Musterlösung: Längsschnittanalaysen und Veränderungsmessung I

Aufgabe 1 (Variabilitätsmodell)

Als Vorbereitung müssen das `lme4`-Paket und der `affect`-Datensatz geladen werden.

```
# setwd("myworkindirectory")
load("affect.rda")
library(lme4)
```

Überprüfen Sie mit einer ANOVA die statistische Bedeutsamkeit der Clusterung für die gute vs. schlechte Stimmung (Variable `gut`):

```
anova(lm(gut ~ ID, affect))
```

```
Analysis of Variance Table

Response: gut
           Df Sum Sq Mean Sq F value    Pr(>F)
ID         72 471.64  6.5505  9.2747 < 2.2e-16
Residuals 724 511.35  0.7063
```

Schätzen Sie das entsprechende Nullmodell mit der `lmer()`-Funktion:

```
null <-  lmer(gut ~ 1 + (1 | ID), affect)
summary(null)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: gut ~ 1 + (1 | ID)
   Data: affect

REML criterion at convergence: 2152.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.5945 -0.3964  0.1647  0.5237  3.1403

Random effects:
 Groups   Name        Variance Std.Dev.
 ID       (Intercept) 0.5642   0.7511
```

```
 Residual             0.7068   0.8407
Number of obs: 797, groups:  ID, 73

Fixed effects:
            Estimate Std. Error t value
(Intercept)  4.93391    0.09292    53.1
```

Berechnen Sie die ICC aus den Varianzkomponenten:

```r
var.u0i <- unlist(VarCorr(null))
var.rti <- sigma(null)^2
var.u0i / (var.u0i + var.rti)
```

```
       ID
0.4438969
```

Aufgabe 2 (Growth Curve Mehrebenenmodell)

Erstellen Sie ein Modell, in dem gute Stimmung linear von der Tageszeit vorhergesagt wird. Das Ausgangsniveau der Stimmung darf dabei über Personen variieren. Gibt es einen (von Null verschiedenen) mittleren linearen Trend der guten Stimmung über alle Personen hinweg?

```r
line <-  lmer(gut ~ 1 + time + (1 | ID), affect)
summary(line)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: gut ~ 1 + time + (1 | ID)
   Data: affect

REML criterion at convergence: 2145.6

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-3.4190 -0.3941  0.1414  0.5482  3.1185 

Random effects:
 Groups   Name        Variance Std.Dev.
 ID       (Intercept) 0.5670   0.7530  
 Residual             0.6935   0.8328  
```

```
Number of obs: 797, groups:  ID, 73

Fixed effects:
            Estimate Std. Error t value
(Intercept) 4.748403   0.104844  45.290
time        0.032136   0.008374   3.838

Correlation of Fixed Effects:
     (Intr)
time -0.461
```

Linearer Trend von Null verschieden?

```
test <- lmerTest::lmer(gut ~ 1 + time + (1 | ID), affect)
summary(test)$coef
```

```
              Estimate  Std. Error       df   t value     Pr(>|t|)
(Intercept) 4.74840323 0.104844370 113.4268 45.290016 1.795806e-74
time        0.03213646 0.008374306 725.3395  3.837507 1.351244e-04
```

```
confint(line, method = "boot", n = 1000)
```

```
                 2.5 %    97.5 %
.sig01      0.60488557 0.8918273
.sigma      0.79170062 0.8794689
(Intercept) 4.54078952 4.9573825
time        0.01542015 0.0488411
```

Ja, $t(725.3) = 3.838$, $p < 0.05$. 95% CI von `time` überdeckt nicht die Null (CI-Grenzen können bei Ihnen aufgrund des Zufallsprozesses leicht abweichen).

Erweitern Sie das lineare Modell für die gute Stimmung, um zu untersuchen, ob Varianz in den linearen Steigunskoeffizienten vorhanden ist.

```
line_rs <-  lmer(gut ~ 1 + time  + (1 + time | ID), affect)
summary(line_rs)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: gut ~ 1 + time + (1 + time | ID)
   Data: affect
```

```
REML criterion at convergence: 2104.1

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-3.4596 -0.4006  0.1561  0.5281  3.0919 

Random effects:
 Groups   Name        Variance Std.Dev. Corr 
 ID       (Intercept) 1.075418 1.03702       
          time        0.007061 0.08403  -0.73
 Residual             0.600454 0.77489       
Number of obs: 797, groups:  ID, 73

Fixed effects:
            Estimate Std. Error t value
(Intercept)  4.74503    0.13274  35.747
time         0.03244    0.01261   2.572

Correlation of Fixed Effects:
     (Intr)
time -0.737
```

```
confint(line_rs, method = "boot", n = 1000)
```

```
                   2.5 %      97.5 %
.sig01       0.831494081  1.23101902
.sig02      -0.856989784 -0.54569231
.sig03       0.061484880  0.10516362
.sigma       0.734729686  0.81779379
(Intercept)  4.503874961  5.00459077
time         0.007147791  0.05759711
```

```
anova(line_rs, line, refit = FALSE)
```

```
Data: affect
Models:
line: gut ~ 1 + time + (1 | ID)
line_rs: gut ~ 1 + time + (1 + time | ID)
        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
```

```
line       4 2153.6 2172.4 -1072.8   2145.6
line_rs    6 2116.1 2144.2 -1052.0   2104.1 41.539  2   9.55e-10
```

Ja, die Standardabweichung der random slope beträgt 0.08403 mit einem CI welches nicht die Null überdeckt (Grenzen können leicht variieren). Der LRT zeigt an, dass das restriktivere Modell (ohne random slopes) signifikant schlechter auf die Daten passt als das Modell mit random slopes.

Wie sind die random effects miteinander korreliert? Was bedeutet diese Korrelation inhaltlich?

```
summary(line_rs)$var
```

```
 Groups   Name        Std.Dev. Corr
 ID       (Intercept) 1.037024
          time        0.084032 -0.731
 Residual             0.774890
```

Korrelation von -0.73 -> Höhere Intercepts gehen mit niedrigeren Steigungen einher

Aufgabe 3 (Latent Growth Curve Modell)

Betrachten Sie nochmals die Übungsaufgaben zur Sitzung 13 SEM des Wintersemesters. Laden Sie den Datensatz `stimmung.rda` (Achtung: Objektname auch `affect` so wie bei dem anderen Datensatz).

```
load("stimmung.rda")
```

```
str(affect)
```

```
'data.frame':   249 obs. of  14 variables:
 $ SW1_1  : int  2 0 2 9 1 0 4 0 2 4 ...
 $ SW2_1  : int  4 5 1 0 1 5 3 0 2 8 ...
 $ SW3_1  : int  4 6 7 0 1 4 6 0 2 6 ...
 $ SW4_1  : int  4 0 2 0 2 0 3 0 2 0 ...
 $ stim1_1: int  4 0 3 1 0 2 2 0 2 3 ...
 $ stim6_1: int  5 0 4 4 2 0 3 0 1 4 ...
 $ stim2_1: int  3 0 2 0 4 3 1 0 1 3 ...
 $ stim5_1: int  3 0 4 0 3 2 1 0 1 3 ...
 $ stim1_2: int  2 3 3 2 0 2 2 0 1 2 ...
```

```
 $ stim6_2: int  6 3 3 2 0 3 2 0 2 4 ...
 $ stim1_3: int  4 3 4 1 0 2 2 1 2 3 ...
 $ stim6_3: int  4 3 2 0 0 2 2 2 2 4 ...
 $ stim1_4: int  3 0 3 2 1 3 2 1 2 4 ...
 $ stim6_4: int  4 0 2 2 0 3 2 2 2 4 ...
```

Spezifizeren Sie ein Single-Indicator Latent Growth Curve Modell für die Stimmungsvariablen stim1. Lassen Sie die Erwartungswerte des Intercept und des Slope Faktors frei schätzen und setzen Sie alle Intercepts der Items auf Null.

```r
lgc1 <- "
Int =~ 1*stim1_1 + 1*stim1_2 + 1*stim1_3 + 1*stim1_4
Slo =~ 0*stim1_1 + 1*stim1_2 + 2*stim1_3 + 3*stim1_4

Int ~ NA*1
Slo ~ NA*1

stim1_1 ~ 0*1
stim1_2 ~ 0*1
stim1_3 ~ 0*1
stim1_4 ~ 0*1
"
```

```r
library(lavaan)
lgc_fit1 <- sem(lgc1, data = affect, meanstructure = TRUE)
```

Betrachten Sie den Fit des Modells und interpretieren Sie die Modellergebnisse.

```r
fitmeasures(lgc_fit1)[c("chisq", "df", "pvalue", "srmr", "rmsea", "cfi", "tli")]
```

```
    chisq        df    pvalue      srmr     rmsea       cfi       tli 
1.2667764 5.0000000 0.9383129 0.0156615 0.0000000 1.0000000 1.0089991 
```

```r
summary(lgc_fit1, standardize = TRUE)
```

```
lavaan 0.6.15 ended normally after 28 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         9
```

```
  Number of observations                           249

Model Test User Model:

  Test statistic                                 1.267
  Degrees of freedom                                 5
  P-value (Chi-square)                           0.938

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
  Int =~
    stim1_1           1.000                               1.598    0.827
    stim1_2           1.000                               1.598    0.854
    stim1_3           1.000                               1.598    0.836
    stim1_4           1.000                               1.598    0.853
  Slo =~
    stim1_1           0.000                               0.000    0.000
    stim1_2           1.000                               0.269    0.144
    stim1_3           2.000                               0.539    0.282
    stim1_4           3.000                               0.808    0.431

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Int ~~
    Slo              -0.126    0.083   -1.513    0.130   -0.292   -0.292

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    Int               3.165    0.116   27.219    0.000    1.981    1.981
    Slo               0.148    0.035    4.275    0.000    0.549    0.549
   .stim1_1           0.000                               0.000    0.000
   .stim1_2           0.000                               0.000    0.000
   .stim1_3           0.000                               0.000    0.000
   .stim1_4           0.000                               0.000    0.000

Variances:
```

```
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .stim1_1           1.181    0.198    5.960    0.000    1.181    0.316
   .stim1_2           1.126    0.136    8.274    0.000    1.126    0.322
   .stim1_3           1.316    0.150    8.747    0.000    1.316    0.360
   .stim1_4           1.062    0.194    5.482    0.000    1.062    0.302
    Int               2.554    0.317    8.062    0.000    1.000    1.000
    Slo               0.073    0.039    1.852    0.064    1.000    1.000
```

Das Modell passt sehr gut auf die Daten. Zum ersten Zeitpunkt zeigen die Personen eine mittlere Stimmung von 3.165, welche über die Zeit hinweg im Mittel ansteigt (0.148 Einheiten mit jedem Messzeitpunkt (ca. eine Stunde)). Es bestehen inter-individuelle Unterschiede im Ausgangswert (Varianz des Intercept Faktors). Die Varianz des Slope Faktors ist nicht signifikant, sodass inter-individuelle Unterschiede im Wachstum ggf. nicht modelliert werden müssten. Intercept und Slope korrelieren zu -0.292 (ein höherer Ausgangswert geht einher mit geringerem Wachstum), jedoch nicht signifikant. Wir betrachten aufgrund der Ergebnisse noch ein reines Variabilitätsmodell:

Modell ohne Slope: Latent State Trait Modell (single-Indicator)

```r
lst1 <- "
Int =~ 1*stim1_1 + 1*stim1_2 + 1*stim1_3 + 1*stim1_4
"
```

```r
lst_fit <- sem(lst1, data = affect, meanstructure = TRUE)
fitmeasures(lst_fit)[c("chisq", "df", "pvalue", "srmr", "rmsea", "cfi", "tli")]
```

```
     chisq         df     pvalue       srmr      rmsea        cfi        tli 
4.38367002 5.00000000 0.49559718 0.01587412 0.00000000 1.00000000 1.00148570 
```

```r
summary(lst_fit, standardize = TRUE)
```

```
lavaan 0.6.15 ended normally after 19 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         9

  Number of observations                           249

Model Test User Model:
```

```
  Test statistic                                 4.384
  Degrees of freedom                                 5
  P-value (Chi-square)                           0.496

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
  Int =~
    stim1_1           1.000                               1.519    0.785
    stim1_2           1.000                               1.519    0.819
    stim1_3           1.000                               1.519    0.799
    stim1_4           1.000                               1.519    0.800

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .stim1_1           3.137    0.123   25.598    0.000    3.137    1.622
   .stim1_2           3.361    0.117   28.616    0.000    3.361    1.813
   .stim1_3           3.446    0.121   28.594    0.000    3.446    1.812
   .stim1_4           3.602    0.120   29.957    0.000    3.602    1.898
    Int               0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .stim1_1           1.432    0.161    8.902    0.000    1.432    0.383
   .stim1_2           1.129    0.136    8.286    0.000    1.129    0.329
   .stim1_3           1.310    0.151    8.686    0.000    1.310    0.362
   .stim1_4           1.294    0.150    8.656    0.000    1.294    0.359
    Int               2.306    0.236    9.772    0.000    1.000    1.000
```

- Das LST Modell mit einem zeitstabilen Faktor und freien Item-Intercepts über die Zeit passt sehr gut auf die Daten.
- Growth Faktor wird nicht benötigt (keine inter-individuellen Unterschiede im Wachstum), aber Mittelwerte der Stimmung steigen über die Zeit hinweg leicht an (–> Item Intercepts; bzw. mittlerer Growth))

Zusatzaufgabe 1 LGC (optional)

Spezifizieren Sie das Latent State Modell für die Stimmung (Items stim1 und stim6) (ohne indikatorspezifischem Residualfaktor) aus dem Skript der Übung 13 (SEM) aus dem Wintersemester (Code siehe Unterlagen des WS –> copy pasten)

```
ls <- "
stim1 =~ 1*stim1_1 + lam*stim6_1
stim2 =~ 1*stim1_2 + lam*stim6_2
stim3 =~ 1*stim1_3 + lam*stim6_3
stim4 =~ 1*stim1_4 + lam*stim6_4
"
```

Erweitern Sie nun das Modell so, dass Sie ein lineares Growth Modell für die *latenten* Variablen `stim1` - `stim4` erstellen. Achten Sie darauf, dass Sie die Intercepts für die Items `stim1_t` auf Null fixieren und die der Items `stim6_t` über die Zeit konstant setzen. Schätzen Sie die Erwartungswerte des Latent Intercept Faktors und des Latent Slope Faktors frei.

```
lgc2 <- "
stim1 =~ 1*stim1_1 + lam*stim6_1
stim2 =~ 1*stim1_2 + lam*stim6_2
stim3 =~ 1*stim1_3 + lam*stim6_3
stim4 =~ 1*stim1_4 + lam*stim6_4

Int =~ 1*stim1 + 1*stim2 + 1*stim3 + 1*stim4
Slo =~ 0*stim1 + 1*stim2 + 2*stim3 + 3*stim4

stim1_1 ~ 0*1
stim1_2 ~ 0*1
stim1_3 ~ 0*1
stim1_4 ~ 0*1
stim6_1 ~ b*1
stim6_2 ~ b*1
stim6_3 ~ b*1
stim6_4 ~ b*1

Int ~ NA*1
Slo ~ NA*1
"

lgc_fit2 <- sem(lgc2, data = affect, meanstructure = TRUE)
fitmeasures(lgc_fit2)[c("chisq", "df", "pvalue", "srmr", "rmsea", "cfi", "tli")]
```

```
      chisq           df       pvalue         srmr        rmsea          cfi 
1.071353e+02 2.500000e+01 3.839928e-12 4.052410e-02 1.148670e-01 9.385335e-01 
        tli 
9.311575e-01 
```

```r
summary(lgc_fit2, standardize = TRUE)
```

```
lavaan 0.6.15 ended normally after 46 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        25
  Number of equality constraints                     6

  Number of observations                           249

Model Test User Model:
                                                      
  Test statistic                               107.135
  Degrees of freedom                                25
  P-value (Chi-square)                           0.000

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
  stim1 =~                                                              
    stim1_1           1.000                               1.807    0.932
    stim6_1   (lam)   0.839    0.030   28.083    0.000    1.517    0.784
  stim2 =~                                                              
    stim1_2           1.000                               1.714    0.922
    stim6_2   (lam)   0.839    0.030   28.083    0.000    1.438    0.765
  stim3 =~                                                              
    stim1_3           1.000                               1.756    0.920
    stim6_3   (lam)   0.839    0.030   28.083    0.000    1.473    0.777
  stim4 =~                                                              
    stim1_4           1.000                               1.750    0.929
```

```
    stim6_4  (lam)     0.839    0.030   28.083    0.000    1.469    0.793
  Int =~
    stim1              1.000                               0.881    0.881
    stim2              1.000                               0.929    0.929
    stim3              1.000                               0.907    0.907
    stim4              1.000                               0.910    0.910
  Slo =~
    stim1              0.000                               0.000    0.000
    stim2              1.000                               0.162    0.162
    stim3              2.000                               0.316    0.316
    stim4              3.000                               0.475    0.475

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Int ~~
    Slo               -0.135    0.080   -1.677    0.094   -0.305   -0.305

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .stim1_1            0.000                               0.000    0.000
   .stim1_2            0.000                               0.000    0.000
   .stim1_3            0.000                               0.000    0.000
   .stim1_4            0.000                               0.000    0.000
   .stim6_1    (b)     0.720    0.110    6.554    0.000    0.720    0.372
   .stim6_2    (b)     0.720    0.110    6.554    0.000    0.720    0.383
   .stim6_3    (b)     0.720    0.110    6.554    0.000    0.720    0.380
   .stim6_4    (b)     0.720    0.110    6.554    0.000    0.720    0.388
    Int                3.172    0.116   27.443    0.000    1.991    1.991
    Slo                0.144    0.034    4.213    0.000    0.521    0.521
   .stim1              0.000                               0.000    0.000
   .stim2              0.000                               0.000    0.000
   .stim3              0.000                               0.000    0.000
   .stim4              0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .stim1_1            0.495    0.146    3.384    0.001    0.495    0.132
   .stim6_1            1.441    0.162    8.883    0.000    1.441    0.385
   .stim1_2            0.517    0.132    3.922    0.000    0.517    0.150
   .stim6_2            1.462    0.157    9.297    0.000    1.462    0.414
   .stim1_3            0.556    0.141    3.943    0.000    0.556    0.153
   .stim6_3            1.423    0.158    9.016    0.000    1.423    0.396
   .stim1_4            0.487    0.136    3.590    0.000    0.487    0.137
```

```
.stim6_4          1.274    0.146    8.747    0.000    1.274    0.371
.stim1            0.730    0.210    3.472    0.001    0.223    0.223
.stim2            0.592    0.151    3.929    0.000    0.202    0.202
.stim3            0.776    0.165    4.698    0.000    0.252    0.252
.stim4            0.641    0.203    3.156    0.002    0.209    0.209
 Int              2.537    0.311    8.164    0.000    1.000    1.000
 Slo              0.077    0.037    2.051    0.040    1.000    1.000
```

Die Ergebnisse des Growth beziehen sich nun auf messfehlerfreie latente Variablen. Wir erhalten zudem Schätzwerte für die Messfehlervarianzen. Die mittleren Werte des latenten Intercept und Growth sowie auch ihre Varianzen sind ähnlich wie beim single-Indikator Modell (s. oben), die Varianz des random Slopes ist nun sign. Das Modell passt nicht gut auf die Daten, was sich wieder, wie im Skript des WS bereits festgestellt und ausgeführt, durch indikator-spezifische stabile Effekte erklären lässt. Wir können auch hier den indikatorspezifischen Residualfaktor für das zweite Item wieder hinzunehmen. Wir müssen darauf achten, dass dieser Faktor (ITR im Modell) nicht mit den anderen Faktoren im Modell korrelieren darf.

Zusatzaufgabe 2 LGC (optional)

```
lgc3 <- "
stim1 =~ 1*stim1_1 + lam*stim6_1
stim2 =~ 1*stim1_2 + lam*stim6_2
stim3 =~ 1*stim1_3 + lam*stim6_3
stim4 =~ 1*stim1_4 + lam*stim6_4

Int =~ 1*stim1 + 1*stim2 + 1*stim3 + 1*stim4
Slo =~ 0*stim1 + 1*stim2 + 2*stim3 + 3*stim4

stim1_1 ~ 0*1
stim1_2 ~ 0*1
stim1_3 ~ 0*1
stim1_4 ~ 0*1
stim6_1 ~ b*1
stim6_2 ~ b*1
stim6_3 ~ b*1
stim6_4 ~ b*1

Int ~ NA*1
Slo ~ NA*1

ITR =~ 1*stim6_1 + 1*stim6_2 + 1*stim6_3 + 1*stim6_4
```

```r
ITR ~~ 0*stim1 + 0*stim2 + 0*stim3 + 0*stim4 + 0*Int + 0*Slo
"

lgc_fit3 <- sem(lgc3, data = affect, meanstructure = TRUE)
fitmeasures(lgc_fit3)[c("chisq", "df", "pvalue", "srmr", "rmsea", "cfi", "tli")]
```

```
      chisq          df      pvalue        srmr       rmsea         cfi 
10.17622325 24.00000000  0.99378134  0.01929373  0.00000000  1.00000000 
        tli 
 1.01206931 
```