Structural Equation Modeling

P.12 - Growth Curve Analysis

01.12.2021

Lab Description

In this assignment you are going to learn how to estimate a *Latent Growth Curve* (LGC) model in `R` using the `lavaan`. In the previous lab you learn how you can use hierarchical models (i.e., multi-level models for nested data). In this lab, we are are still going to work with nested data (i.e., observations nested under participants), but this time we are going to approach the problem from the perspective of *Structural Equation Models* (SEM) For this practical you will need the following packages: `lavaan` and `semPlot`. You can install and load these packages using the following code:

```
# Install packages.
install.packages(c("lavaan", "semPlot"))

# Load the packages.
library(lavaan)
library(semPlot)
```

Exercise 1

In this exercise, you are going to estimate a LGC model similar as the one depicted in *Figure 1*, but with six waves instead of four. Specifically, you are going to investigate changes in body weight over 12 years (i.e., six waves of data separated by two years each) using the health and aging data set `health.dat` with $N = 5335$. Body weight was operationalized as the *Body Mass Index* (BMI), which is a ratio of weight to square of height (i.e., kg/m^2). The data set `health.dat` is available on Canvas in the module corresponding to this lab. You can use the code below to load the data and set the variables names.

Set the working directory to the location where your data file has been downloaded and load the data.

```
# For example.
setwd("/Users/mihai/Downloads")

# Load data.
data <- read.table("health.dat")

# Inspect the data.
```

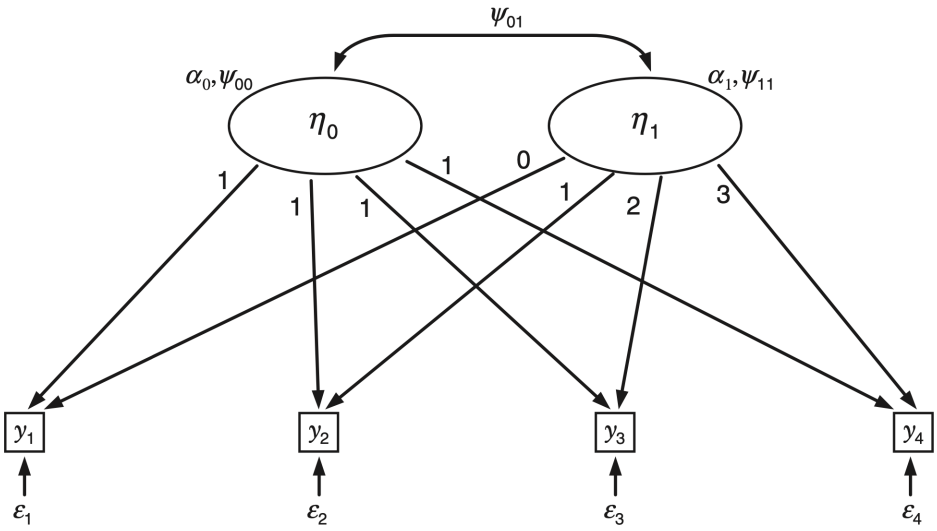


Figure 1: Example of a LGT model reproduced from Newsom (2015, p. 174).

```
View(data)
```

Set the variable names.

```
# Variable names.
variable_names_ex_1 = c(
    "age", "srh1", "srh2", "srh3", "srh4", "srh5", "srh6", "bmi1",
    "bmi2", "bmi3", "bmi4", "bmi5", "bmi6", "cesdna1", "cesdpa1", "cesdso1",
    "cesdna2", "cesdpa2", "cesdso2", "cesdna3", "cesdpa3", "cesdso3",
    "cesdna4", "cesdpa4", "cesdso4", "cesdna5", "cesdpa5", "cesdso5",
    "cesdna6", "cesdpa6", "cesdso6", "diab1", "diab2", "diab3 ", "diab4", "diab5", "diab6"
)

# Set the names.
names(data) <- variable_names_ex_1
```

List the variables.

```
# List variables.
str(data)
```

Specify which fit measures we are interested in:

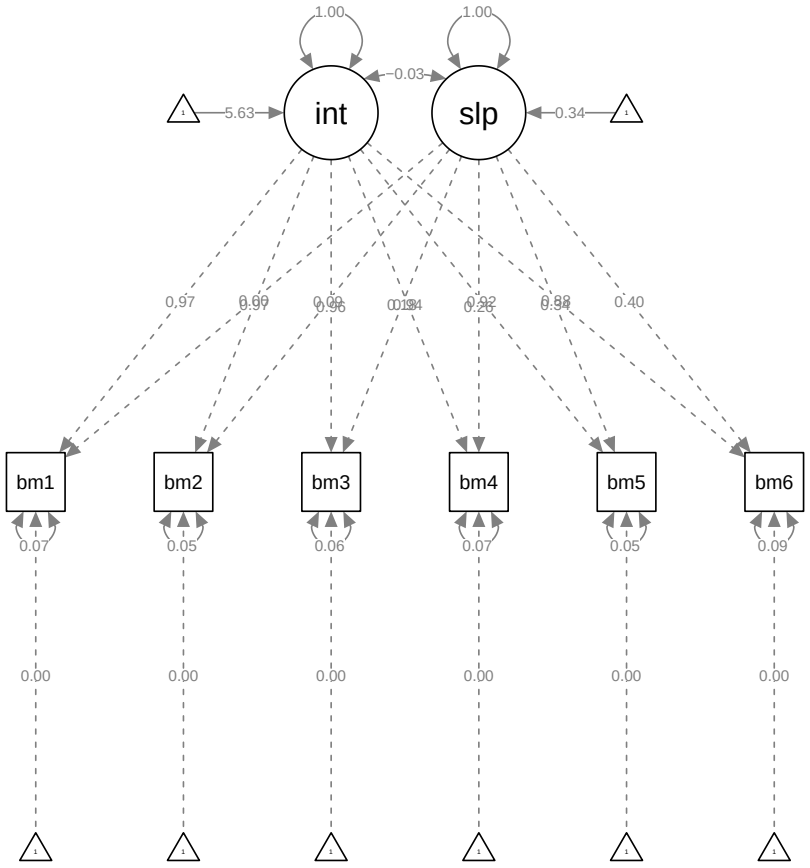
```
# Fit indices to print.
fit_indices <- c("chisq", "df", "pvalue", "cfi", "tli", "rmsea", "rmsea.pvalue", "srmr")
```

a. Estimate a *LGC* model in which you set each intercept factor loading equal to 1 and the slope factor loadings equal to 0, 1, 2, 3, 4, and 5. Do not include correlated measurement residuals in this model. Evaluate the fit of this model, and interpret the mean of the latent intercept and mean of the latent slope.

It is important to note that instead of functions `sem` *and* `cfa`*, this time we use the function* **`growth`** *for fitting*

the models.

```
# Model syntax.
model_ex_1_a <- "
    # Latent intercept variable.
    int =~ 1 * bmi1 + 1 * bmi2 + 1 * bmi3 + 1 * bmi4 + 1 * bmi5 + 1 * bmi6

    # Latent slope variable.
    slp =~ 0 * bmi1 + 1 * bmi2 + 2 * bmi3 + 3 * bmi4 + 4 * bmi5 + 5 * bmi6

    # Latent variances and covariances.
    int ~~ int
    slp ~~ slp
    int ~~ slp
"

# Fit model.
model_ex_1_a_fit <- growth(model_ex_1_a, data = data)

# Visualize the model.
semPaths(model_ex_1_a_fit, what = "paths", whatLabels = "std")
```




```
# Model summary.
summary(model_ex_1_a_fit, standardized = TRUE, rsquare = TRUE)
```

```
## lavaan 0.6-9 ended normally after 109 iterations
##
```

```
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        11
##
##   Number of observations                          5335
##
## Model Test User Model:
##
##   Test statistic                               623.877
##   Degrees of freedom                                16
##   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
##   int =~
##     bmi1             1.000                               4.835    0.967
##     bmi2             1.000                               4.835    0.970
##     bmi3             1.000                               4.835    0.957
##     bmi4             1.000                               4.835    0.937
##     bmi5             1.000                               4.835    0.921
##     bmi6             1.000                               4.835    0.877
##   slp =~
##     bmi1             0.000                               0.000    0.000
##     bmi2             1.000                               0.444    0.089
##     bmi3             2.000                               0.888    0.176
##     bmi4             3.000                               1.332    0.258
##     bmi5             4.000                               1.776    0.338
##     bmi6             5.000                               2.220    0.403
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~~
##     slp             -0.057    0.037   -1.512    0.130   -0.026   -0.026
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .bmi1              0.000                               0.000    0.000
##    .bmi2              0.000                               0.000    0.000
##    .bmi3              0.000                               0.000    0.000
##    .bmi4              0.000                               0.000    0.000
##    .bmi5              0.000                               0.000    0.000
##    .bmi6              0.000                               0.000    0.000
##     int              27.211    0.067  403.933    0.000    5.628    5.628
##     slp               0.150    0.008   19.884    0.000    0.337    0.337
##
```

```
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     int              23.377    0.469   49.846    0.000    1.000    1.000
##     slp               0.197    0.006   32.627    0.000    1.000    1.000
##    .bmi1              1.632    0.054   30.325    0.000    1.632    0.065
##    .bmi2              1.360    0.038   35.955    0.000    1.360    0.055
##    .bmi3              1.567    0.037   42.142    0.000    1.567    0.061
##    .bmi4              1.823    0.043   42.616    0.000    1.823    0.068
##    .bmi5              1.478    0.043   34.149    0.000    1.478    0.054
##    .bmi6              2.626    0.074   35.390    0.000    2.626    0.086
##
## R-Square:
##                    Estimate
##     bmi1              0.935
##     bmi2              0.945
##     bmi3              0.939
##     bmi4              0.932
##     bmi5              0.946
##     bmi6              0.914
```

```
# Fit measures.
fitMeasures(model_ex_1_a_fit, fit.measures = fit_indices)
```

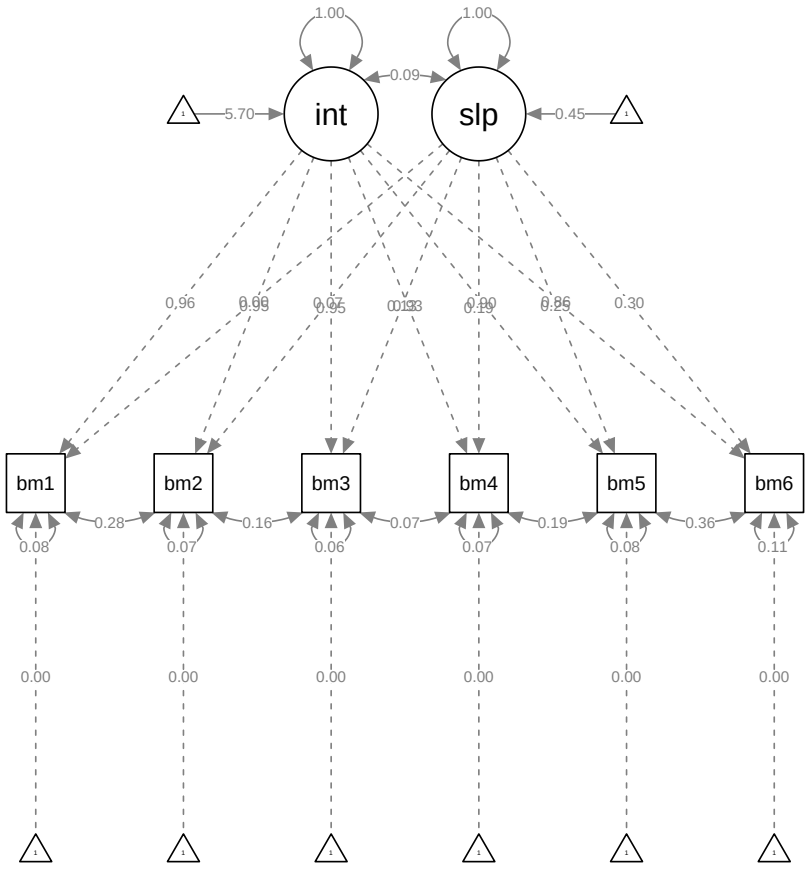
```
##        chisq           df       pvalue          cfi          tli        rmsea rmsea.pvalue
##      623.877       16.000        0.000        0.990        0.990        0.084        0.000
##         srmr
##        0.016
```

b. Re-estimate the model from point *(a)*, but now add auto-correlations among adjacent time points of the measurement residuals (e.g., ε_1 with ε_2, ε_2 with ε_3, and so on). Evaluate the fit of this model and test its fit against the more restricted model estimated at point *(a)*.

We can add the adjacent correlated residuals as follows:

```
# Model syntax.
model_ex_1_b <- "
    # Latent intercept variable.
    int =~ 1 * bmi1 + 1 * bmi2 + 1 * bmi3 + 1 * bmi4 + 1 * bmi5 + 1 * bmi6

    # Latent slope variable.
    slp =~ 0 * bmi1 + 1 * bmi2 + 2 * bmi3 + 3 * bmi4 + 4 * bmi5 + 5 * bmi6

    # Latent variances and covariances.
    int ~~ int
    slp ~~ slp
    int ~~ slp

    # Adjacent correlated residuals.
    bmi1 ~~ bmi2
    bmi2 ~~ bmi3
    bmi3 ~~ bmi4
    bmi4 ~~ bmi5
    bmi5 ~~ bmi6
```

```
"

# Fit model.
model_ex_1_b_fit <- growth(model_ex_1_b, data = data)

# Visualize the model.
semPaths(model_ex_1_b_fit, what = "paths", whatLabels = "std")
```




```
# Model summary.
summary(model_ex_1_b_fit, standardized = TRUE, rsquare = TRUE)
```

```
## lavaan 0.6-9 ended normally after 125 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        16
##
##   Number of observations                          5335
##
## Model Test User Model:
##
##   Test statistic                               187.542
##   Degrees of freedom                                11
##   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
##   int =~
##     bmi1             1.000                               4.773    0.957
##     bmi2             1.000                               4.773    0.953
##     bmi3             1.000                               4.773    0.948
##     bmi4             1.000                               4.773    0.927
##     bmi5             1.000                               4.773    0.901
##     bmi6             1.000                               4.773    0.864
##   slp =~
##     bmi1             0.000                               0.000    0.000
##     bmi2             1.000                               0.332    0.066
##     bmi3             2.000                               0.664    0.132
##     bmi4             3.000                               0.996    0.193
##     bmi5             4.000                               1.328    0.251
##     bmi6             5.000                               1.659    0.300
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~~
##     slp              0.150    0.042    3.579    0.000    0.095    0.095
##  .bmi1 ~~
##    .bmi2             0.552    0.073    7.522    0.000    0.552    0.278
##  .bmi2 ~~
##    .bmi3             0.268    0.035    7.650    0.000    0.268    0.159
##  .bmi3 ~~
##    .bmi4             0.125    0.044    2.852    0.004    0.125    0.075
##  .bmi4 ~~
##    .bmi5             0.384    0.039    9.756    0.000    0.384    0.186
##  .bmi5 ~~
##    .bmi6             1.010    0.088   11.489    0.000    1.010    0.357
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .bmi1             0.000                               0.000    0.000
##    .bmi2             0.000                               0.000    0.000
##    .bmi3             0.000                               0.000    0.000
##    .bmi4             0.000                               0.000    0.000
##    .bmi5             0.000                               0.000    0.000
##    .bmi6             0.000                               0.000    0.000
##     int             27.214    0.067  404.569    0.000    5.702    5.702
##     slp              0.151    0.007   20.357    0.000    0.454    0.454
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     int             22.783    0.471   48.340    0.000    1.000    1.000
```

```
##     slp               0.110    0.010   11.569    0.000    1.000    1.000
##    .bmi1              2.114    0.104   20.288    0.000    2.114    0.085
##    .bmi2              1.869    0.073   25.656    0.000    1.869    0.075
##    .bmi3              1.523    0.057   26.789    0.000    1.523    0.060
##    .bmi4              1.841    0.061   30.121    0.000    1.841    0.069
##    .bmi5              2.308    0.084   27.595    0.000    2.308    0.082
##    .bmi6              3.462    0.131   26.508    0.000    3.462    0.114
##
## R-Square:
##                    Estimate
##     bmi1              0.915
##     bmi2              0.925
##     bmi3              0.940
##     bmi4              0.931
##     bmi5              0.918
##     bmi6              0.886
```

```
# Fit measures.
fitMeasures(model_ex_1_b_fit, fit.measures = fit_indices)
```

```
##        chisq           df       pvalue          cfi          tli        rmsea rmsea.pvalue
##      187.542       11.000        0.000        0.997        0.996        0.055        0.116
##         srmr
##        0.018
```

To compare the fit of `model_ex_1_b` to `model_ex_1_a` we can take a look at the fit measures and also perform a *Likelihood Ratio Test* (LRT).

```
# Put all fit measures in a data frame or convenience.
fit_measures_ex_1 <- data.frame(
    model_ex_1_a = fitMeasures(model_ex_1_a_fit, fit.measures = fit_indices),
    model_ex_1_b = fitMeasures(model_ex_1_b_fit, fit.measures = fit_indices)
)

# Print all fit measures rounded to four decimals.
round(fit_measures_ex_1, 4)
```

```
##              model_ex_1_a model_ex_1_b
## chisq            623.8767     187.5420
## df                16.0000      11.0000
## pvalue             0.0000       0.0000
## cfi                0.9897       0.9970
## tli                0.9903       0.9959
## rmsea              0.0844       0.0548
## rmsea.pvalue       0.0000       0.1163
## srmr               0.0162       0.0180
```

```
# Perform the LRT.
anova(model_ex_1_a_fit, model_ex_1_b_fit)
```

```
## Chi-Squared Difference Test
##
##                  Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## model_ex_1_b_fit 11 136932 137037 187.54
```

```
## model_ex_1_a_fit 16 137358 137430 623.88     436.33       5  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c. In the model estimated at point *(b)*, inspect the estimate for the covariance between the latent intercept and the latent slope. Provide a substantive interpretation for this value.

We can use the function `parTable` to list all parameters.

```
# List parameters.
lavaan::parTable(model_ex_1_b_fit)
```

```
##    id  lhs op  rhs user block group free ustart exo label plabel  start    est    se
## 1   1  int =~ bmi1    1     1     1    0      1   0         .p1.  1.000  1.000 0.000
## 2   2  int =~ bmi2    1     1     1    0      1   0         .p2.  1.000  1.000 0.000
## 3   3  int =~ bmi3    1     1     1    0      1   0         .p3.  1.000  1.000 0.000
## 4   4  int =~ bmi4    1     1     1    0      1   0         .p4.  1.000  1.000 0.000
## 5   5  int =~ bmi5    1     1     1    0      1   0         .p5.  1.000  1.000 0.000
## 6   6  int =~ bmi6    1     1     1    0      1   0         .p6.  1.000  1.000 0.000
## 7   7  slp =~ bmi1    1     1     1    0      0   0         .p7.  0.000  0.000 0.000
## 8   8  slp =~ bmi2    1     1     1    0      1   0         .p8.  1.000  1.000 0.000
## 9   9  slp =~ bmi3    1     1     1    0      2   0         .p9.  2.000  2.000 0.000
## 10 10  slp =~ bmi4    1     1     1    0      3   0        .p10.  3.000  3.000 0.000
## 11 11  slp =~ bmi5    1     1     1    0      4   0        .p11.  4.000  4.000 0.000
## 12 12  slp =~ bmi6    1     1     1    0      5   0        .p12.  5.000  5.000 0.000
## 13 13  int ~~  int    1     1     1    1     NA   0        .p13.  0.050 22.783 0.471
## 14 14  slp ~~  slp    1     1     1    2     NA   0        .p14.  0.050  0.110 0.010
## 15 15  int ~~  slp    1     1     1    3     NA   0        .p15.  0.000  0.150 0.042
## 16 16 bmi1 ~~ bmi2    1     1     1    4     NA   0        .p16.  0.000  0.552 0.073
## 17 17 bmi2 ~~ bmi3    1     1     1    5     NA   0        .p17.  0.000  0.268 0.035
## 18 18 bmi3 ~~ bmi4    1     1     1    6     NA   0        .p18.  0.000  0.125 0.044
## 19 19 bmi4 ~~ bmi5    1     1     1    7     NA   0        .p19.  0.000  0.384 0.039
## 20 20 bmi5 ~~ bmi6    1     1     1    8     NA   0        .p20.  0.000  1.010 0.088
## 21 21 bmi1 ~~ bmi1    0     1     1    9     NA   0        .p21. 12.008  2.114 0.104
## 22 22 bmi2 ~~ bmi2    0     1     1   10     NA   0        .p22. 12.680  1.869 0.073
## 23 23 bmi3 ~~ bmi3    0     1     1   11     NA   0        .p23. 13.010  1.523 0.057
## 24 24 bmi4 ~~ bmi4    0     1     1   12     NA   0        .p24. 13.469  1.841 0.061
## 25 25 bmi5 ~~ bmi5    0     1     1   13     NA   0        .p25. 13.827  2.308 0.084
## 26 26 bmi6 ~~ bmi6    0     1     1   14     NA   0        .p26. 14.718  3.462 0.131
## 27 27 bmi1 ~1         0     1     1    0      0   0        .p27.  0.000  0.000 0.000
## 28 28 bmi2 ~1         0     1     1    0      0   0        .p28.  0.000  0.000 0.000
## 29 29 bmi3 ~1         0     1     1    0      0   0        .p29.  0.000  0.000 0.000
## 30 30 bmi4 ~1         0     1     1    0      0   0        .p30.  0.000  0.000 0.000
## 31 31 bmi5 ~1         0     1     1    0      0   0        .p31.  0.000  0.000 0.000
## 32 32 bmi6 ~1         0     1     1    0      0   0        .p32.  0.000  0.000 0.000
## 33 33  int ~1         0     1     1   15     NA   0        .p33.  0.000 27.214 0.067
## 34 34  slp ~1         0     1     1   16     NA   0        .p34.  0.000  0.151 0.007
```

In our case, the parameter of interest is located at row 15.

```
lavaan::parTable(model_ex_1_b_fit)[15, ]
```

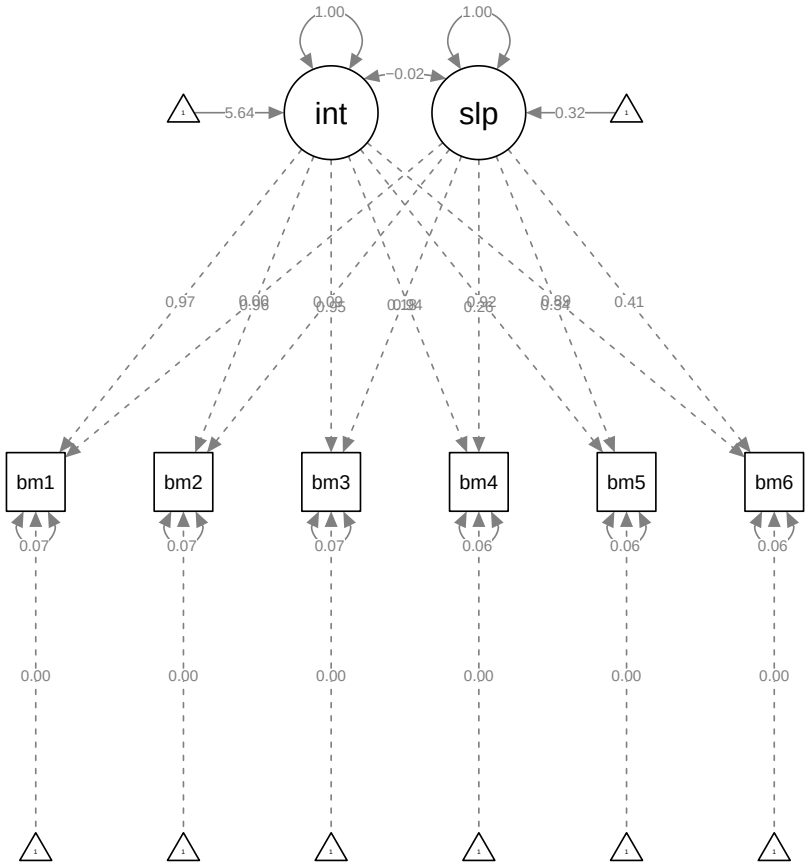
```
##    id lhs op rhs user block group free ustart exo label plabel start  est    se
## 15 15 int ~~ slp    1     1     1    3     NA   0        .p15.     0 0.15 0.042
```

We obtained a covariance between the latent intercept and slope variables of .15, with a $SE = 0.042$.

d. Estimate a model that assumes homogeneity of variance of the measurement residuals. In this model, remove the correlated measurement residuals so you can test this model against the unconstrained model that was estimated at point *(a)*. What can you conclude from the comparison of both models, and from the model that included the auto-correlations between error terms?

To assume homogeneity of variance for the measurement residuals, we need to constrain the residuals to be equal across measurement occasions.

```
# Model syntax.
model_ex_1_d <- "
    # Latent intercept variable.
    int =~ 1 * bmi1 + 1 * bmi2 + 1 * bmi3 + 1 * bmi4 + 1 * bmi5 + 1 * bmi6

    # Latent slope variable.
    slp =~ 0 * bmi1 + 1 * bmi2 + 2 * bmi3 + 3 * bmi4 + 4 * bmi5 + 5 * bmi6

    # Latent variances and covariances.
    int ~~ int
    slp ~~ slp
    int ~~ slp

    # Constrained residuals.
    bmi1 ~~ a * bmi1
    bmi2 ~~ a * bmi2
    bmi3 ~~ a * bmi3
    bmi4 ~~ a * bmi4
    bmi5 ~~ a * bmi5
    bmi6 ~~ a * bmi6
"

# Fit model.
model_ex_1_d_fit <- growth(model_ex_1_d, data = data)

# Visualize the model.
semPaths(model_ex_1_d_fit, what = "paths", whatLabels = "std")
```




```
# Model summary.
summary(model_ex_1_d_fit, standardized = TRUE, rsquare = TRUE)
```

```
## lavaan 0.6-9 ended normally after 65 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        11
##   Number of equality constraints                     5
##
##   Number of observations                          5335
##
## Model Test User Model:
##
##   Test statistic                               969.387
##   Degrees of freedom                                21
##   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
##   int =~
##     bmi1             1.000                              4.827    0.965
##     bmi2             1.000                              4.827    0.963
##     bmi3             1.000                              4.827    0.954
##     bmi4             1.000                              4.827    0.938
##     bmi5             1.000                              4.827    0.916
##     bmi6             1.000                              4.827    0.889
##   slp =~
##     bmi1             0.000                              0.000    0.000
##     bmi2             1.000                              0.446    0.089
##     bmi3             2.000                              0.892    0.176
##     bmi4             3.000                              1.338    0.260
##     bmi5             4.000                              1.785    0.339
##     bmi6             5.000                              2.231    0.411
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~~
##     slp             -0.053    0.037   -1.434    0.152   -0.025   -0.025
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .bmi1             0.000                              0.000    0.000
##    .bmi2             0.000                              0.000    0.000
##    .bmi3             0.000                              0.000    0.000
##    .bmi4             0.000                              0.000    0.000
##    .bmi5             0.000                              0.000    0.000
##    .bmi6             0.000                              0.000    0.000
##     int             27.219    0.067  404.180    0.000    5.639    5.639
##     slp              0.145    0.007   19.379    0.000    0.324    0.324
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     int             23.296    0.469   49.718    0.000    1.000    1.000
##     slp              0.199    0.006   34.125    0.000    1.000    1.000
##    .bmi1       (a)   1.718    0.017  103.296    0.000    1.718    0.069
##    .bmi2       (a)   1.718    0.017  103.296    0.000    1.718    0.068
##    .bmi3       (a)   1.718    0.017  103.296    0.000    1.718    0.067
##    .bmi4       (a)   1.718    0.017  103.296    0.000    1.718    0.065
##    .bmi5       (a)   1.718    0.017  103.296    0.000    1.718    0.062
##    .bmi6       (a)   1.718    0.017  103.296    0.000    1.718    0.058
##
## R-Square:
##                   Estimate
##     bmi1             0.931
##     bmi2             0.932
##     bmi3             0.933
##     bmi4             0.935
##     bmi5             0.938
##     bmi6             0.942
```

```r
# Fit measures.
fitMeasures(model_ex_1_d_fit, fit.measures = fit_indices)
```

```
##        chisq           df       pvalue          cfi          tli        rmsea rmsea.pvalue
##      969.387       21.000        0.000        0.984        0.989        0.092        0.000
##         srmr
##        0.015
```

To compare the fit of `model_ex_1_d` to `model_ex_1_a` we can take a look at the fit measures and also perform a *Likelihood Ratio Test* (LRT).

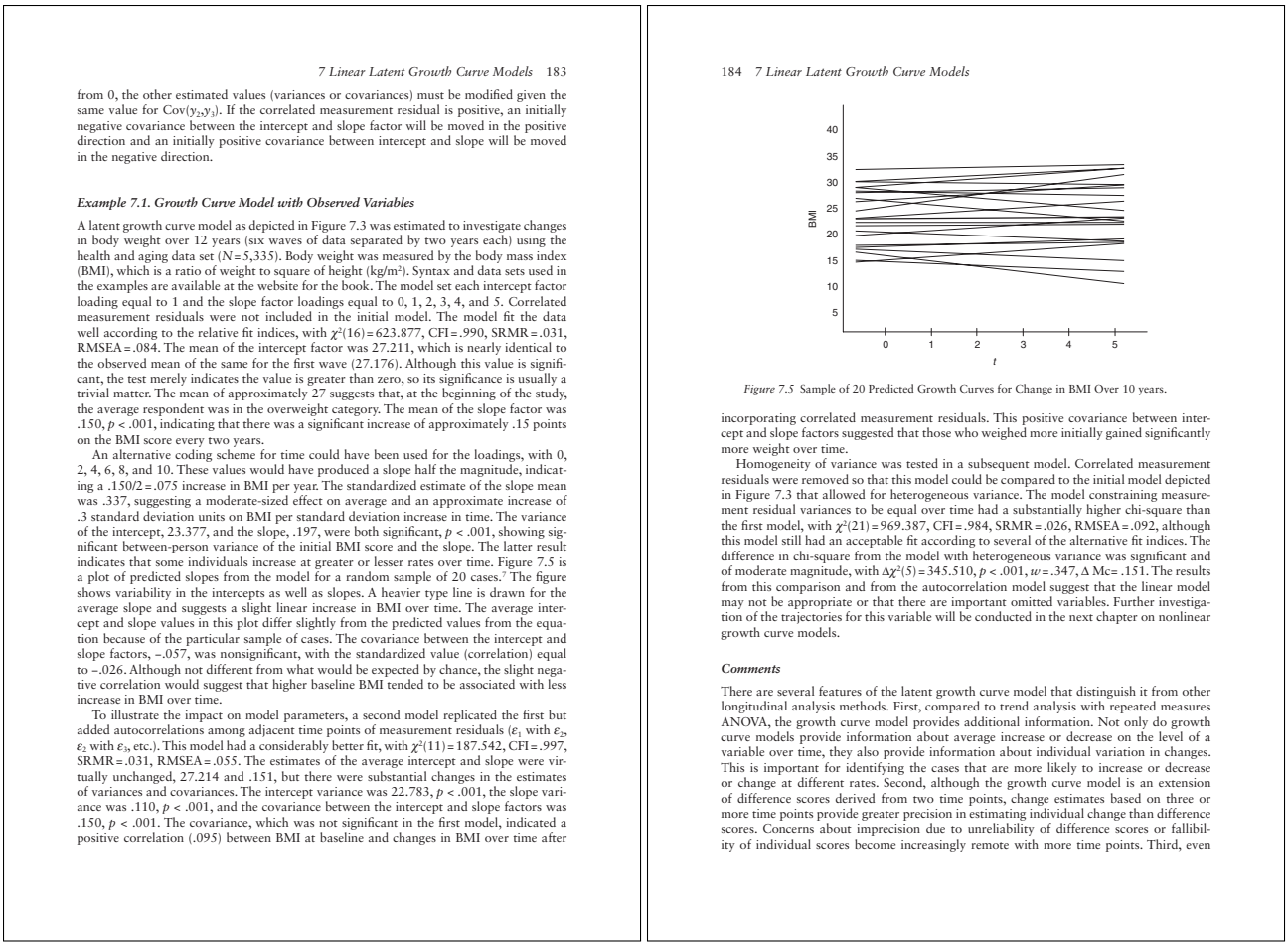
```r
# Put all fit measures in a data frame or convenience.
fit_measures_ex_1 <- cbind(
    fit_measures_ex_1,
    model_ex_1_d = fitMeasures(model_ex_1_d_fit, fit.measures = fit_indices)
)

# Print all fit measures rounded to four decimals.
round(fit_measures_ex_1, 4)
```

```
##              model_ex_1_a model_ex_1_b model_ex_1_d
## chisq             623.8767     187.5420     969.3870
## df                 16.0000      11.0000      21.0000
## pvalue              0.0000       0.0000       0.0000
## cfi                 0.9897       0.9970       0.9839
## tli                 0.9903       0.9959       0.9885
## rmsea               0.0844       0.0548       0.0920
## rmsea.pvalue        0.0000       0.1163       0.0000
## srmr                0.0162       0.0180       0.0146
```

```r
# Perform the LRT.
anova(model_ex_1_a_fit, model_ex_1_d_fit)
```

```
## Chi-Squared Difference Test
##
##                  Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## model_ex_1_a_fit 16 137358 137430 623.88
## model_ex_1_d_fit 21 137693 137733 969.39     345.51       5  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For a nuanced explanation, see *Figure 2* that shows the explanation provided by Newsom (2015, p. 183) under section *Example 7.1. Growth Curve Model with Observed Variables.*



7 Linear Latent Growth Curve Models 183

from 0, the other estimated values (variances or covariances) must be modified given the same value for Cov(y_2,y_3). If the correlated measurement residual is positive, an initially negative covariance between the intercept and slope factor will be moved in the positive direction and an initially positive covariance between intercept and slope will be moved in the negative direction.

Example 7.1. Growth Curve Model with Observed Variables

A latent growth curve model as depicted in Figure 7.3 was estimated to investigate changes in body weight over 12 years (six waves of data separated by two years each) using the health and aging data set (N = 5,335). Body weight was measured by the body mass index (BMI), which is a ratio of weight to square of height (kg/m^2). Syntax and data sets used in the examples are available at the website for the book. The model set each intercept factor loading equal to 1 and the slope factor loadings equal to 0, 1, 2, 3, 4, and 5. Correlated measurement residuals were not included in the initial model. The model fit the data well according to the relative fit indices, with $\chi^2(16) = 623.877$, CFI = .990, SRMR = .031, RMSEA = .084. The mean of the intercept factor was 27.211, which is nearly identical to the observed mean of the same for the first wave (27.176). Although this value is significant, the test merely indicates the value is greater than zero, so its significance is usually a trivial matter. The mean of approximately 27 suggests that, at the beginning of the study, the average respondent was in the overweight category. The mean of the slope factor was .150, $p < .001$, indicating that there was a significant increase of approximately .15 points on the BMI score every two years.

An alternative coding scheme for time could have been used for the loadings, with 0, 2, 4, 6, 8, and 10. These values would have produced a slope half the magnitude, indicating a .150/2 = .075 increase in BMI per year. The standardized estimate of the slope mean was .337, suggesting a moderate-sized effect on average and an approximate increase of .3 standard deviation units on BMI per standard deviation increase in time. The variance of the intercept, 23.377, and the slope, .197, were both significant, $p < .001$, showing significant between-person variance of the initial BMI score and the slope. The latter result indicates that some individuals increase at greater or lesser rates over time. Figure 7.5 is a plot of predicted slopes from the model for a random sample of 20 cases.[7] The figure shows variability in the intercepts as well as slopes. A heavier type line is drawn for the average slope and suggests a slight linear increase in BMI over time. The average intercept and slope values in this plot differ slightly from the predicted values from the equation because of the particular sample of cases. The covariance between the intercept and slope factors, −.057, was nonsignificant, with the standardized value (correlation) equal to −.026. Although not different from what would be expected by chance, the slight negative correlation would suggest that higher baseline BMI tended to be associated with less increase in BMI over time.

To illustrate the impact on model parameters, a second model replicated the first but added autocorrelations among adjacent time points of measurement residuals (ε_1 with ε_2, ε_2 with ε_3, etc.). This model had a considerably better fit, with $\chi^2(11) = 187.542$, CFI = .997, SRMR = .031, RMSEA = .055. The estimates of the average intercept and slope were virtually unchanged, 27.214 and .151, but there were substantial changes in the estimates of variances and covariances. The intercept variance was 22.783, $p < .001$, the slope variance was .110, $p < .001$, and the covariance between the intercept and slope factors was .150, $p < .001$. The covariance, which was not significant in the first model, indicated a positive correlation (.095) between BMI at baseline and changes in BMI over time after

184 7 Linear Latent Growth Curve Models

Figure 7.5 Sample of 20 Predicted Growth Curves for Change in BMI Over 10 years.

incorporating correlated measurement residuals. This positive covariance between intercept and slope factors suggested that those who weighed more initially gained significantly more weight over time.

Homogeneity of variance was tested in a subsequent model. Correlated measurement residuals were removed so that this model could be compared to the initial model depicted in Figure 7.3 that allowed for heterogeneous variance. The model constraining measurement residual variances to be equal over time had a substantially higher chi-square than the first model, with $\chi^2(21) = 969.387$, CFI = .984, SRMR = .026, RMSEA = .092, although this model still had an acceptable fit according to several of the alternative fit indices. The difference in chi-square from the model with heterogeneous variance was significant and of moderate magnitude, with $\Delta\chi^2(5) = 345.510$, $p < .001$, w = .347, Δ Mc= .151. The results from this comparison and from the autocorrelation model suggest that the linear model may not be appropriate or that there are important omitted variables. Further investigation of the trajectories for this variable will be conducted in the next chapter on nonlinear growth curve models.

Comments

There are several features of the latent growth curve model that distinguish it from other longitudinal analysis methods. First, compared to trend analysis with repeated measures ANOVA, the growth curve model provides additional information. Not only do growth curve models provide information about average increase or decrease on the level of a variable over time, they also provide information about individual variation in changes. This is important for identifying the cases that are more likely to increase or decrease or change at different rates. Second, although the growth curve model is an extension of difference scores derived from two time points, change estimates based on three or more time points provide greater precision in estimating individual change than difference scores. Concerns about imprecision due to unreliability of difference scores or fallibility of individual scores become increasingly remote with more time points. Third, even

Figure 2: Explanation for *Exercise 1*.

Exercise 2

In this exercise you are going to keep using the `health.dat` and extend the model from *Exercise 1* with a time-invariant covariate, namely the age of the participants.

a. Compute a new variable `age_c` which is the grand mean centered age of the participants.

   - *Tip.* You can obtain the sample mean using `mean(data$age)`.

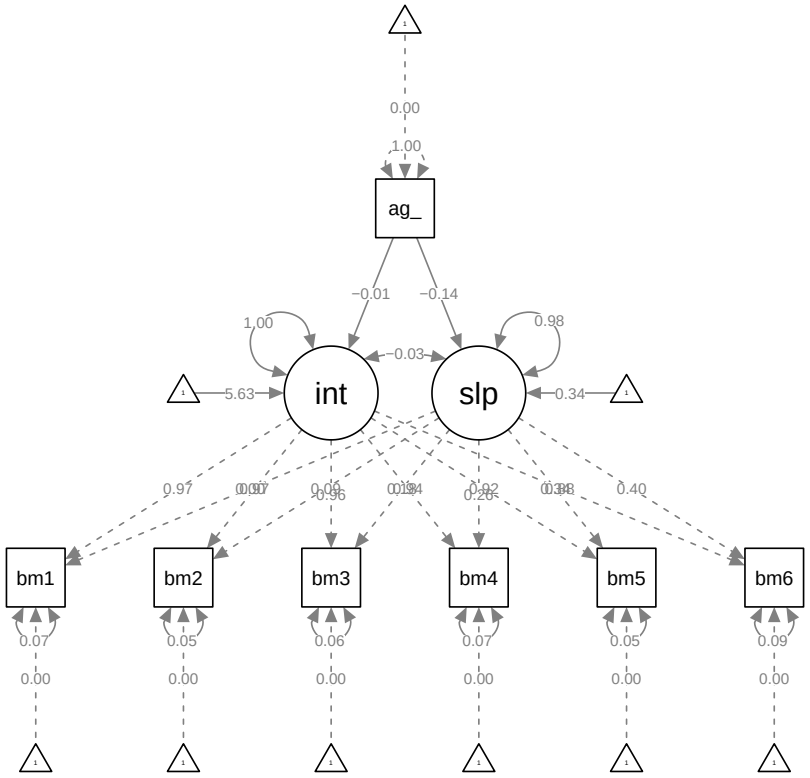
To create `age_c`, we need to subtract the grand mean from the `age` variable.

```
# Create the centered age variable.
data$age_c <- data$age - mean(data$age)
```

b. Estimate the conditional *LGC* model of *BMI* by regressing the latent intercept and the latent slope on `age_c`.

```
# Model syntax.
model_ex_2_b <- "
    # Latent intercept variable.
    int =~ 1 * bmi1 + 1 * bmi2 + 1 * bmi3 + 1 * bmi4 + 1 * bmi5 + 1 * bmi6
```

```
    # Latent slope variable.
    slp =~ 0 * bmi1 + 1 * bmi2 + 2 * bmi3 + 3 * bmi4 + 4 * bmi5 + 5 * bmi6

    # Latent variances and covariances.
    int ~~ int
    slp ~~ slp
    int ~~ slp

    # Regression equations.
    int + slp ~ age_c
"

# Fit model.
model_ex_2_b_fit <- growth(model_ex_2_b, data = data)

# Visualize the model.
semPaths(model_ex_2_b_fit, what = "paths", whatLabels = "std")
```




```
# Model summary.
summary(model_ex_2_b_fit, standardized = TRUE, rsquare = TRUE)
```

```
## lavaan 0.6-9 ended normally after 89 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                          5335
```

```
##
## Model Test User Model:
##
##   Test statistic                               641.026
##   Degrees of freedom                                20
##   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
##   int =~
##     bmi1             1.000                               4.835    0.967
##     bmi2             1.000                               4.835    0.970
##     bmi3             1.000                               4.835    0.957
##     bmi4             1.000                               4.835    0.937
##     bmi5             1.000                               4.835    0.921
##     bmi6             1.000                               4.835    0.878
##   slp =~
##     bmi1             0.000                               0.000    0.000
##     bmi2             1.000                               0.444    0.089
##     bmi3             2.000                               0.888    0.176
##     bmi4             3.000                               1.332    0.258
##     bmi5             4.000                               1.776    0.338
##     bmi6             5.000                               2.220    0.403
##
## Regressions:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~
##     age_c           -0.011    0.016   -0.652    0.514   -0.002   -0.009
##   slp ~
##     age_c           -0.015    0.002   -8.403    0.000   -0.035   -0.141
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .int ~~
##    .slp             -0.060    0.037   -1.598    0.110   -0.028   -0.028
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .bmi1              0.000                               0.000    0.000
##    .bmi2              0.000                               0.000    0.000
##    .bmi3              0.000                               0.000    0.000
##    .bmi4              0.000                               0.000    0.000
##    .bmi5              0.000                               0.000    0.000
##    .bmi6              0.000                               0.000    0.000
##    .int              27.211    0.067  403.954    0.000    5.628    5.628
```

```
##    .slp               0.150    0.007   20.012    0.000    0.337    0.337
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .int              23.375    0.469   49.846    0.000    1.000    1.000
##    .slp               0.193    0.006   32.379    0.000    0.980    0.980
##    .bmi1              1.631    0.054   30.386    0.000    1.631    0.065
##    .bmi2              1.361    0.038   36.002    0.000    1.361    0.055
##    .bmi3              1.568    0.037   42.144    0.000    1.568    0.061
##    .bmi4              1.826    0.043   42.640    0.000    1.826    0.069
##    .bmi5              1.483    0.043   34.263    0.000    1.483    0.054
##    .bmi6              2.612    0.074   35.383    0.000    2.612    0.086
##
## R-Square:
##                   Estimate
##     int               0.000
##     slp               0.020
##     bmi1              0.935
##     bmi2              0.945
##     bmi3              0.939
##     bmi4              0.931
##     bmi5              0.946
##     bmi6              0.914
```

```
# Fit measures.
fitMeasures(model_ex_2_b_fit, fit.measures = fit_indices)
```

```
##        chisq           df       pvalue          cfi          tli        rmsea rmsea.pvalue
##      641.026       20.000        0.000        0.989        0.989        0.076        0.000
##         srmr
##        0.014
```

c. Evaluate the fit of this model, and the effect of `age_c` on the latent intercept and latent slope.

```
# Fit measures.
fitMeasures(model_ex_2_b_fit, fit.measures = fit_indices)
```

```
##        chisq           df       pvalue          cfi          tli        rmsea rmsea.pvalue
##      641.026       20.000        0.000        0.989        0.989        0.076        0.000
##         srmr
##        0.014
```

d. Compute a new binary age variable called `age_b` for implementing the *MIMIC* approach discussed during *Lecture 8*, using the following `R` code:

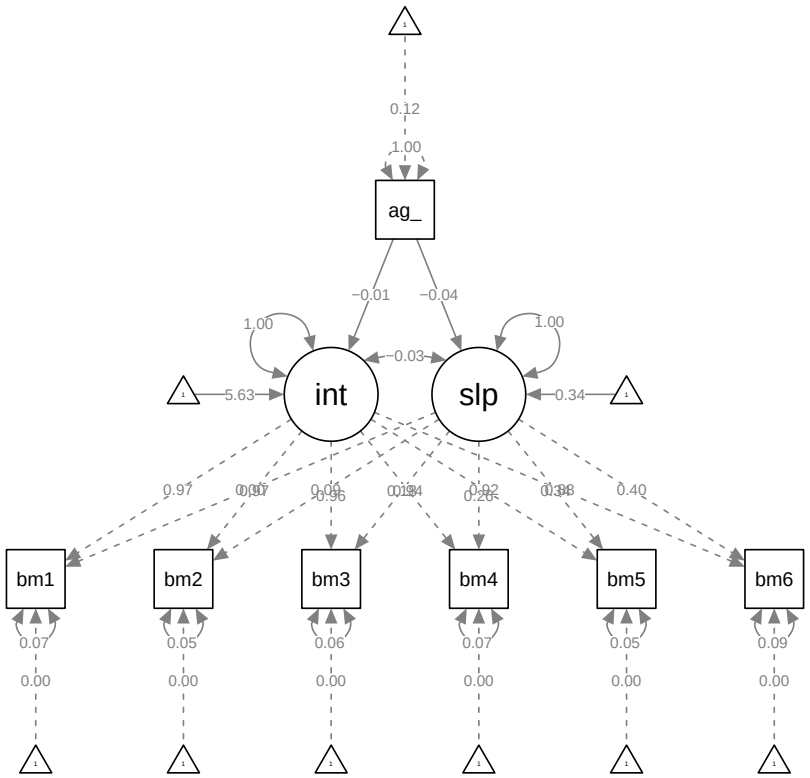
First we compute the binary variable.

```
data$age_b <- ifelse(data$age <= 65, 0, 1)
```

e. Estimate the *MIMIC* model by regressing the intercept and the slope factors on the binary age variable. Interpret the effects of the binary age variable.

Now we can implement the *MIMIC* approach. Note that the only thing that changes is that in our regression equations we now use the binary variable `age_b` instead of the mean-centered variable `age_c`.

```
# Model syntax.
model_ex_2_e <- "
    # Latent intercept variable.
    int =~ 1 * bmi1 + 1 * bmi2 + 1 * bmi3 + 1 * bmi4 + 1 * bmi5 + 1 * bmi6

    # Latent slope variable.
    slp =~ 0 * bmi1 + 1 * bmi2 + 2 * bmi3 + 3 * bmi4 + 4 * bmi5 + 5 * bmi6

    # Latent variances and covariances.
    int ~~ int
    slp ~~ slp
    int ~~ slp

    # Regression equations.
    int + slp ~ age_b
"

# Fit model.
model_ex_2_e_fit <- growth(model_ex_2_e, data = data)

# Visualize the model.
semPaths(model_ex_2_e_fit, what = "paths", whatLabels = "std")
```




```
# Model summary.
summary(model_ex_2_e_fit, standardized = TRUE, rsquare = TRUE)
```

```
## lavaan 0.6-9 ended normally after 109 iterations
##
```

```
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                          5335
##
## Model Test User Model:
##
##   Test statistic                               626.009
##   Degrees of freedom                                20
##   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
##   int =~
##     bmi1             1.000                               4.835    0.967
##     bmi2             1.000                               4.835    0.970
##     bmi3             1.000                               4.835    0.957
##     bmi4             1.000                               4.835    0.937
##     bmi5             1.000                               4.835    0.921
##     bmi6             1.000                               4.835    0.877
##   slp =~
##     bmi1             0.000                               0.000    0.000
##     bmi2             1.000                               0.444    0.089
##     bmi3             2.000                               0.888    0.176
##     bmi4             3.000                               1.332    0.258
##     bmi5             4.000                               1.776    0.338
##     bmi6             5.000                               2.220    0.403
##
## Regressions:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~
##     age_b           -0.569    0.580   -0.981    0.327   -0.118   -0.014
##   slp ~
##     age_b           -0.157    0.065   -2.419    0.016   -0.353   -0.041
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##  .int ~~
##    .slp             -0.058    0.037   -1.545    0.122   -0.027   -0.027
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .bmi1             0.000                               0.000    0.000
##    .bmi2             0.000                               0.000    0.000
```

```
##    .bmi3              0.000                           0.000    0.000
##    .bmi4              0.000                           0.000    0.000
##    .bmi5              0.000                           0.000    0.000
##    .bmi6              0.000                           0.000    0.000
##    .int              27.219    0.068  401.313    0.000    5.630    5.630
##    .slp               0.152    0.008   20.042    0.000    0.342    0.342
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .int              23.373    0.469   49.846    0.000    1.000    1.000
##    .slp               0.197    0.006   32.607    0.000    0.998    0.998
##    .bmi1              1.632    0.054   30.323    0.000    1.632    0.065
##    .bmi2              1.361    0.038   35.962    0.000    1.361    0.055
##    .bmi3              1.567    0.037   42.143    0.000    1.567    0.061
##    .bmi4              1.823    0.043   42.619    0.000    1.823    0.068
##    .bmi5              1.478    0.043   34.148    0.000    1.478    0.054
##    .bmi6              2.626    0.074   35.399    0.000    2.626    0.086
##
## R-Square:
##                   Estimate
##     int               0.000
##     slp               0.002
##     bmi1              0.935
##     bmi2              0.945
##     bmi3              0.939
##     bmi4              0.932
##     bmi5              0.946
##     bmi6              0.914
```

```
# Fit measures.
fitMeasures(model_ex_2_e_fit, fit.measures = fit_indices)
```

```
##        chisq           df       pvalue          cfi          tli        rmsea rmsea.pvalue
##      626.009       20.000        0.000        0.990        0.989        0.075        0.000
##         srmr
##        0.014
```

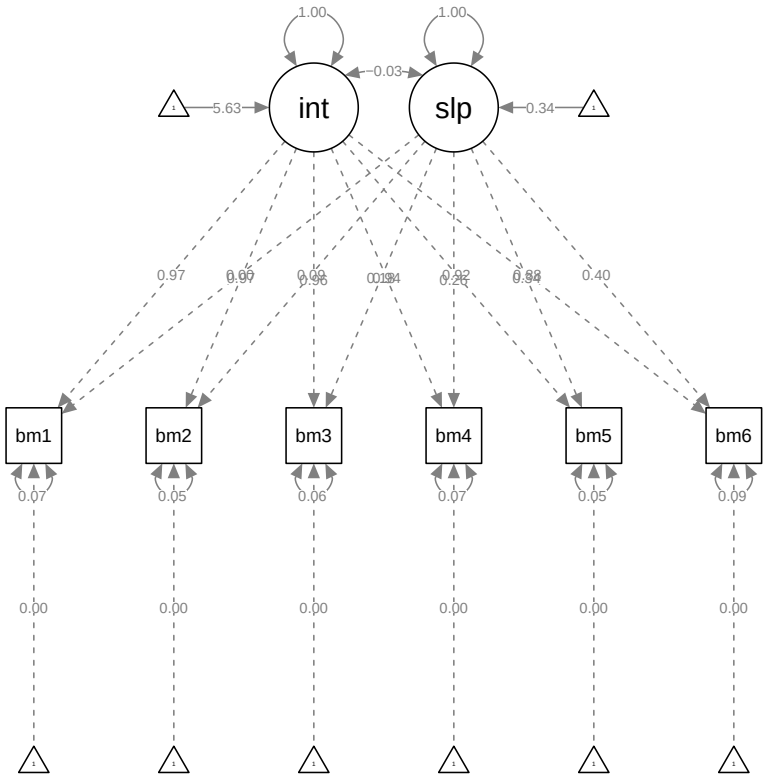
f. Test the same hypothesis of age differences in trajectories, using the multi-group approach and comparing those under 65 to those aged 65 and older.

For this model, the intercept and slope factor variances were constrained to be equal in both age groups. The constraints are placed to obtain a more stable estimate of the variances in > 65 group (i.e., which had a small sample size), and to provide more comparable results to the *MIMIC* modeling approach. The variance constraints did not significantly degrade the fit of the model, so these appeared to be empirically justifiable assumptions.

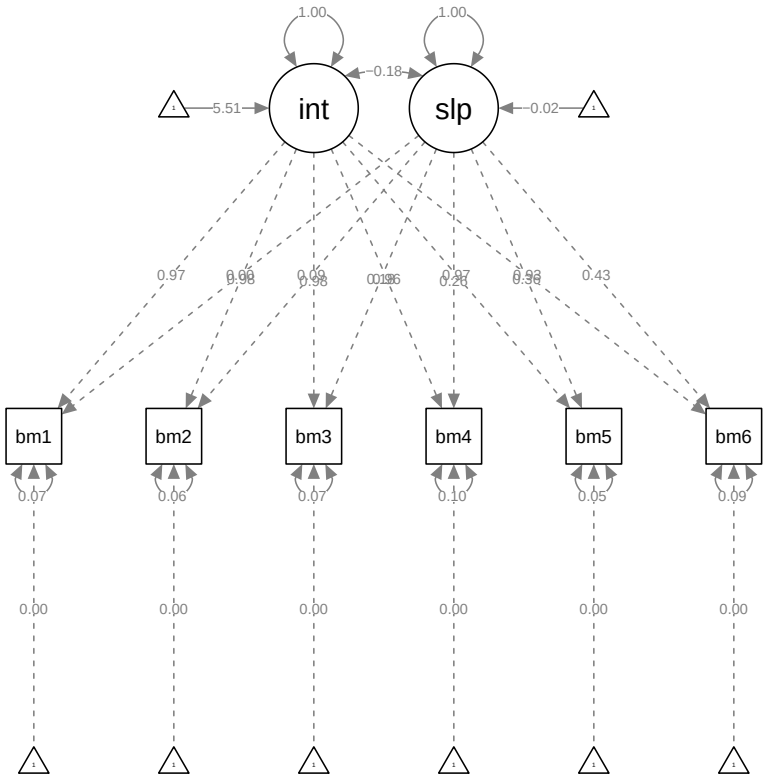
First, we fit a model (i.e., `model_ex_1_a`) where we constrain the latent variable variances to be equal across the age groups. We can use the `group` and `group.equal` arguments in `lavaan` to do so. Note that we do not yet constrain the means of the slope latent variable yet.

```
model_ex_2_f <- "
    # Latent intercept variable.
    int =~ 1 * bmi1 + 1 * bmi2 + 1 * bmi3 + 1 * bmi4 + 1 * bmi5 + 1 * bmi6

    # Latent slope variable.
    slp =~ 0 * bmi1 + 1 * bmi2 + 2 * bmi3 + 3 * bmi4 + 4 * bmi5 + 5 * bmi6

    # Latent variances and covariances.
    int ~~ int
    slp ~~ slp
    int ~~ slp
"

# Fit model.
model_ex_2_f_fit <- growth(model_ex_2_f, data = data, group = "age_b", group.equal = "lv.variances")

# Visualize the model.
semPaths(model_ex_2_f_fit, what = "paths", whatLabels = "std")
```

1



2




```
# Model summary.
summary(model_ex_2_f_fit, standardized = TRUE, rsquare = TRUE)
```

```
## lavaan 0.6-9 ended normally after 231 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        22
##   Number of equality constraints                     2
##
##   Number of observations per group:
##     0                                            5262
##     1                                              73
##
## Model Test User Model:
##
##   Test statistic                               633.350
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.000
##   Test statistic for each group:
##     0                                          617.660
##     1                                           15.691
##
## Parameter Estimates:
##
```

```
##   Standard errors                           Standard
##   Information                               Expected
##   Information saturated (h1) model          Structured
##
##
## Group 1 [0]:
##
## Latent Variables:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int =~
##     bmi1             1.000                               4.835    0.967
##     bmi2             1.000                               4.835    0.970
##     bmi3             1.000                               4.835    0.957
##     bmi4             1.000                               4.835    0.937
##     bmi5             1.000                               4.835    0.921
##     bmi6             1.000                               4.835    0.877
##   slp =~
##     bmi1             0.000                               0.000    0.000
##     bmi2             1.000                               0.444    0.089
##     bmi3             2.000                               0.888    0.176
##     bmi4             3.000                               1.331    0.258
##     bmi5             4.000                               1.775    0.338
##     bmi6             5.000                               2.219    0.403
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~~
##     slp             -0.054    0.038   -1.443    0.149   -0.025   -0.025
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .bmi1             0.000                               0.000    0.000
##    .bmi2             0.000                               0.000    0.000
##    .bmi3             0.000                               0.000    0.000
##    .bmi4             0.000                               0.000    0.000
##    .bmi5             0.000                               0.000    0.000
##    .bmi6             0.000                               0.000    0.000
##     int             27.219    0.068  401.310    0.000    5.630    5.630
##     slp              0.152    0.008   20.043    0.000    0.342    0.342
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     int     (.13.)   23.373    0.469   49.846    0.000    1.000    1.000
##     slp     (.14.)    0.197    0.006   32.612    0.000    1.000    1.000
##    .bmi1              1.632    0.054   30.137    0.000    1.632    0.065
##    .bmi2              1.360    0.038   35.716    0.000    1.360    0.055
##    .bmi3              1.567    0.037   41.858    0.000    1.567    0.061
##    .bmi4              1.813    0.043   42.280    0.000    1.813    0.068
##    .bmi5              1.479    0.044   33.939    0.000    1.479    0.054
##    .bmi6              2.628    0.075   35.176    0.000    2.628    0.087
##
```

```
## R-Square:
##                    Estimate
##     bmi1              0.935
##     bmi2              0.945
##     bmi3              0.939
##     bmi4              0.932
##     bmi5              0.946
##     bmi6              0.913
##
##
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int =~
##     bmi1              1.000                               4.835    0.967
##     bmi2              1.000                               4.835    0.983
##     bmi3              1.000                               4.835    0.982
##     bmi4              1.000                               4.835    0.959
##     bmi5              1.000                               4.835    0.971
##     bmi6              1.000                               4.835    0.930
##   slp =~
##     bmi1              0.000                               0.000    0.000
##     bmi2              1.000                               0.444    0.090
##     bmi3              2.000                               0.888    0.180
##     bmi4              3.000                               1.331    0.264
##     bmi5              4.000                               1.775    0.356
##     bmi6              5.000                               2.219    0.427
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~~
##     slp              -0.385    0.292   -1.319    0.187   -0.179   -0.179
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .bmi1              0.000                               0.000    0.000
##    .bmi2              0.000                               0.000    0.000
##    .bmi3              0.000                               0.000    0.000
##    .bmi4              0.000                               0.000    0.000
##    .bmi5              0.000                               0.000    0.000
##    .bmi6              0.000                               0.000    0.000
##     int              26.648    0.576   46.268    0.000    5.512    5.512
##     slp              -0.009    0.064   -0.137    0.891   -0.020   -0.020
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     int     (.13.)   23.373    0.469   49.846    0.000    1.000    1.000
##     slp     (.14.)    0.197    0.006   32.612    0.000    1.000    1.000
##    .bmi1              1.632    0.463    3.525    0.000    1.632    0.065
##    .bmi2              1.397    0.335    4.169    0.000    1.397    0.058
```

```
##    .bmi3              1.605    0.327    4.913    0.000    1.605    0.066
##    .bmi4              2.559    0.485    5.279    0.000    2.559    0.101
##    .bmi5              1.350    0.356    3.797    0.000    1.350    0.054
##    .bmi6              2.549    0.622    4.101    0.000    2.549    0.094
##
## R-Square:
##                    Estimate
##     bmi1              0.935
##     bmi2              0.942
##     bmi3              0.934
##     bmi4              0.899
##     bmi5              0.946
##     bmi6              0.906
```

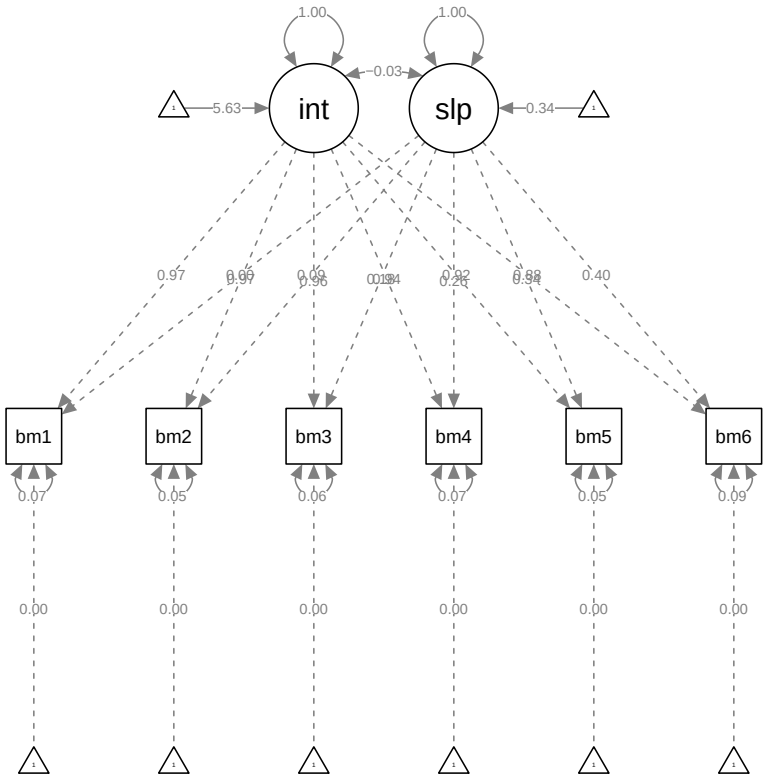
```
# Fit measures.
fitMeasures(model_ex_2_f_fit, fit.measures = fit_indices)
```

```
##        chisq           df       pvalue          cfi          tli        rmsea rmsea.pvalue
##      633.350       34.000        0.000        0.990        0.991        0.081        0.000
##         srmr
##        0.019
```

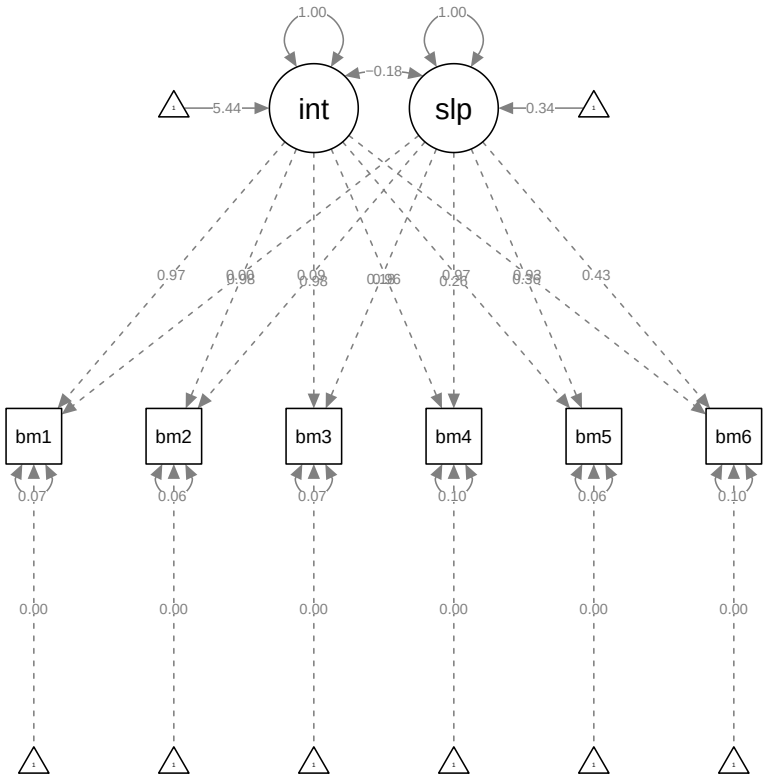
Now, on top of the variance constraints, we also constrain the intercept of the slope latent variable to be equal across groups. Note that since we use the `group` argument in `lavaan`, this time we have to specify the constraints in the form of a vector (i.e., `c(a, a)`, which means that both the < 65 and > 65 age groups will be applied the same label `a` for the latent slope variable intercept).

```
model_ex_2_f_slp_con <- "
    # Latent intercept variable.
    int =~ 1 * bmi1 + 1 * bmi2 + 1 * bmi3 + 1 * bmi4 + 1 * bmi5 + 1 * bmi6

    # Latent slope variable.
    slp =~ 0 * bmi1 + 1 * bmi2 + 2 * bmi3 + 3 * bmi4 + 4 * bmi5 + 5 * bmi6

    # Latent variances and covariances.
    int ~~ int
    slp ~~ slp
    int ~~ slp

    # Intercept constraint.
    slp ~ c(a, a) * 1
"

# Fit model.
model_ex_2_f_slp_con_fit <- growth(model_ex_2_f_slp_con, data = data, group = "age_b", group.equal = "lv.variances")

# Visualize the model.
semPaths(model_ex_2_f_slp_con_fit, what = "paths", whatLabels = "std")
```

1


1.00
1.00
−0.03
int
slp
5.63
0.34
0.97
0.40
bm1
bm2
bm3
bm4
bm5
bm6
0.07
0.05
0.06
0.07
0.05
0.09
0.00
0.00
0.00
0.00
0.00
0.00

2




```
# Model summary.
summary(model_ex_2_f_slp_con_fit, standardized = TRUE, rsquare = TRUE)
```

```
## lavaan 0.6-9 ended normally after 244 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        22
##   Number of equality constraints                     3
##
##   Number of observations per group:
##     0                                            5262
##     1                                              73
##
## Model Test User Model:
##
##   Test statistic                               639.474
##   Degrees of freedom                                35
##   P-value (Chi-square)                           0.000
##   Test statistic for each group:
##     0                                          617.744
##     1                                           21.730
##
## Parameter Estimates:
##
```

```
##   Standard errors                           Standard
##   Information                               Expected
##   Information saturated (h1) model          Structured
##
##
## Group 1 [0]:
##
## Latent Variables:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int =~
##     bmi1             1.000                               4.835    0.967
##     bmi2             1.000                               4.835    0.970
##     bmi3             1.000                               4.835    0.957
##     bmi4             1.000                               4.835    0.937
##     bmi5             1.000                               4.835    0.921
##     bmi6             1.000                               4.835    0.877
##   slp =~
##     bmi1             0.000                               0.000    0.000
##     bmi2             1.000                               0.444    0.089
##     bmi3             2.000                               0.888    0.176
##     bmi4             3.000                               1.332    0.258
##     bmi5             4.000                               1.777    0.338
##     bmi6             5.000                               2.221    0.403
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~~
##     slp             -0.055    0.038   -1.452    0.147   -0.026   -0.026
##
## Intercepts:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     slp        (a)    0.150    0.008   19.880    0.000    0.337    0.337
##    .bmi1             0.000                               0.000    0.000
##    .bmi2             0.000                               0.000    0.000
##    .bmi3             0.000                               0.000    0.000
##    .bmi4             0.000                               0.000    0.000
##    .bmi5             0.000                               0.000    0.000
##    .bmi6             0.000                               0.000    0.000
##     int             27.221    0.068  401.367    0.000    5.630    5.630
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     int     (.13.)   23.374    0.469   49.846    0.000    1.000    1.000
##     slp     (.14.)    0.197    0.006   32.632    0.000    1.000    1.000
##    .bmi1              1.632    0.054   30.132    0.000    1.632    0.065
##    .bmi2              1.360    0.038   35.709    0.000    1.360    0.055
##    .bmi3              1.567    0.037   41.859    0.000    1.567    0.061
##    .bmi4              1.813    0.043   42.282    0.000    1.813    0.068
##    .bmi5              1.479    0.044   33.933    0.000    1.479    0.054
##    .bmi6              2.627    0.075   35.165    0.000    2.627    0.086
##
```

```
## R-Square:
##                    Estimate
##     bmi1              0.935
##     bmi2              0.945
##     bmi3              0.939
##     bmi4              0.932
##     bmi5              0.946
##     bmi6              0.914
##
##
## Group 2 [1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int =~
##     bmi1              1.000                               4.835    0.966
##     bmi2              1.000                               4.835    0.984
##     bmi3              1.000                               4.835    0.983
##     bmi4              1.000                               4.835    0.961
##     bmi5              1.000                               4.835    0.971
##     bmi6              1.000                               4.835    0.931
##   slp =~
##     bmi1              0.000                               0.000    0.000
##     bmi2              1.000                               0.444    0.090
##     bmi3              2.000                               0.888    0.181
##     bmi4              3.000                               1.332    0.265
##     bmi5              4.000                               1.777    0.357
##     bmi6              5.000                               2.221    0.428
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   int ~~
##     slp              -0.393    0.292   -1.344    0.179   -0.183   -0.183
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     slp        (a)    0.150    0.008   19.880    0.000    0.337    0.337
##    .bmi1              0.000                               0.000    0.000
##    .bmi2              0.000                               0.000    0.000
##    .bmi3              0.000                               0.000    0.000
##    .bmi4              0.000                               0.000    0.000
##    .bmi5              0.000                               0.000    0.000
##    .bmi6              0.000                               0.000    0.000
##     int              26.315    0.560   46.977    0.000    5.443    5.443
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     int     (.13.)   23.374    0.469   49.846    0.000    1.000    1.000
##     slp     (.14.)    0.197    0.006   32.632    0.000    1.000    1.000
##    .bmi1              1.695    0.472    3.593    0.000    1.695    0.068
##    .bmi2              1.372    0.333    4.124    0.000    1.372    0.057
```

```
##    .bmi3              1.592    0.325    4.893    0.000    1.592    0.066
##    .bmi4              2.531    0.481    5.260    0.000    2.531    0.100
##    .bmi5              1.416    0.366    3.866    0.000    1.416    0.057
##    .bmi6              2.578    0.630    4.095    0.000    2.578    0.096
##
## R-Square:
##                    Estimate
##     bmi1              0.932
##     bmi2              0.943
##     bmi3              0.934
##     bmi4              0.900
##     bmi5              0.943
##     bmi6              0.904
```

```
# Fit measures.
fitMeasures(model_ex_2_f_slp_con_fit, fit.measures = fit_indices)
```

```
##        chisq           df       pvalue          cfi          tli        rmsea rmsea.pvalue
##      639.474       35.000        0.000        0.990        0.991        0.080        0.000
##         srmr
##        0.019
```

We can now compare the fit of the two models (i.e., `model_ex_2_f` and `model_ex_2_f_slp_con`) and also perform a *LRT*.

```
# Put all fit measures in a data frame or convenience.
fit_measures_ex_2_f <- data.frame(
    model_ex_2_f = fitMeasures(model_ex_2_f_fit, fit.measures = fit_indices),
    model_ex_2_f_slp_con = fitMeasures(model_ex_2_f_slp_con_fit, fit.measures = fit_indices)
)

# Print all fit measures rounded to four decimals.
round(fit_measures_ex_2_f, 4)
```

```
##              model_ex_2_f model_ex_2_f_slp_con
## chisq            633.3504             639.4737
## df                34.0000              35.0000
## pvalue             0.0000               0.0000
## cfi                0.9898               0.9898
## tli                0.9910               0.9912
## rmsea              0.0813               0.0805
## rmsea.pvalue       0.0000               0.0000
## srmr               0.0187               0.0187
```

```
# Perform the LRT.
anova(model_ex_2_f_fit, model_ex_2_f_slp_con_fit)
```

```
## Chi-Squared Difference Test
##
##                           Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## model_ex_2_f_fit          34 137364 137496 633.35
## model_ex_2_f_slp_con_fit 35 137368 137493 639.47     6.1233       1    0.01334 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

References

Newsom, J. T. (2015). *Longitudinal structural equation modeling: A comprehensive introduction.* Routledge, Taylor and Francis Group.