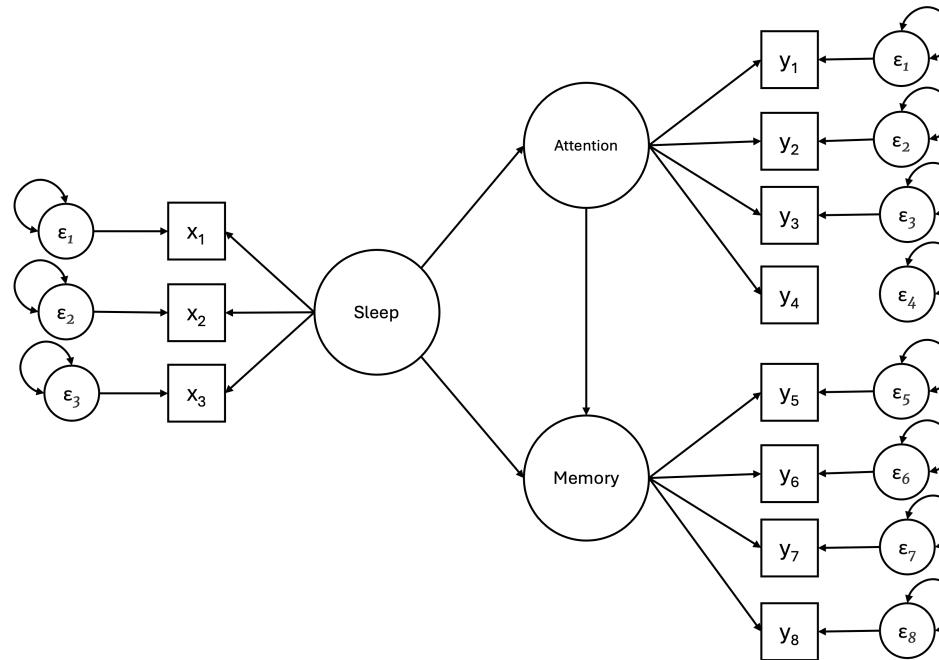
Latent Variables:					
	Estimate	Std.Err	z-value	P(> z)	
visual =~					
x1	1.000				
x2	0.554	0.100	5.554	0.000	
x3	0.729	0.109	6.685	0.000	
textual =~					
x4	1.000				
x5	1.113	0.065	17.014	0.000	
x6	0.926	0.055	16.703	0.000	
speed =~					
x7	1.000				
x8	1.180	0.165	7.152	0.000	
x9	1.082	0.151	7.155	0.000	
Covariances:					
	Estimate	Std.Err	z-value	P(> z)	
visual ~~					
textual	0.408	0.074	5.552	0.000	
speed	0.262	0.056	4.660	0.000	
textual ~~					
speed	0.173	0.049	3.518	0.000	
Variances:					
	Estimate	Std.Err	z-value	P(> z)	
.x1	0.549	0.114	4.833	0.000	
.x2	1.134	0.102	11.146	0.000	
.x3	0.844	0.091	9.317	0.000	
.x4	0.371	0.048	7.779	0.000	
.x5	0.446	0.058	7.642	0.000	
.x6	0.356	0.043	8.277	0.000	
.x7	0.799	0.081	9.823	0.000	
.x8	0.488	0.074	6.573	0.000	
.x9	0.566	0.071	8.003	0.000	
visual	0.809	0.145	5.564	0.000	
textual	0.979	0.112	8.737	0.000	
speed	0.384	0.086	4.451	0.000	

Fit Funktionen in `lavaan`

- `lavaan` unterstützt folgende Funktionen um ein Modell auf die Daten anzupassen:
 - `cfa()` → Konfirmatorische Faktorenanalysen
 - `sem()` → Strukturgleichungsmodelle
 - `growth()` → latente Wachstuskurvenmodelle (Modelle mit Messwiederholungen)
- Benutzerfreundliche Funktionen, die viele Details automatisch übernehmen
- Um automatische Durchführung bestimmter Schritte zu vermeiden, kann Low-Level-Funktion `lavaan()` verwendet werden, die volle Kontrolle ermöglicht.

Strukturgleichungsmodelle in R - SEM Beispiel

- Multivariate Zusammenhänge zwischen Schlafdauer, Aufmerksamkeit und Gedächtnisleistung.
- Aufmerksamkeit gleichzeitig endogen (wird erklärt durch Schlafdauer) und exogen (erklärt Gedächtnisleistung).



Strukturgleichungsmodelle in R - SEM Beispiel

- Multivariate Zusammenhänge zwischen Schlafdauer, Aufmerksamkeit und Gedächtnisleistung.
- Aufmerksamkeit gleichzeitig endogen (wird erklärt durch Schlafdauer) und exogen (erklärt Gedächtnisleistung).

Aufstellen des Modells in `lavaan`:

```
model <-   
  # measurement model  
  sleep =~ x1 + x2 + x3  
  attention =~ y1 + y2 + y3 + y4  
  memory =~ y5 + y6 + y7 + y8  
  
  # regressions  
  attention ~ sleep  
  memory ~ sleep + attention  
'  
  
fit <- sem(model, data = df)
```

In diesem Beispiel verwenden wir 2 Formeltypen:

- Definitionen latenter Variablen (mit dem Operator `=~`)
- Regressionsformeln (mit dem Operator `~`).

Strukturgleichungsmodelle in R - SEM Beispiel

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

Latent Variables:						
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
sleep =~						
x1	1.000				0.669	0.920
x2	2.182	0.139	15.714	0.000	1.461	0.973
x3	1.819	0.152	11.956	0.000	1.218	0.872
attention =~						
y1	1.000				2.201	0.845
y2	1.354	0.175	7.755	0.000	2.980	0.760
y3	1.044	0.150	6.961	0.000	2.298	0.705
y4	1.300	0.138	9.412	0.000	2.860	0.860
memory =~						
y5	1.000				2.084	0.803
y6	1.258	0.164	7.651	0.000	2.623	0.783
y7	1.282	0.158	8.137	0.000	2.673	0.819
y8	1.310	0.154	8.529	0.000	2.730	0.847
Regressions:						
	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
attention ~						
sleep	1.474	0.392	3.763	0.000	0.448	0.448
memory ~						
sleep	0.453	0.220	2.064	0.039	0.146	0.146
attention	0.864	0.113	7.671	0.000	0.913	0.913

Strukturgleichungsmodelle in R - SEM Beispiel

Zahlen im Output standardisieren

- Das Argument `standardized = TRUE` innerhalb der `summary()` Funktion erweitert den Output um standardisierte Parameterwerte.
- Es werden zwei zusätzliche Spalten mit standardisierten Parameterwerten angezeigt:
 - In der ersten Spalte (beschriftet mit `Std.lv`) werden nur die latenten Variablen standardisiert.
 - In der zweiten Spalte (beschriftet mit `Std.all`) werden sowohl die latenten als auch die beobachteten Variablen standardisiert.
- Letztere wird oft als „vollständig standardisierte Lösung“ bezeichnet.

Strukturgleichungsmodelle in R

Fixieren von Modellparametern

- Standardmäßig fixiert lavaan die Faktorladung des ersten Indikators immer auf 1
- Die anderen Parameter sind frei und ihre Werte werden vom Modell geschätzt.
- Es gibt jedoch inhaltliche Gründe Parameter in Modellen zu fixieren
- Zur Fixierung muss die entsprechende Variable in der Formel mit einem festen Wert multipliziert werden.

Folgende Formel fixiert z.B. alle Ladungen auf 1:

```
Faktor =~ x1 + 1*x2 + 1*x3 + 1*x4
```

Strukturgleichungsmodelle in R

Fixieren von Modellparametern

- Modellparameter können mit 0 fixiert werden, um voreingestellte Berechnungen zu ignorieren
- Beispiel: In einem CFA-Modell wird standardmäßig angenommen, dass alle exogenen latenten Variablen korreliert sind
- Wenn Sie jedoch die Korrelation zwischen einem Paar latenter Variablen auf Null festlegen werden soll, muss dies in der Modellsyntax berücksichtigt werden.
- Folgender Code würde annehmen, dass es zwischen den Faktoren visual/textual und speed keine Korrelation gibt (diese orthogonal sind), aber zwischen visual und textual schon:

```
model = '  
  visual =~ x1 + x2 + x3  
  textual =~ x4 + x5 + x6  
  speed   =~ x7 + x8 + x9  
  
  visual ~~ 0*speed  
  textual ~~ 0*speed  
'
```

Strukturgleichungsmodelle in R

Fixieren von Modellparametern - Equality constraints

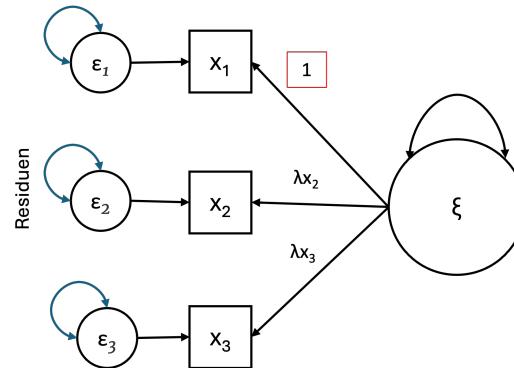
- Equality constraints zwingen einzelne Parameter in einem Modell "gleich zu sein"
- Dies macht Sinn, wenn die Hypothese gleicher Ladungen oder gleicher Effekte zwischen Gruppen getestet werden soll
- Dies wird erreicht, indem für die jeweiligen Variablen ein identischer Multiplikator genutzt wird
- Folgendes Modell nimmt z.B. identische Ladungen für `x2` und `x3` an:

```
model = '
visual  =~ x1 + a*x2 + a*x3
textual =~ x4 + x5 + x6
speed   =~ x7 + x8 + x9
'
```

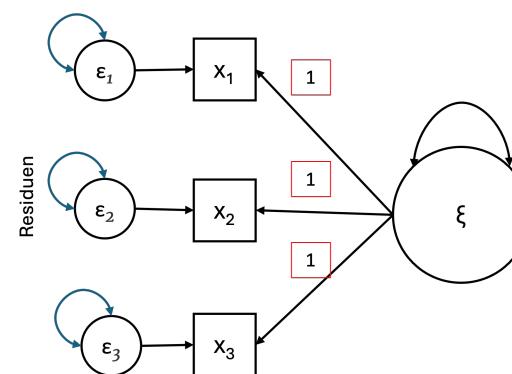
Modellvergleiche

- Modellvergleiche können für geschachtelte Modelle geprüft werden
- Geschachtelt = ein Modell konzeptuell im anderen enthalten (einfaches vs. komplexes Modell)
- Modelle, in denen Parameter fixiert werden sind "einfacher", da weniger freie Werte geschätzt werden müssen

Komplexes Modell
6 zu schätzende Parameter



Einfaches Modell
4 zu schätzende Parameter



Modellvergleiche

- Geschachtelte Modelle können mittels **χ^2 -likelihood-ratio-test** verglichen werden.
- Teststatistik des Modellvergleichs: $\Delta\chi^2 = \chi^2_{\text{einfach}} - \chi^2_{\text{komplex}}$
- Freiheitsgrade des Modellvergleichs: $\Delta df = df_{\text{einfach}} - df_{\text{komplex}}$
- Einfacheres Modell hat immer einen schlechteren Modellfit als ein komplexeres Modell. (Zu prüfen: Ist es signifikant schlechter?)

Interpretation:

- Signifikanter Modellvergleich: Das einfache Modell ist signifikant schlechter als das komplexe Modell → komplexes Modell wählen.
- Nicht signifikanter Modellvergleich: Das einfache Modell ist nicht signifikant schlechter als das komplexe Modell → einfaches Modell wählen.

Modellvergleiche

- Beispiel: 2 Modelle werden angepasst
 - komplexes Modell: alle Parameter frei schätzen (außer die der ersten Indikatoren)
 - einfaches Modell: Ladungen für x_2 und x_3 sind fixiert.

```
komplex = '
visual  =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
speed   =~ x7 + x8 + x9
'

einfach = '
visual  =~ x1 + 1*x2 + 1*x3
textual =~ x4 + x5 + x6
speed   =~ x7 + x8 + x9
'
```

Modellvergleiche

```
fiteinfach = cfa(einfach, data = HolzingerSwineford1939)
fitkomplex = cfa(komplex, data = HolzingerSwineford1939)

anova(fiteinfach, fitkomplex)

## 
## Chi-Squared Difference Test
##
##          Df      AIC      BIC  Chisq Chisq diff    RMSEA Df diff Pr(>Chisq)
## fitkomplex 24 7517.5 7595.3 85.305
## fiteinfach 26 7524.4 7594.9 96.253      10.948 0.12192      2  0.004195 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Komplexes Modell hat zwei freie Parameter mehr als das einfache Modell / Einfaches Modell hat zwei zu schätzende Parameter weniger
- Modellvergleich ist signifikant → komplexes Modell hat signifikant bessere Modellpassung

Modellvergleiche

- Sind Modelle nicht geschachtelt, können keine direkten Modellvergleiche gerechnet werden.
- Modellpassung muss anhand relativen Modelfits verglichen werden
- Hierfür wird oft das Akaike Information Criterion herangezogen (AIC)
- AIC-Interpretation: Das Modell mit dem kleineren AIC Wert ist zu bevorzugen.

```
anova(fiteinfach, fitkomplex) # Angenommen Modelle wären nicht geschachtelt
```

```
##  
## Chi-Squared Difference Test  
##  
##          Df      AIC      BIC  Chisq Chisq diff    RMSEA Df diff Pr(>Chisq)  
## fitkomplex 24 7517.5 7595.3 85.305  
## fiteinfach 26 7524.4 7594.9 96.253      10.948 0.12192      2  0.004195 **  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Kategoriale Daten

- Folgende Variablenarten werden als kategorial (nicht kontinuierlich) betrachtet:
 - Binäre Variablen
 - ordinale Variablen
 - nominale Variablen
- Wichtig: Es macht einen großen Unterschied, ob diese kategorialen Variablen im Modell exogen (unabhängig) oder endogen (abhängig) sind.
 - Exogene kategoriale Variablen sind relativ unproblematisch zu modellieren
 - Endogene kategoriale Variablen sind herausfordernder in der Modellierung

Kategoriale Daten

Exogene kategoriale Variablen

- Binäre exogene Variable (z.B. Geschlecht)
 - Muss als Dummy-Variable (0/1) codiert werden, genauso wie in einem klassischen Regressionsmodell.
- Exogenen ordinale Variable
 - Es kann ein Codierungsschema verwendet werden, das die Reihenfolge widerspiegelt (z. B. 1, 2, 3, ...)
 - Variable kann wie jede andere numerische Variable behandelt werden.
- nominale exogene Variable mit $K > 2$ Kategorien
 - muss durch ein Set von $K - 1$ Dummy-Variablen ersetzt werden

Kategoriale Daten

Endogene kategoriale Variablen

- `lavaan` kann derzeit mit binären und ordinalen (aber nicht nominalen) endogenen Variablen umgehen
- Dafür müssen kategoriale Variablen als `ordered` definiert werden
- Das `ordered`-Argument kann beim Fitting des Modells (cfa/sem/growth/lavaan) verwendet werden.
- Wenn Sie z.B. 3 binäre/ordinale Variablen haben (z. B. item1, item2, item3):

```
fit <- cfa(model, data = df, ordered = c("item1", "item2", "item3"))
```

- Wenn alle (endogenen) Variablen als kategorial behandelt werden sollen, kann pauschal `ordered = TRUE` verwendet werden.

```
fit <- cfa(model, data = df, ordered = TRUE)
```

- Wenn das `ordered = TRUE` Argument verwendet wird, wechselt `lavaan` automatisch auf den `WLSMV` Schätzalgorithmus.

Fehlende Werte

- Wenn die Daten fehlende Werte enthalten, ist die Standardvorgehensweise die listwise deletion (Fallweises Löschen).
- `lavaan` bietet die Möglichkeit der fallweisen Maximum-Likelihood-Schätzung (full information likelihood, FIML).
- Voraussetzung: Werte fehlen MCAR (missing completely at random) oder MAR (missing at random)
- FIML-Schätzung kann aktiviert werden, indem das Argument `missing = "ML"` beim Aufruf der Anpassungsfunktion verwendet wird.

```
fit <- cfa(model, data = df, missing = "ML")
```

Robuste Standardfehler und Bootstrapping

Robuste Standardfehler

- Robuste Standardfehler können explizit mit dem Argument `se = "robust"` angefordert werden.
- Ebenso können robuste Teststatistiken explizit mit `test = "robust"` angefordert werden.
- Sinnvoll z.B. bei Verteilungsverletzungen o.ä.

Bootstrapping

- Es kann entweder `se = "bootstrap"` oder `test = "bootstrap"` beim Anpassen des Modells verwendet werden
- Dadurch werden Bootstrap-Standardfehler bzw. ein auf Bootstrap basierender p-Wert ausgegeben
- Sinnvoll bei nicht-normalverteilten Daten, kleinen Stichproben, komplexen Modellen oder unsicheren Modellannahmen
- Bootstrapping kann rechenintensiv sein, insbesondere bei großen Datensätzen

Vergleich implementierter Schätzmethoden

- Maximum-Likelihood-Methode: Beantwortet die Frage, „Welche Parameter in der Population sind am wahrscheinlichsten, gegeben die beobachteten Daten?“ → liefert eine optimale Schätzung der Parameter der Population.
- GLS- und ULS-Methode: Minimieren die quadrierten Differenzen zwischen der beobachteten und der modellimplizierten Kovarianzmatrix.
- ADF-Methode: Verwendet eine spezielle Kovarianzmatrix und basiert auf der Generalized Least Squares (GLS)-Schätzung.
 - benötigt große Stichproben ($N > 3000$ empfohlen) und ist nur bei wenig komplexen Modellen gut
 - Vorteil: macht keine Verteilungsannahmen
- WLSMV-Methode: Beruht auf einer gewichteten Schätzung der kleinsten quadratischen Abweichungen und berücksichtigt dabei Anpassungen für das arithmetische Mittel und die Varianz.
 - Vorteil: ermöglicht eine robuste Schätzung von Modellen mit dichotomen Variablen

Take-aways

- Die Güte der Modellpassung eines SEMs kann mittels Fit-Indizes angegeben werden.
- Fit-Indizes werden in exakte bzw. approximative Maße unterteilt - approximative Maße weiterhin in relative und absolute Maße unterteilt.
- Die Güte eines Indizes wird anhand gängiger Cutoffs bewertet, welche in der Literatur jedoch auch Kritik unterliegen.
- Mit fehlenden Werten kann in SEMs mittels FIML-Schätzung umgegangen werden.
- Konkurrierende Modelle können mittels geschachtelten Modelltests oder durch den Abgleich relativer Indizes des Modellfits (z.B. AIC) verglichen werden
- `lavaan` kann im Falle von Voraussetzungsverletzungen robuste Standardfehler berechnen und Bootstrapping durchführen.