Structural Equation Modeling

P.05 - Model Fit and Fit Indices

November 06, 2022

Lab Description

For this practical you will need the following package: `lavaan` and `semPlot`.

You can install and load this package using the following code:

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

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

Exercise 1

a. Import the dataset `ELEMM1.csv` that is available in the course folder for *Lecture 4* on Canvas.

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.csv("ELEMM1.csv")

# Inspect the data.
View(data)
```

b. In *Practical 4*, you estimated the model in Figure 1 (see below), using the Satorra-Bentler estimator and obtained a value for the *MFTS*.
   - Re-estimate this model and now request that the modification indices are also printed in the output.
   - Evaluate the fit of this model using fit indices. *Tip: check the arguments `fit.measures` and `modindices` in `lavaan`. Also check the functions `fitmeasures()` and `modificationIndices()` in `lavaan`.*

First we are specifying the model using `lavaan` syntax.

```
# Model syntax.
model_ex_1 <- "
    # Measurement part.
    EMO =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM8 + ITEM13 + ITEM14 + ITEM16 + ITEM20
    DEP =~ ITEM5 + ITEM10 + ITEM11 + ITEM15 + ITEM22
    ACC =~ ITEM4 + ITEM7 + ITEM9 + ITEM12 + ITEM17 + ITEM18 + ITEM19 + ITEM21

    # Covariances between latent variables.
    EMO ~~ DEP
    DEP ~~ ACC
    EMO ~~ ACC
"
```

Then we are estimating the model using the Satorra-Bentler estimator and requesting the summary and the modification indices.

```
# Fit the model.
model_ex_1_fit <- cfa(model_ex_1, data = data,  estimator = "MLM")
```

To include fit measures and modification indices information in the summary output we need to indicate that to `lavaan` using the `fit.measures` and `modindices` arguments.

```
# Model summary.
summary(model_ex_1_fit, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-12 ended normally after 46 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        47
##
##   Number of observations                           372
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               695.719     567.753
##   Degrees of freedom                               206         206
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.225
##     Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                              3452.269    2911.466
##   Degrees of freedom                               231         231
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.186
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.848       0.865
```

```
##   Tucker-Lewis Index (TLI)                       0.830       0.849
##
##   Robust Comparative Fit Index (CFI)                         0.861
##   Robust Tucker-Lewis Index (TLI)                            0.844
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -12811.043  -12811.043
##   Loglikelihood unrestricted model (H1)     -12463.184  -12463.184
##
##   Akaike (AIC)                               25716.087   25716.087
##   Bayesian (BIC)                             25900.275   25900.275
##   Sample-size adjusted Bayesian (BIC)        25751.158   25751.158
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.080       0.069
##   90 Percent confidence interval - lower         0.073       0.063
##   90 Percent confidence interval - upper         0.087       0.075
##   P-value RMSEA <= 0.05                          0.000       0.000
##
##   Robust RMSEA                                               0.076
##   90 Percent confidence interval - lower                     0.069
##   90 Percent confidence interval - upper                     0.084
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.073       0.073
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO =~
##     ITEM1             1.000                               1.275    0.768
##     ITEM2             0.887    0.040   22.391    0.000    1.131    0.732
##     ITEM3             1.021    0.053   19.310    0.000    1.302    0.752
##     ITEM6             0.764    0.070   10.974    0.000    0.975    0.616
##     ITEM8             1.143    0.059   19.366    0.000    1.458    0.845
##     ITEM13            1.017    0.062   16.340    0.000    1.297    0.772
##     ITEM14            0.848    0.058   14.584    0.000    1.081    0.627
##     ITEM16            0.715    0.066   10.826    0.000    0.912    0.634
##     ITEM20            0.753    0.061   12.303    0.000    0.960    0.679
##   DEP =~
##     ITEM5             1.000                               0.839    0.565
##     ITEM10            1.142    0.152    7.509    0.000    0.958    0.663
##     ITEM11            1.353    0.162    8.368    0.000    1.135    0.743
```

```
##     ITEM15          0.905    0.123    7.366    0.000    0.760    0.586
##     ITEM22          0.768    0.122    6.284    0.000    0.644    0.408
##   ACC =~
##     ITEM4           1.000                               0.439    0.440
##     ITEM7           0.970    0.128    7.563    0.000    0.426    0.507
##     ITEM9           1.780    0.322    5.529    0.000    0.781    0.594
##     ITEM12          1.499    0.241    6.232    0.000    0.658    0.552
##     ITEM17          1.348    0.200    6.757    0.000    0.592    0.695
##     ITEM18          1.918    0.298    6.435    0.000    0.842    0.662
##     ITEM19          1.716    0.287    5.978    0.000    0.753    0.634
##     ITEM21          1.356    0.227    5.984    0.000    0.595    0.471
##
## Covariances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO ~~
##     DEP             0.701    0.106    6.608    0.000    0.655    0.655
##   DEP ~~
##     ACC            -0.172    0.036   -4.777    0.000   -0.466   -0.466
##   EMO ~~
##     ACC            -0.192    0.040   -4.796    0.000   -0.343   -0.343
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .ITEM1           1.128    0.093   12.177    0.000    1.128    0.410
##    .ITEM2           1.105    0.088   12.506    0.000    1.105    0.464
##    .ITEM3           1.301    0.106   12.317    0.000    1.301    0.434
##    .ITEM6           1.553    0.134   11.550    0.000    1.553    0.621
##    .ITEM8           0.852    0.082   10.450    0.000    0.852    0.286
##    .ITEM13          1.142    0.124    9.173    0.000    1.142    0.404
##    .ITEM14          1.804    0.142   12.730    0.000    1.804    0.607
##    .ITEM16          1.235    0.110   11.278    0.000    1.235    0.598
##    .ITEM20          1.075    0.137    7.860    0.000    1.075    0.539
##    .ITEM5           1.503    0.179    8.381    0.000    1.503    0.681
##    .ITEM10          1.169    0.147    7.959    0.000    1.169    0.560
##    .ITEM11          1.044    0.141    7.398    0.000    1.044    0.447
##    .ITEM15          1.106    0.153    7.220    0.000    1.106    0.657
##    .ITEM22          2.076    0.184   11.266    0.000    2.076    0.833
##    .ITEM4           0.802    0.113    7.124    0.000    0.802    0.806
##    .ITEM7           0.523    0.075    7.010    0.000    0.523    0.743
##    .ITEM9           1.117    0.149    7.487    0.000    1.117    0.647
##    .ITEM12          0.987    0.126    7.852    0.000    0.987    0.695
##    .ITEM17          0.375    0.056    6.635    0.000    0.375    0.517
##    .ITEM18          0.909    0.143    6.376    0.000    0.909    0.562
##    .ITEM19          0.844    0.111    7.622    0.000    0.844    0.598
##    .ITEM21          1.245    0.133    9.338    0.000    1.245    0.778
##     EMO             1.625    0.148   11.004    0.000    1.000    1.000
##     DEP             0.705    0.158    4.452    0.000    1.000    1.000
##     ACC             0.193    0.050    3.839    0.000    1.000    1.000
```

However, displaying the modification indices via `summary()` generates a lot of output that can be hard to read. We may also extract the fit measures and the modification indices using the functions `fitmeasures()`

and `modificationIndices()`. For example, `modificationIndices()` allows use to sort based on the value of the modification index and to only show those values above a certain threshold.

```r
modificationIndices(model_ex_1_fit, minimum.value = 10, sort. = TRUE)
```

```
##        lhs op    rhs     mi    epc sepc.lv sepc.all sepc.nox
## 158  ITEM6 ~~ ITEM16 91.282  0.733   0.733    0.529    0.529
## 95   ITEM1 ~~  ITEM2 82.448  0.613   0.613    0.549    0.549
## 59     EMO =~ ITEM12 41.517 -0.313  -0.400   -0.335   -0.335
## 260 ITEM10 ~~ ITEM11 38.081  0.580   0.580    0.525    0.525
## 310  ITEM7 ~~ ITEM21 33.529  0.263   0.263    0.326    0.326
## 298  ITEM4 ~~  ITEM7 33.432  0.209   0.209    0.324    0.324
## 81     ACC =~  ITEM1 28.732  0.872   0.383    0.231    0.231
## 323 ITEM18 ~~ ITEM19 18.607  0.250   0.250    0.285    0.285
## 160  ITEM6 ~~  ITEM5 17.193  0.354   0.354    0.232    0.232
## 250  ITEM5 ~~ ITEM15 15.584  0.313   0.313    0.243    0.243
## 150  ITEM3 ~~ ITEM12 15.511 -0.255  -0.255   -0.225   -0.225
## 176  ITEM8 ~~ ITEM20 14.211  0.230   0.230    0.240    0.240
## 76     DEP =~ ITEM12 14.168 -0.329  -0.276   -0.232   -0.232
## 304  ITEM4 ~~ ITEM21 13.102  0.201   0.201    0.201    0.201
## 192 ITEM13 ~~ ITEM20 13.066  0.237   0.237    0.214    0.214
## 82     ACC =~  ITEM2 12.690  0.565   0.248    0.161    0.161
## 86     ACC =~ ITEM13 12.656 -0.583  -0.256   -0.152   -0.152
## 308  ITEM7 ~~ ITEM18 11.815 -0.145  -0.145   -0.211   -0.211
## 157  ITEM6 ~~ ITEM14 11.329 -0.311  -0.311   -0.186   -0.186
## 119  ITEM2 ~~ ITEM13 10.340 -0.219  -0.219   -0.195   -0.195
## 99   ITEM1 ~~ ITEM13 10.257 -0.225  -0.225   -0.199   -0.199
```

Similarly, we can also extract the fit measures via `fitmeasures()`.

```r
fitmeasures(model_ex_1_fit)
```

```
##                          npar                          fmin
##                        47.000                         0.935
##                         chisq                            df
##                       695.719                       206.000
##                        pvalue                  chisq.scaled
##                         0.000                       567.753
##                     df.scaled                 pvalue.scaled
##                       206.000                         0.000
##          chisq.scaling.factor                baseline.chisq
##                         1.225                      3452.269
##                   baseline.df               baseline.pvalue
##                       231.000                         0.000
##         baseline.chisq.scaled            baseline.df.scaled
##                      2911.466                       231.000
##        baseline.pvalue.scaled baseline.chisq.scaling.factor
##                         0.000                         1.186
##                           cfi                           tli
##                         0.848                         0.830
##                          nnfi                           rfi
##                         0.830                         0.774
##                           nfi                          pnfi
```

```
##                          0.798                        0.712
##                            ifi                          rni
##                          0.849                        0.848
##                     cfi.scaled                   tli.scaled
##                          0.865                        0.849
##                     cfi.robust                   tli.robust
##                          0.861                        0.844
##                    nnfi.scaled                  nnfi.robust
##                          0.849                        0.844
##                     rfi.scaled                   nfi.scaled
##                          0.781                        0.805
##                     ifi.scaled                   rni.scaled
##                          0.866                        0.865
##                     rni.robust                         logl
##                          0.861                   -12811.043
##              unrestricted.logl                          aic
##                     -12463.184                    25716.087
##                            bic                       ntotal
##                      25900.275                      372.000
##                           bic2                        rmsea
##                      25751.158                        0.080
##                 rmsea.ci.lower               rmsea.ci.upper
##                          0.073                        0.087
##                   rmsea.pvalue                 rmsea.scaled
##                          0.000                        0.069
##          rmsea.ci.lower.scaled        rmsea.ci.upper.scaled
##                          0.063                        0.075
##            rmsea.pvalue.scaled                 rmsea.robust
##                          0.000                        0.076
##          rmsea.ci.lower.robust        rmsea.ci.upper.robust
##                          0.069                        0.084
##            rmsea.pvalue.robust                          rmr
##                             NA                        0.141
##                     rmr_nomean                         srmr
##                          0.141                        0.073
##                   srmr_bentler          srmr_bentler_nomean
##                          0.073                        0.073
##                           crmr                  crmr_nomean
##                          0.076                        0.076
##                     srmr_mplus            srmr_mplus_nomean
##                          0.073                        0.073
##                          cn_05                        cn_01
##                        129.587                      137.957
##                            gfi                         agfi
##                          0.849                        0.815
##                           pgfi                          mfi
##                          0.691                        0.518
##                           ecvi
##                          2.123
```

Based on this information, the hypothesis that the model exactly reproduces the data must be rejected.

Other fit indices also indicate weak model fit.

c. Do you see possibilities to improve the fit of the model? Which one(s)? What would be your strategy for improving the fit of this model?

Inspection of modification indices shows some high values:

Table 1: High modification indices for model `model_ex_1`.

Symbol	Value	Type
`ITEM1 ~~  ITEM2`	82.448	error covariance
`ITEM6 ~~ ITEM16`	91.282	error covariance
`EMO =~ ITEM12`	41.517	cross-loading

We start with the highest value and add that relationship to the model syntax. Then we re-estimate the model with the newly added relationship and evaluate the model fit and the newly computed modification indices. We can then repeat the whole process.

d. Implement the model improvements and test if the improved model is significant using the Likelihood Ratio Test (LRT).
   - *Note: strictly speaking, the standard LRT is not correct when the Robust Maximum Likelihood is used because the scaled χ^2 values are not χ^2 distributed. However, for the sake of the exercise we will proceed this way despite of this limitation.*

We add `ITEM6 ~~ ITEM16` to the model syntax and re-estimate the model and evaluate the model fit, fit measures and modification indices.

```r
# Model syntax.
model_ex_1_modification_1 <- "
    # Measurement part.
    EMO =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM8 + ITEM13 + ITEM14 + ITEM16 + ITEM20
    DEP =~ ITEM5 + ITEM10 + ITEM11 + ITEM15 + ITEM22
    ACC =~ ITEM4 + ITEM7 + ITEM9 + ITEM12 + ITEM17 + ITEM18 + ITEM19 + ITEM21

    # Covariances between latent variables.
    EMO ~~ DEP
    DEP ~~ ACC
    EMO ~~ ACC

    # Covariance between error terms.
    ITEM6 ~~ ITEM16
"

# Fit the model.
model_ex_1_modification_1_fit <- cfa(model_ex_1_modification_1, data = data,  estimator = "MLM")

# Model summary.
summary(model_ex_1_modification_1_fit, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-12 ended normally after 48 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        48
##
##   Number of observations                           372
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               597.731     493.398
##   Degrees of freedom                               205         205
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.211
##     Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                              3452.269    2911.466
##   Degrees of freedom                               231         231
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.186
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.878       0.892
##   Tucker-Lewis Index (TLI)                       0.863       0.879
##
##   Robust Comparative Fit Index (CFI)                         0.890
##   Robust Tucker-Lewis Index (TLI)                            0.876
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -12762.049  -12762.049
##   Loglikelihood unrestricted model (H1)     -12463.184  -12463.184
##
##   Akaike (AIC)                               25620.098   25620.098
##   Bayesian (BIC)                             25808.205   25808.205
##   Sample-size adjusted Bayesian (BIC)        25655.916   25655.916
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.072       0.061
##   90 Percent confidence interval - lower         0.065       0.055
##   90 Percent confidence interval - upper         0.078       0.068
##   P-value RMSEA <= 0.05                          0.000       0.002
##
##   Robust RMSEA                                               0.068
##   90 Percent confidence interval - lower                     0.060
##   90 Percent confidence interval - upper                     0.075
##
```

```
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.071       0.071
##
## Parameter Estimates:
##
##   Standard errors                             Robust.sem
##   Information                                   Expected
##   Information saturated (h1) model            Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO =~
##     ITEM1             1.000                               1.289    0.777
##     ITEM2             0.887    0.040   22.303    0.000    1.143    0.741
##     ITEM3             1.015    0.052   19.632    0.000    1.309    0.756
##     ITEM6             0.715    0.069   10.369    0.000    0.921    0.582
##     ITEM8             1.133    0.058   19.698    0.000    1.460    0.846
##     ITEM13            1.002    0.062   16.227    0.000    1.291    0.768
##     ITEM14            0.847    0.058   14.692    0.000    1.092    0.633
##     ITEM16            0.672    0.065   10.294    0.000    0.866    0.602
##     ITEM20            0.746    0.061   12.288    0.000    0.962    0.681
##   DEP =~
##     ITEM5             1.000                               0.835    0.562
##     ITEM10            1.151    0.154    7.473    0.000    0.961    0.665
##     ITEM11            1.363    0.164    8.329    0.000    1.138    0.745
##     ITEM15            0.909    0.124    7.351    0.000    0.759    0.585
##     ITEM22            0.771    0.123    6.252    0.000    0.644    0.408
##   ACC =~
##     ITEM4             1.000                               0.439    0.441
##     ITEM7             0.969    0.128    7.564    0.000    0.426    0.508
##     ITEM9             1.779    0.322    5.529    0.000    0.782    0.595
##     ITEM12            1.496    0.240    6.232    0.000    0.657    0.551
##     ITEM17            1.347    0.199    6.756    0.000    0.592    0.695
##     ITEM18            1.917    0.298    6.441    0.000    0.842    0.662
##     ITEM19            1.714    0.287    5.979    0.000    0.753    0.634
##     ITEM21            1.356    0.227    5.985    0.000    0.596    0.471
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO ~~
##     DEP               0.697    0.106    6.605    0.000    0.648    0.648
##   DEP ~~
##     ACC              -0.171    0.036   -4.768    0.000   -0.466   -0.466
##   EMO ~~
##     ACC              -0.188    0.040   -4.670    0.000   -0.333   -0.333
##  .ITEM6 ~~
##    .ITEM16            0.733    0.121    6.069    0.000    0.733    0.497
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
```

```
##    .ITEM1            1.091    0.092   11.824    0.000    1.091    0.396
##    .ITEM2            1.076    0.088   12.219    0.000    1.076    0.452
##    .ITEM3            1.283    0.105   12.211    0.000    1.283    0.428
##    .ITEM6            1.654    0.141   11.710    0.000    1.654    0.661
##    .ITEM8            0.844    0.080   10.559    0.000    0.844    0.284
##    .ITEM13           1.156    0.129    8.945    0.000    1.156    0.409
##    .ITEM14           1.780    0.141   12.655    0.000    1.780    0.599
##    .ITEM16           1.317    0.115   11.413    0.000    1.317    0.637
##    .ITEM20           1.071    0.136    7.863    0.000    1.071    0.536
##    .ITEM5            1.511    0.180    8.414    0.000    1.511    0.684
##    .ITEM10           1.164    0.147    7.927    0.000    1.164    0.558
##    .ITEM11           1.038    0.141    7.364    0.000    1.038    0.445
##    .ITEM15           1.108    0.153    7.225    0.000    1.108    0.658
##    .ITEM22           2.077    0.184   11.269    0.000    2.077    0.834
##    .ITEM4            0.801    0.112    7.124    0.000    0.801    0.806
##    .ITEM7            0.523    0.075    7.011    0.000    0.523    0.742
##    .ITEM9            1.116    0.149    7.484    0.000    1.116    0.646
##    .ITEM12           0.988    0.126    7.855    0.000    0.988    0.696
##    .ITEM17           0.375    0.056    6.636    0.000    0.375    0.517
##    .ITEM18           0.909    0.143    6.376    0.000    0.909    0.562
##    .ITEM19           0.844    0.111    7.626    0.000    0.844    0.598
##    .ITEM21           1.244    0.133    9.339    0.000    1.244    0.778
##     EMO              1.662    0.148   11.216    0.000    1.000    1.000
##     DEP              0.697    0.158    4.424    0.000    1.000    1.000
##     ACC              0.193    0.050    3.842    0.000    1.000    1.000
```

```r
# Modification indices.
modificationIndices(model_ex_1_modification_1_fit, minimum.value = 10, sort. = TRUE)
```

```
##        lhs op    rhs     mi    epc sepc.lv sepc.all sepc.nox
## 96   ITEM1 ~~  ITEM2 78.275  0.591   0.591    0.545    0.545
## 60     EMO =~ ITEM12 41.936 -0.310  -0.400   -0.336   -0.336
## 260 ITEM10 ~~ ITEM11 37.348  0.578   0.578    0.526    0.526
## 310  ITEM7 ~~ ITEM21 33.497  0.263   0.263    0.326    0.326
## 298  ITEM4 ~~  ITEM7 33.386  0.209   0.209    0.323    0.323
## 82     ACC =~  ITEM1 28.188  0.851   0.374    0.225    0.225
## 323 ITEM18 ~~ ITEM19 18.617  0.250   0.250    0.285    0.285
## 250  ITEM5 ~~ ITEM15 16.067  0.318   0.318    0.246    0.246
## 151  ITEM3 ~~ ITEM12 15.294 -0.253  -0.253   -0.225   -0.225
## 87     ACC =~ ITEM13 14.632 -0.628  -0.276   -0.164   -0.164
## 77     DEP =~ ITEM12 14.187 -0.331  -0.276   -0.232   -0.232
## 176  ITEM8 ~~ ITEM20 13.662  0.227   0.227    0.239    0.239
## 192 ITEM13 ~~ ITEM20 13.398  0.242   0.242    0.218    0.218
## 304  ITEM4 ~~ ITEM21 13.065  0.200   0.200    0.200    0.200
## 100  ITEM1 ~~ ITEM13 12.399 -0.249  -0.249   -0.221   -0.221
## 120  ITEM2 ~~ ITEM13 12.103 -0.237  -0.237   -0.213   -0.213
## 83     ACC =~  ITEM2 12.083  0.543   0.239    0.155    0.155
## 308  ITEM7 ~~ ITEM18 11.880 -0.146  -0.146   -0.211   -0.211
## 123  ITEM2 ~~ ITEM20 11.866 -0.216  -0.216   -0.202   -0.202
## 160  ITEM6 ~~  ITEM5 10.713  0.246   0.246    0.156    0.156
```

Now we perform a LRT to see whether the modification we made improves the model fit significantly.

```r
# LRT test via `anova()`.
anova(model_ex_1_fit, model_ex_1_modification_1_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                               Df   AIC   BIC  Chisq Chisq diff Df diff
## model_ex_1_modification_1_fit 205 25620 25808 597.73
## model_ex_1_fit                206 25716 25900 695.72     24.007       1
##                               Pr(>Chisq)
## model_ex_1_modification_1_fit
## model_ex_1_fit                   9.6e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We may also use a `lavaan` specific function, namely `lavTestLRT()`, which will produce the same output. See `?lavTestLRT` for my information.

```r
lavTestLRT(model_ex_1_fit, model_ex_1_modification_1_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                               Df   AIC   BIC  Chisq Chisq diff Df diff
## model_ex_1_modification_1_fit 205 25620 25808 597.73
## model_ex_1_fit                206 25716 25900 695.72     24.007       1
##                               Pr(>Chisq)
## model_ex_1_modification_1_fit
## model_ex_1_fit                   9.6e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that the model with added error covariance fits significantly better than the constrained model.

Based on the newly computed modification indices, the highest value is 78.275 for the error covariance between `ITEM1 ~~ ITEM2`. We proceed again by adding this covariance to the model and repeating the steps above.

```r
# Model syntax.
model_ex_1_modification_2 <- "
    # Measurement part.
    EMO =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM8 + ITEM13 + ITEM14 + ITEM16 + ITEM20
    DEP =~ ITEM5 + ITEM10 + ITEM11 + ITEM15 + ITEM22
    ACC =~ ITEM4 + ITEM7 + ITEM9 + ITEM12 + ITEM17 + ITEM18 + ITEM19 + ITEM21

```

```
    # Covariances between latent variables.
    EMO ~~ DEP
    DEP ~~ ACC
    EMO ~~ ACC

    # Covariances between error terms.
    ITEM6 ~~ ITEM16
    ITEM1 ~~ ITEM2
"

# Fit the model.
model_ex_1_modification_2_fit <- cfa(model_ex_1_modification_2, data = data,  estimator = "MLM")

# Model summary.
summary(model_ex_1_modification_2_fit, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-12 ended normally after 46 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        49
##
##   Number of observations                           372
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               520.481     431.496
##   Degrees of freedom                               204         204
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.206
##     Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                              3452.269    2911.466
##   Degrees of freedom                               231         231
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.186
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.902       0.915
##   Tucker-Lewis Index (TLI)                       0.889       0.904
##
##   Robust Comparative Fit Index (CFI)                         0.914
##   Robust Tucker-Lewis Index (TLI)                            0.902
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -12723.424  -12723.424
##   Loglikelihood unrestricted model (H1)     -12463.184  -12463.184
```

```
##
##   Akaike (AIC)                               25544.849   25544.849
##   Bayesian (BIC)                             25736.875   25736.875
##   Sample-size adjusted Bayesian (BIC)        25581.413   25581.413
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.065       0.055
##   90 Percent confidence interval - lower         0.058       0.048
##   90 Percent confidence interval - upper         0.071       0.061
##   P-value RMSEA <= 0.05                          0.000       0.114
##
##   Robust RMSEA                                               0.060
##   90 Percent confidence interval - lower                     0.052
##   90 Percent confidence interval - upper                     0.068
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.069       0.069
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO =~
##     ITEM1             1.000                               1.215    0.732
##     ITEM2             0.877    0.041   21.156    0.000    1.066    0.691
##     ITEM3             1.068    0.059   18.221    0.000    1.298    0.750
##     ITEM6             0.767    0.077   10.025    0.000    0.933    0.590
##     ITEM8             1.216    0.067   18.238    0.000    1.478    0.857
##     ITEM13            1.086    0.069   15.688    0.000    1.320    0.785
##     ITEM14            0.884    0.063   14.109    0.000    1.074    0.623
##     ITEM16            0.727    0.072   10.053    0.000    0.883    0.614
##     ITEM20            0.811    0.067   12.137    0.000    0.986    0.698
##   DEP =~
##     ITEM5             1.000                               0.835    0.562
##     ITEM10            1.151    0.154    7.478    0.000    0.960    0.665
##     ITEM11            1.363    0.163    8.346    0.000    1.138    0.745
##     ITEM15            0.910    0.124    7.363    0.000    0.760    0.586
##     ITEM22            0.769    0.123    6.264    0.000    0.642    0.407
##   ACC =~
##     ITEM4             1.000                               0.439    0.440
##     ITEM7             0.969    0.128    7.566    0.000    0.425    0.507
##     ITEM9             1.782    0.323    5.524    0.000    0.782    0.595
##     ITEM12            1.505    0.241    6.239    0.000    0.660    0.554
##     ITEM17            1.349    0.200    6.759    0.000    0.592    0.695
##     ITEM18            1.919    0.298    6.431    0.000    0.842    0.662
```

```
##     ITEM19            1.718    0.287    5.979    0.000    0.753    0.634
##     ITEM21            1.356    0.227    5.977    0.000    0.595    0.470
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO ~~
##     DEP               0.672    0.103    6.524    0.000    0.662    0.662
##   DEP ~~
##     ACC              -0.171    0.036   -4.764    0.000   -0.466   -0.466
##   EMO ~~
##     ACC              -0.193    0.039   -4.914    0.000   -0.363   -0.363
##  .ITEM6 ~~
##    .ITEM16            0.708    0.122    5.804    0.000    0.708    0.488
##  .ITEM1 ~~
##    .ITEM2             0.596    0.087    6.891    0.000    0.596    0.473
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .ITEM1             1.276    0.105   12.210    0.000    1.276    0.464
##    .ITEM2             1.246    0.098   12.669    0.000    1.246    0.523
##    .ITEM3             1.312    0.110   11.931    0.000    1.312    0.438
##    .ITEM6             1.633    0.143   11.444    0.000    1.633    0.652
##    .ITEM8             0.793    0.083    9.559    0.000    0.793    0.266
##    .ITEM13            1.081    0.124    8.712    0.000    1.081    0.383
##    .ITEM14            1.819    0.145   12.567    0.000    1.819    0.612
##    .ITEM16            1.287    0.117   11.021    0.000    1.287    0.623
##    .ITEM20            1.024    0.136    7.506    0.000    1.024    0.513
##    .ITEM5             1.511    0.179    8.429    0.000    1.511    0.684
##    .ITEM10            1.165    0.147    7.912    0.000    1.165    0.558
##    .ITEM11            1.037    0.140    7.392    0.000    1.037    0.445
##    .ITEM15            1.106    0.153    7.238    0.000    1.106    0.657
##    .ITEM22            2.079    0.184   11.292    0.000    2.079    0.835
##    .ITEM4             0.802    0.113    7.126    0.000    0.802    0.807
##    .ITEM7             0.523    0.075    7.018    0.000    0.523    0.743
##    .ITEM9             1.116    0.149    7.495    0.000    1.116    0.646
##    .ITEM12            0.985    0.125    7.853    0.000    0.985    0.693
##    .ITEM17            0.375    0.056    6.643    0.000    0.375    0.517
##    .ITEM18            0.909    0.143    6.374    0.000    0.909    0.562
##    .ITEM19            0.844    0.110    7.636    0.000    0.844    0.598
##    .ITEM21            1.245    0.133    9.334    0.000    1.245    0.779
##     EMO               1.477    0.150    9.869    0.000    1.000    1.000
##     DEP               0.697    0.157    4.428    0.000    1.000    1.000
##     ACC               0.192    0.050    3.836    0.000    1.000    1.000
```

```
# Modification indices.
modificationIndices(model_ex_1_modification_2_fit, minimum.value = 10, sort. = TRUE)
```

```
##        lhs op    rhs     mi    epc sepc.lv sepc.all sepc.nox
## 61     EMO =~ ITEM12 41.026 -0.332  -0.404   -0.339   -0.339
## 260 ITEM10 ~~ ITEM11 37.190  0.575   0.575    0.523    0.523
## 310  ITEM7 ~~ ITEM21 33.636  0.264   0.264    0.327    0.327
## 298  ITEM4 ~~  ITEM7 33.523  0.210   0.210    0.324    0.324
```

```
## 323 ITEM18 ~~ ITEM19 18.591  0.250   0.250    0.285    0.285
## 151  ITEM3 ~~ ITEM12 16.431 -0.265  -0.265   -0.233   -0.233
## 250  ITEM5 ~~ ITEM15 15.931  0.316   0.316    0.245    0.245
## 83     ACC =~  ITEM1 14.440  0.560   0.246    0.148    0.148
## 78     DEP =~ ITEM12 14.001 -0.329  -0.274   -0.230   -0.230
## 97   ITEM1 ~~  ITEM3 13.922  0.248   0.248    0.192    0.192
## 304  ITEM4 ~~ ITEM21 13.153  0.201   0.201    0.201    0.201
## 308  ITEM7 ~~ ITEM18 11.652 -0.144  -0.144   -0.209   -0.209
## 160  ITEM6 ~~  ITEM5 10.791  0.247   0.247    0.157    0.157
```

We perform another LRT, this time between models `model_ex_1_modification_1` and `model_ex_1_modification_2`.

```
# LRT test via `anova()`.
anova(model_ex_1_modification_1_fit, model_ex_1_modification_2_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                               Df   AIC   BIC  Chisq Chisq diff Df diff
## model_ex_1_modification_2_fit 204 25545 25737 520.48
## model_ex_1_modification_1_fit 205 25620 25808 597.73     33.902       1
##                              Pr(>Chisq)
## model_ex_1_modification_2_fit
## model_ex_1_modification_1_fit  5.795e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The modification still improves the model fit significantly. We repeat the steps again and add the relationship with the highest modification value as reported by `lavaan`. In this case this relationship is the error covariance between items `ITEM10 ~~ ITEM11` with a value a modification value of 37.190.

```
# Model syntax.
model_ex_1_modification_3 <- "
    # Measurement part.
    EMO =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM8 + ITEM13 + ITEM14 + ITEM16 + ITEM20
    DEP =~ ITEM5 + ITEM10 + ITEM11 + ITEM15 + ITEM22
    ACC =~ ITEM4 + ITEM7 + ITEM9 + ITEM12 + ITEM17 + ITEM18 + ITEM19 + ITEM21

    # Covariances between latent variables.
    EMO ~~ DEP
    DEP ~~ ACC
    EMO ~~ ACC

    # Covariances between error terms.
    ITEM6 ~~ ITEM16
    ITEM1 ~~ ITEM2
    ITEM10 ~~ ITEM11
"
```

```r
# Fit the model.
model_ex_1_modification_3_fit <- cfa(model_ex_1_modification_3, data = data,  estimator = "MLM")

# Model summary.
summary(model_ex_1_modification_3_fit, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-12 ended normally after 47 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        50
##
##   Number of observations                           372
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               487.893     403.049
##   Degrees of freedom                               203         203
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.211
##     Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                              3452.269    2911.466
##   Degrees of freedom                               231         231
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.186
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.912       0.925
##   Tucker-Lewis Index (TLI)                       0.899       0.915
##
##   Robust Comparative Fit Index (CFI)                         0.924
##   Robust Tucker-Lewis Index (TLI)                            0.913
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -12707.131  -12707.131
##   Loglikelihood unrestricted model (H1)     -12463.184  -12463.184
##
##   Akaike (AIC)                               25514.261   25514.261
##   Bayesian (BIC)                             25710.206   25710.206
##   Sample-size adjusted Bayesian (BIC)        25551.571   25551.571
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.061       0.051
##   90 Percent confidence interval - lower         0.054       0.045
##   90 Percent confidence interval - upper         0.068       0.058
```

```
##   P-value RMSEA <= 0.05                          0.004       0.351
##
##   Robust RMSEA                                               0.057
##   90 Percent confidence interval - lower                     0.049
##   90 Percent confidence interval - upper                     0.065
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.068       0.068
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO =~
##     ITEM1             1.000                               1.215    0.732
##     ITEM2             0.878    0.042   21.148    0.000    1.067    0.691
##     ITEM3             1.068    0.058   18.270    0.000    1.298    0.750
##     ITEM6             0.770    0.077   10.070    0.000    0.936    0.592
##     ITEM8             1.214    0.067   18.174    0.000    1.476    0.855
##     ITEM13            1.086    0.069   15.677    0.000    1.320    0.786
##     ITEM14            0.885    0.063   14.089    0.000    1.075    0.624
##     ITEM16            0.727    0.072   10.070    0.000    0.884    0.615
##     ITEM20            0.811    0.067   12.121    0.000    0.986    0.698
##   DEP =~
##     ITEM5             1.000                               0.896    0.603
##     ITEM10            0.886    0.123    7.205    0.000    0.794    0.550
##     ITEM11            1.102    0.129    8.559    0.000    0.987    0.646
##     ITEM15            0.919    0.119    7.716    0.000    0.823    0.635
##     ITEM22            0.776    0.116    6.685    0.000    0.696    0.441
##   ACC =~
##     ITEM4             1.000                               0.438    0.440
##     ITEM7             0.976    0.129    7.581    0.000    0.428    0.510
##     ITEM9             1.783    0.324    5.504    0.000    0.782    0.595
##     ITEM12            1.499    0.240    6.241    0.000    0.657    0.552
##     ITEM17            1.348    0.199    6.771    0.000    0.591    0.694
##     ITEM18            1.917    0.298    6.430    0.000    0.840    0.661
##     ITEM19            1.724    0.289    5.971    0.000    0.756    0.636
##     ITEM21            1.356    0.228    5.947    0.000    0.594    0.470
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO ~~
##     DEP               0.751    0.106    7.096    0.000    0.689    0.689
##   DEP ~~
##     ACC              -0.190    0.039   -4.890    0.000   -0.484   -0.484
##   EMO ~~
```

```
##     ACC             -0.193    0.039   -4.907    0.000   -0.362   -0.362
##  .ITEM6 ~~
##    .ITEM16           0.703    0.122    5.769    0.000    0.703    0.487
##  .ITEM1 ~~
##    .ITEM2            0.596    0.086    6.905    0.000    0.596    0.473
##  .ITEM10 ~~
##    .ITEM11           0.519    0.110    4.731    0.000    0.519    0.369
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .ITEM1            1.276    0.105   12.182    0.000    1.276    0.464
##    .ITEM2            1.245    0.098   12.697    0.000    1.245    0.523
##    .ITEM3            1.313    0.110   11.905    0.000    1.313    0.438
##    .ITEM6            1.626    0.142   11.432    0.000    1.626    0.650
##    .ITEM8            0.799    0.083    9.635    0.000    0.799    0.268
##    .ITEM13           1.080    0.124    8.705    0.000    1.080    0.383
##    .ITEM14           1.817    0.144   12.585    0.000    1.817    0.611
##    .ITEM16           1.285    0.117   11.015    0.000    1.285    0.622
##    .ITEM20           1.024    0.136    7.510    0.000    1.024    0.513
##    .ITEM5            1.404    0.180    7.787    0.000    1.404    0.636
##    .ITEM10           1.457    0.150    9.726    0.000    1.457    0.698
##    .ITEM11           1.358    0.159    8.532    0.000    1.358    0.582
##    .ITEM15           1.005    0.141    7.110    0.000    1.005    0.597
##    .ITEM22           2.006    0.182   11.019    0.000    2.006    0.806
##    .ITEM4            0.802    0.113    7.124    0.000    0.802    0.807
##    .ITEM7            0.521    0.074    7.007    0.000    0.521    0.740
##    .ITEM9            1.116    0.149    7.490    0.000    1.116    0.646
##    .ITEM12           0.988    0.125    7.890    0.000    0.988    0.696
##    .ITEM17           0.376    0.057    6.641    0.000    0.376    0.519
##    .ITEM18           0.912    0.143    6.383    0.000    0.912    0.564
##    .ITEM19           0.840    0.110    7.606    0.000    0.840    0.595
##    .ITEM21           1.246    0.133    9.333    0.000    1.246    0.779
##     EMO              1.477    0.150    9.864    0.000    1.000    1.000
##     DEP              0.803    0.171    4.706    0.000    1.000    1.000
##     ACC              0.192    0.050    3.828    0.000    1.000    1.000
```

```r
# Modification indices.
modificationIndices(model_ex_1_modification_3_fit, minimum.value = 10, sort. = TRUE)
```

```
##        lhs op    rhs     mi    epc sepc.lv sepc.all sepc.nox
## 62     EMO =~ ITEM12 40.621 -0.331  -0.402   -0.337   -0.337
## 310  ITEM7 ~~ ITEM21 33.404  0.262   0.262    0.326    0.326
## 298  ITEM4 ~~  ITEM7 33.318  0.209   0.209    0.323    0.323
## 323 ITEM18 ~~ ITEM19 18.400  0.248   0.248    0.284    0.284
## 152  ITEM3 ~~ ITEM12 16.749 -0.268  -0.268   -0.236   -0.236
## 84     ACC =~  ITEM1 14.481  0.561   0.246    0.148    0.148
## 79     DEP =~ ITEM12 14.270 -0.324  -0.290   -0.243   -0.243
## 98   ITEM1 ~~  ITEM3 13.974  0.249   0.249    0.192    0.192
## 304  ITEM4 ~~ ITEM21 13.190  0.201   0.201    0.201    0.201
## 308  ITEM7 ~~ ITEM18 12.056 -0.147  -0.147   -0.213   -0.213
## 303  ITEM4 ~~ ITEM19 10.108 -0.154  -0.154   -0.187   -0.187
```

```r
# LRT test via `anova()`.
anova(model_ex_1_modification_2_fit, model_ex_1_modification_3_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                                Df   AIC   BIC  Chisq Chisq diff Df diff
## model_ex_1_modification_3_fit 203 25514 25710 487.89
## model_ex_1_modification_2_fit 204 25545 25737 520.48      96.56       1
##                               Pr(>Chisq)
## model_ex_1_modification_3_fit
## model_ex_1_modification_2_fit  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The modification still improves the model fit significantly. Based on the values reported by `lavaan` for the modification indices, the next addition we make is adding the cross-loading between `EMO ~= ITEM12`.

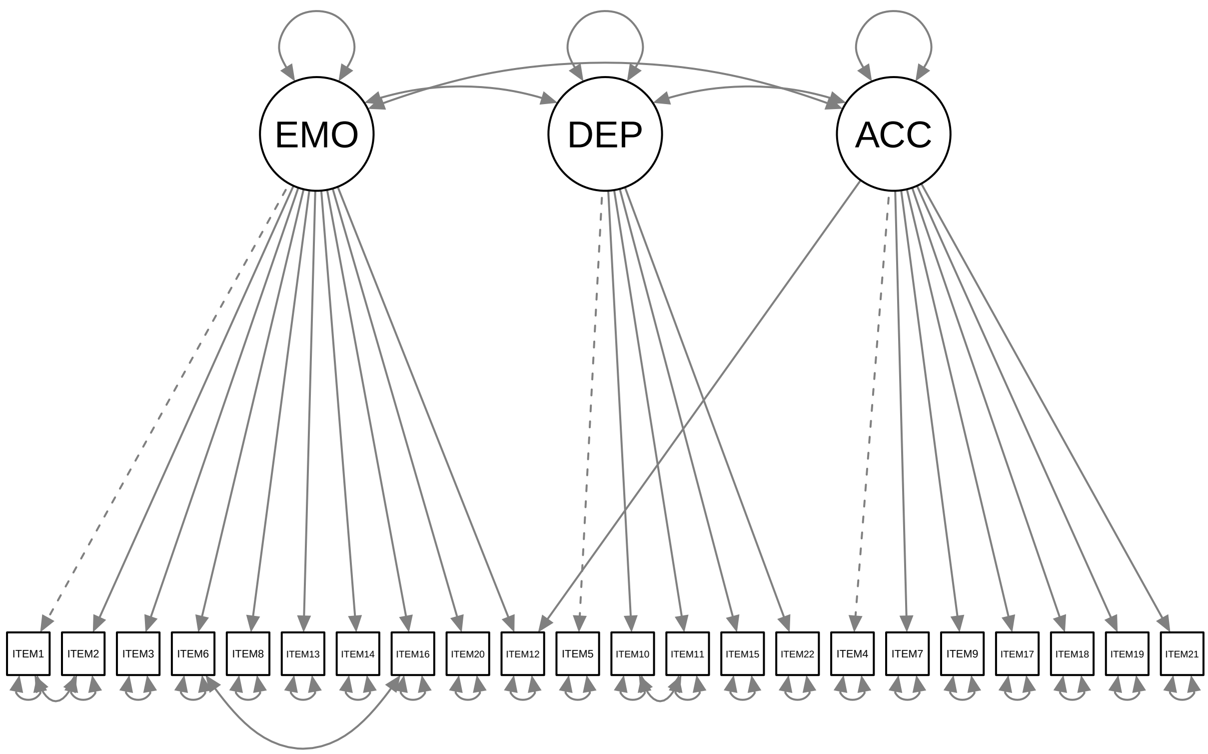
```r
# Model syntax.
model_ex_1_modification_4 <- "
    # Measurement part.
    EMO =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM8 + ITEM13 + ITEM14 + ITEM16 + ITEM20 + ITEM12
    DEP =~ ITEM5 + ITEM10 + ITEM11 + ITEM15 + ITEM22
    ACC =~ ITEM4 + ITEM7 + ITEM9 + ITEM12 + ITEM17 + ITEM18 + ITEM19 + ITEM21

    # Covariances between latent variables.
    EMO ~~ DEP
    DEP ~~ ACC
    EMO ~~ ACC

    # Covariances between error terms.
    ITEM6 ~~ ITEM16
    ITEM1 ~~ ITEM2
    ITEM10 ~~ ITEM11
"

# Fit the model.
model_ex_1_modification_4_fit <- cfa(model_ex_1_modification_4, data = data,  estimator = "MLM")

# Visualize the model.
semPaths(model_ex_1_modification_4_fit, what = "paths", sizeMan = 3)
```




```r
# Model summary.
summary(model_ex_1_modification_4_fit, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-12 ended normally after 52 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        51
##
##   Number of observations                           372
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               446.419     369.998
##   Degrees of freedom                               202         202
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.207
##     Satorra-Bentler correction
##
## Model Test Baseline Model:
##
##   Test statistic                              3452.269    2911.466
##   Degrees of freedom                               231         231
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.186
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.924       0.937
```

```
##   Tucker-Lewis Index (TLI)                       0.913       0.928
##
##   Robust Comparative Fit Index (CFI)                         0.936
##   Robust Tucker-Lewis Index (TLI)                            0.927
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)            -12686.394  -12686.394
##   Loglikelihood unrestricted model (H1)    -12463.184  -12463.184
##
##   Akaike (AIC)                              25474.787   25474.787
##   Bayesian (BIC)                            25674.651   25674.651
##   Sample-size adjusted Bayesian (BIC)       25512.844   25512.844
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.057       0.047
##   90 Percent confidence interval - lower         0.050       0.040
##   90 Percent confidence interval - upper         0.064       0.054
##   P-value RMSEA <= 0.05                          0.052       0.735
##
##   Robust RMSEA                                              0.052
##   90 Percent confidence interval - lower                    0.044
##   90 Percent confidence interval - upper                    0.060
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.057       0.057
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO =~
##     ITEM1             1.000                               1.219    0.735
##     ITEM2             0.878    0.041   21.316    0.000    1.070    0.693
##     ITEM3             1.073    0.058   18.460    0.000    1.308    0.756
##     ITEM6             0.764    0.076    9.992    0.000    0.931    0.589
##     ITEM8             1.215    0.066   18.382    0.000    1.481    0.859
##     ITEM13            1.072    0.070   15.415    0.000    1.307    0.778
##     ITEM14            0.880    0.063   14.071    0.000    1.072    0.622
##     ITEM16            0.727    0.072   10.032    0.000    0.886    0.616
##     ITEM20            0.806    0.066   12.127    0.000    0.983    0.696
##     ITEM12           -0.316    0.054   -5.890    0.000   -0.385   -0.323
##   DEP =~
##     ITEM5             1.000                               0.895    0.602
##     ITEM10            0.889    0.124    7.178    0.000    0.795    0.551
```

```
##     ITEM11            1.105    0.130    8.530    0.000    0.989    0.647
##     ITEM15            0.921    0.120    7.671    0.000    0.824    0.635
##     ITEM22            0.776    0.116    6.668    0.000    0.695    0.440
##   ACC =~
##     ITEM4             1.000                               0.446    0.448
##     ITEM7             0.973    0.128    7.602    0.000    0.435    0.518
##     ITEM9             1.763    0.317    5.561    0.000    0.787    0.599
##     ITEM12            1.131    0.202    5.607    0.000    0.505    0.424
##     ITEM17            1.327    0.198    6.717    0.000    0.592    0.696
##     ITEM18            1.890    0.291    6.497    0.000    0.844    0.663
##     ITEM19            1.695    0.286    5.933    0.000    0.757    0.637
##     ITEM21            1.342    0.224    5.993    0.000    0.599    0.474
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   EMO ~~
##     DEP               0.747    0.106    7.038    0.000    0.685    0.685
##   DEP ~~
##     ACC              -0.181    0.038   -4.788    0.000   -0.453   -0.453
##   EMO ~~
##     ACC              -0.167    0.038   -4.355    0.000   -0.306   -0.306
##  .ITEM6 ~~
##    .ITEM16            0.706    0.122    5.773    0.000    0.706    0.488
##  .ITEM1 ~~
##    .ITEM2             0.588    0.086    6.870    0.000    0.588    0.469
##  .ITEM10 ~~
##    .ITEM11            0.517    0.110    4.719    0.000    0.517    0.368
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .ITEM1             1.268    0.103   12.268    0.000    1.268    0.460
##    .ITEM2             1.238    0.098   12.631    0.000    1.238    0.520
##    .ITEM3             1.285    0.108   11.939    0.000    1.285    0.429
##    .ITEM6             1.636    0.143   11.474    0.000    1.636    0.654
##    .ITEM8             0.783    0.080    9.828    0.000    0.783    0.263
##    .ITEM13            1.115    0.128    8.693    0.000    1.115    0.395
##    .ITEM14            1.822    0.144   12.651    0.000    1.822    0.613
##    .ITEM16            1.281    0.116   11.047    0.000    1.281    0.620
##    .ITEM20            1.031    0.137    7.519    0.000    1.031    0.516
##    .ITEM12            0.898    0.105    8.557    0.000    0.898    0.632
##    .ITEM5             1.407    0.181    7.771    0.000    1.407    0.638
##    .ITEM10            1.455    0.150    9.710    0.000    1.455    0.697
##    .ITEM11            1.355    0.159    8.504    0.000    1.355    0.581
##    .ITEM15            1.004    0.142    7.094    0.000    1.004    0.596
##    .ITEM22            2.008    0.182   11.021    0.000    2.008    0.806
##    .ITEM4             0.795    0.112    7.108    0.000    0.795    0.800
##    .ITEM7             0.515    0.074    6.997    0.000    0.515    0.732
##    .ITEM9             1.108    0.150    7.407    0.000    1.108    0.641
##    .ITEM17            0.374    0.056    6.694    0.000    0.374    0.516
##    .ITEM18            0.906    0.143    6.335    0.000    0.906    0.560
##    .ITEM19            0.838    0.113    7.436    0.000    0.838    0.594
```

```
##    .ITEM21           1.240    0.132    9.366    0.000    1.240    0.776
##     EMO              1.486    0.150    9.933    0.000    1.000    1.000
##     DEP              0.800    0.171    4.684    0.000    1.000    1.000
##     ACC              0.199    0.051    3.896    0.000    1.000    1.000
```

```r
# Modification indices.
modificationIndices(model_ex_1_modification_4_fit, minimum.value = 10, sort. = TRUE)
```

```
##        lhs op    rhs     mi    epc sepc.lv sepc.all sepc.nox
## 315  ITEM7 ~~ ITEM21 32.503  0.259   0.259    0.323    0.323
## 305  ITEM4 ~~  ITEM7 32.009  0.204   0.204    0.319    0.319
## 323 ITEM18 ~~ ITEM19 18.274  0.250   0.250    0.287    0.287
## 84     ACC =~  ITEM1 14.649  0.541   0.241    0.145    0.145
## 313  ITEM7 ~~ ITEM18 14.409 -0.161  -0.161   -0.236   -0.236
## 194 ITEM13 ~~ ITEM12 13.063  0.212   0.212    0.212    0.212
## 98   ITEM1 ~~  ITEM3 12.963  0.237   0.237    0.186    0.186
## 310  ITEM4 ~~ ITEM21 12.460  0.195   0.195    0.197    0.197
## 309  ITEM4 ~~ ITEM19 11.555 -0.165  -0.165   -0.202   -0.202
## 144  ITEM3 ~~ ITEM12 11.417 -0.210  -0.210   -0.196   -0.196
## 89     ACC =~ ITEM13 10.511 -0.518  -0.231   -0.138   -0.138
## 162  ITEM6 ~~  ITEM5 10.057  0.236   0.236    0.155    0.155
```

```r
# LRT test via `anova()`.
anova(model_ex_1_modification_3_fit, model_ex_1_modification_4_fit)
```

```
## Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
##
## lavaan NOTE:
##     The "Chisq" column contains standard test statistics, not the
##     robust test that should be reported per model. A robust difference
##     test is a function of two standard (not robust) statistics.
##
##                               Df   AIC   BIC  Chisq Chisq diff Df diff
## model_ex_1_modification_4_fit 202 25475 25675 446.42
## model_ex_1_modification_3_fit 203 25514 25710 487.89     20.634       1
##                               Pr(>Chisq)
## model_ex_1_modification_4_fit
## model_ex_1_modification_3_fit   5.56e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yet again the modification improves the model fit significantly. However, we stop here, because:

- we have already added four parameters and we must not forget about the parsimony of the model
- in addition, other parameter suggestions refer to error covariances which are more difficult to justify from a substantive point of view
- `CFI` and `TLI`, and `RMSEA` and `SRMR` indicate fairly good model fit

Note. *Remember that modification indices tell us how the model fit would change if we added new parameters to the model. Since our factor model is confirmatory by nature, misusing modification indices can be dangerous. More specifically, making changes based on the modification indices can run the risk of over-fitting the data and reducing the generalizability of the results. In practice, one should make changes to the model based on* *the modification indices only when such changes can be theoretically justified.*

Note on the χ^2 difference value in the LRT.

In the past, some students have indicated that the χ^2 difference value in the LRT computation is not the same as the actual difference between the standard χ^2 reported by `lavaan` (e.g., during model `summary()`).

This seems to be the case because we estimated the models using the *Satorra Bentler* method. In this scenario, `lavaan` will use the standard χ^2 values, however it will apply a scaled test statistic using the `satorra.bentler.2001` method. This is mentioned both in the output of `lavTestLRT()` and the documentation for this function.

For example, if we run `lavTestLRT()`,

```
lavTestLRT(model_ex_1_fit, model_ex_1_modification_1_fit)
```

we see the following note:

```
Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")

lavaan NOTE:
    The "Chisq" column contains standard test statistics, not the
    robust test that should be reported per model. A robust difference
    test is a function of two standard (not robust) statistics.
```

Then, in the documentation of `lavTestLRT()` we see the following:

```
?lavTestLRT
```

```
The anova function for lavaan objects simply calls the lavTestLRT function,
which has a few additional arguments.

If `type = "Chisq"` and the test statistics are scaled, a special scaled
difference test statistic is computed. If method is `"satorra.bentler.2001"`, a
simple approximation is used described in Satorra & Bentler (2001). In some
settings, this can lead to a negative test statistic. To ensure a positive test
statistic, we can use the method proposed by Satorra & Bentler (2010).
Alternatively, when method is `"satorra.2000"`, the original formulas of Satorra
(2000) are used.
```

We know that if we use the `ML` estimator, then the χ^2 difference value in the LRT should be in fact the difference between the standard χ^2 values. In our case, we expect this difference to be the result of adding another parameter to the model based on the modification indices. We can check this as follows:

```
# Model syntax model 1.
model_1 <- "
    # Measurement part.
    EMO =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM8 + ITEM13 + ITEM14 + ITEM16 + ITEM20
    DEP =~ ITEM5 + ITEM10 + ITEM11 + ITEM15 + ITEM22
```

```
    ACC =~ ITEM4 + ITEM7 + ITEM9 + ITEM12 + ITEM17 + ITEM18 + ITEM19 + ITEM21

    # Covariances between latent variables.
    EMO ~~ DEP
    DEP ~~ ACC
    EMO ~~ ACC
"

# Model syntax model 2.
model_2 <- "
    # Measurement part.
    EMO =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM8 + ITEM13 + ITEM14 + ITEM16 + ITEM20
    DEP =~ ITEM5 + ITEM10 + ITEM11 + ITEM15 + ITEM22
    ACC =~ ITEM4 + ITEM7 + ITEM9 + ITEM12 + ITEM17 + ITEM18 + ITEM19 + ITEM21

    # Covariances between latent variables.
    EMO ~~ DEP
    DEP ~~ ACC
    EMO ~ ~ACC

    # Covariances between error terms.
    ITEM6 ~~ ITEM16
"

# Model fit model 1.
model_1_fit <- cfa(model_1, data = data, estimator = "ML")

# Model fit model 2.
model_2_fit <- cfa(model_2, data = data,  estimator = "ML")

# Print first 3 modification indices for `model_1_fit`.
modificationIndices(model_1_fit, sort. = TRUE)[1:3, ]
```

```
##       lhs op    rhs     mi    epc sepc.lv sepc.all sepc.nox
## 158 ITEM6 ~~ ITEM16 91.282  0.733   0.733    0.529    0.529
## 95  ITEM1 ~~  ITEM2 82.448  0.613   0.613    0.549    0.549
## 59    EMO =~ ITEM12 41.517 -0.313  -0.400   -0.335   -0.335
```

We expect the reduction in χ^2 for `model_2` to be roughly 91.282, based on the inclusion of parameter `ITEM6 ~~ ITEM16`. We see that this is indeed the case if we subtract the χ^2 values for `model_1_fit` and `model_2_fit`:

$$\chi^2_{\text{model 1}} - \chi^2_{\text{model 2}} = 695.719 - 597.731 = 97.988$$

Now, if we perform a LRT we expect to see the same difference since this time we used the default `ML` estimator.

```
# Likelihood Ratio Test.
lavTestLRT(model_1_fit, model_2_fit)
```

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

```
##             Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## model_2_fit 205 25620 25808 597.73
## model_1_fit 206 25716 25900 695.72     97.988       1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Again, we see a χ^2 difference of 97.988.

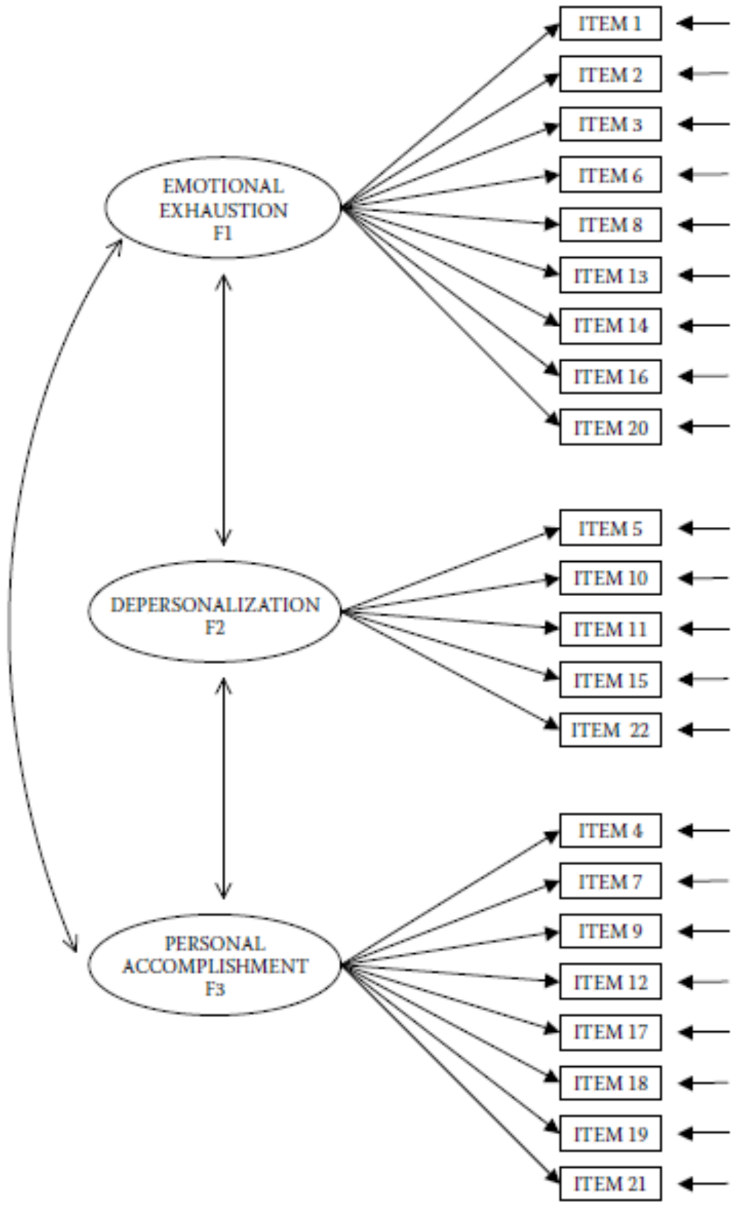
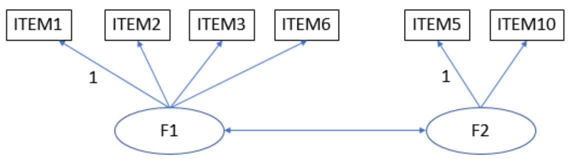


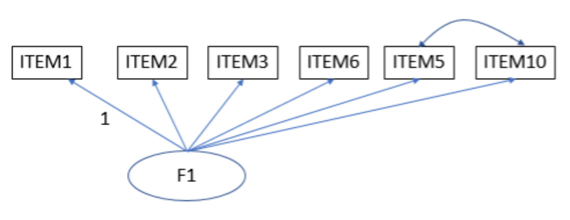
Figure 1: Hypothesized CFA model of factorial structure for the *Maslach Burnout Inventory* (MBI).

Exercise 2

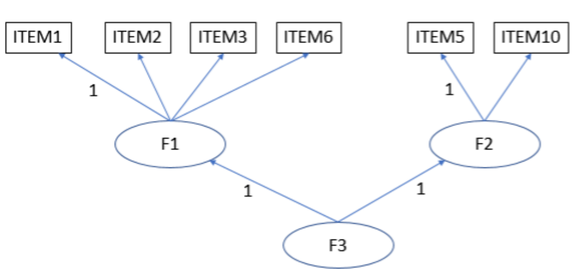
a. Estimate and visualize each of the following four models in Figure 2 using the dataset from the previous exercise.



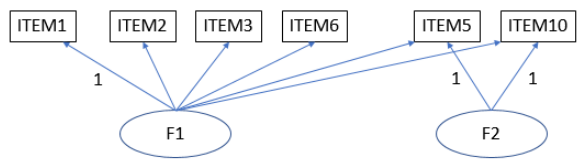
(a) First model.



(b) Second model.



(c) Third model.



(d) Fourth model.

Figure 2: Four models that have something in common.

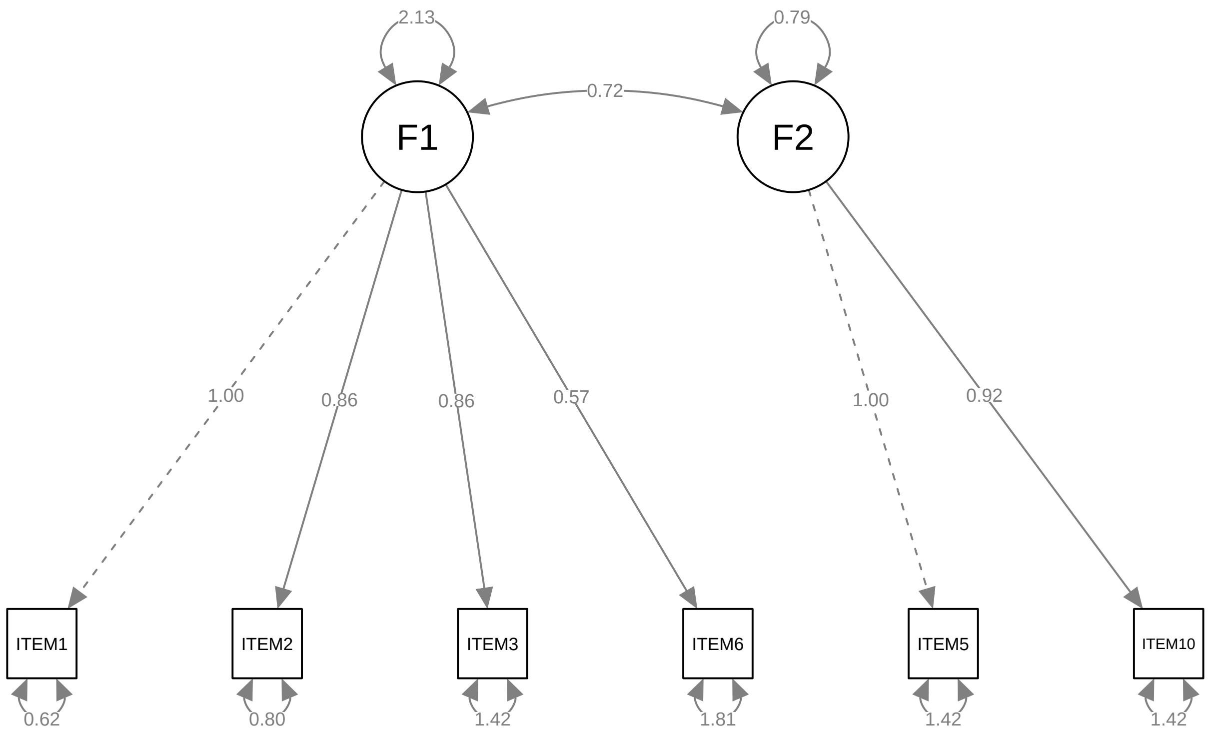
Model 2a

```
# Model syntax.
model_ex_2_a <- "
    # Measurement part.
    F1 =~ ITEM1 + ITEM2 + ITEM3 + ITEM6
    F2 =~ ITEM5 + ITEM10

    # Covariance between latent variables.
    F1 ~~ F2
"

# Fit the model.
model_ex_2_a_fit <- cfa(model_ex_2_a, data = data, estimator = "ML")

# Visualize the model.
semPaths(model_ex_2_a_fit, what = "paths", whatLabels = "est", sizeMan = 5)
```




```r
# Model summary.
summary(model_ex_2_a_fit)
```

```
## lavaan 0.6-12 ended normally after 35 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                           372
##
## Model Test User Model:
##
##   Test statistic                                50.645
##   Degrees of freedom                                 8
##   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|)
##   F1 =~
##     ITEM1             1.000
##     ITEM2             0.861    0.050   17.150    0.000
##     ITEM3             0.860    0.057   15.139    0.000
##     ITEM6             0.569    0.056   10.219    0.000
```

```
##   F2 =~
##     ITEM5             1.000
##     ITEM10            0.917    0.182    5.028    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 ~~
##     F2                0.718    0.127    5.641    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .ITEM1             0.620    0.096    6.446    0.000
##    .ITEM2             0.802    0.087    9.173    0.000
##    .ITEM3             1.417    0.125   11.315    0.000
##    .ITEM6             1.813    0.141   12.888    0.000
##    .ITEM5             1.417    0.188    7.531    0.000
##    .ITEM10            1.422    0.168    8.459    0.000
##     F1                2.134    0.214    9.961    0.000
##     F2                0.791    0.200    3.955    0.000
```

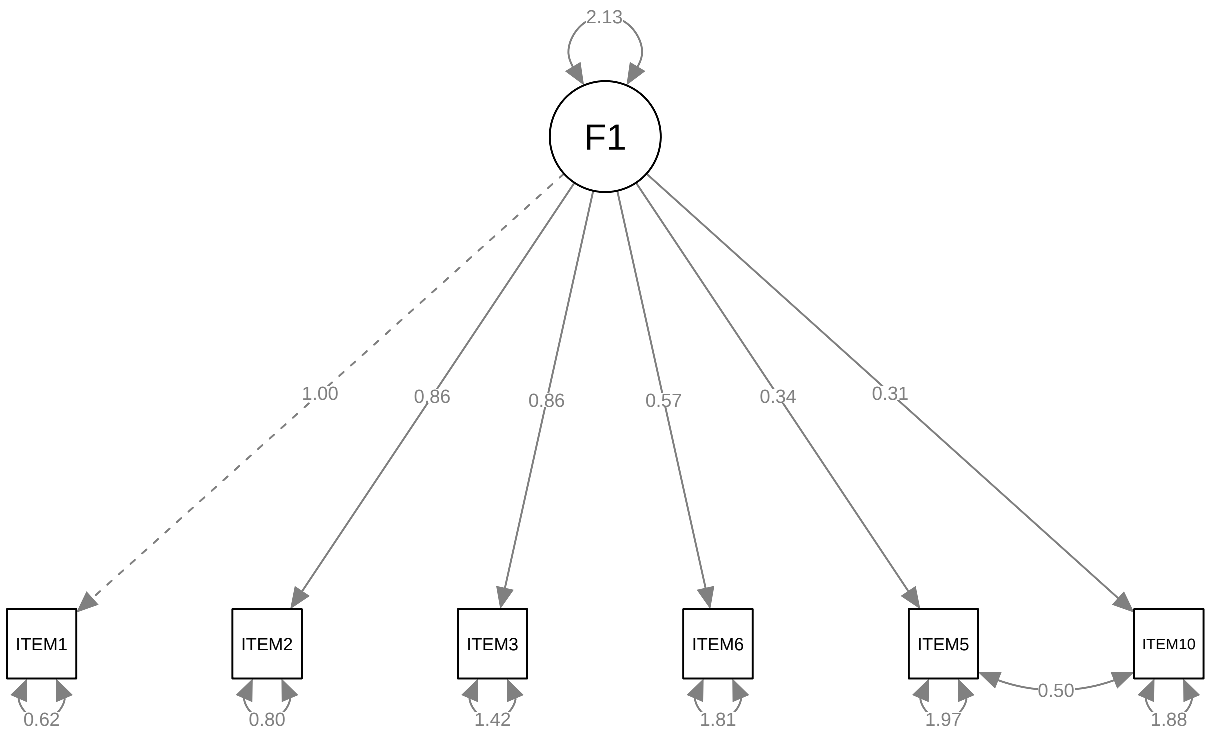
Model 2b

```
# Model syntax.
model_ex_2_b <- "
    # Measurement part.
    F1 =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM5 + ITEM10

    # Covariance between error terms.
    ITEM5 ~~ ITEM10
"

# Fit the model.
model_ex_2_b_fit <- cfa(model_ex_2_b, data = data, estimator = "ML")

# Visualize the model.
semPaths(model_ex_2_b_fit, what = "paths", whatLabels = "est", sizeMan = 5)
```




```
# Model summary.
summary(model_ex_2_b_fit)
```

```
## lavaan 0.6-12 ended normally after 28 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                           372
##
## Model Test User Model:
##
##   Test statistic                                50.645
##   Degrees of freedom                                 8
##   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|)
##   F1 =~
##     ITEM1             1.000
##     ITEM2             0.861    0.050   17.150    0.000
##     ITEM3             0.860    0.057   15.139    0.000
##     ITEM6             0.569    0.056   10.219    0.000
```

```
##     ITEM5             0.337    0.055    6.147    0.000
##     ITEM10            0.309    0.053    5.778    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .ITEM5 ~~
##    .ITEM10            0.504    0.105    4.788    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .ITEM1             0.620    0.096    6.446    0.000
##    .ITEM2             0.802    0.087    9.173    0.000
##    .ITEM3             1.417    0.125   11.315    0.000
##    .ITEM6             1.813    0.141   12.888    0.000
##    .ITEM5             1.966    0.147   13.398    0.000
##    .ITEM10            1.884    0.140   13.427    0.000
##     F1                2.134    0.214    9.961    0.000
```

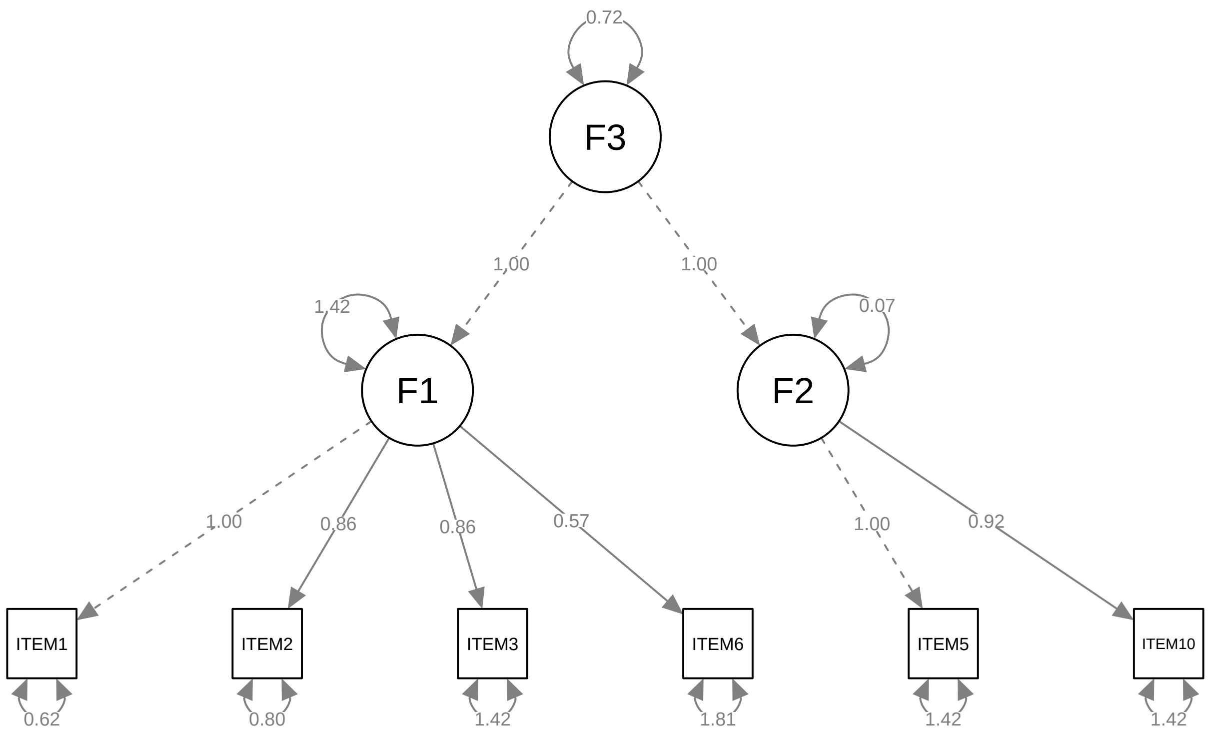
Model 2c

```
# Model syntax.
model_ex_2_c <- "
    # Measurement part.
    F1 =~ ITEM1 + ITEM2 + ITEM3 + ITEM6
    F2 =~ ITEM5 + ITEM10

    # Apply constraints.
    F3 =~ 1 * F1 + 1 * F2
"

# Fit the model.
model_ex_2_c_fit <- cfa(model_ex_2_c, data = data, estimator = "ML")

# Visualize the model.
semPaths(model_ex_2_c_fit, what = "paths", whatLabels = "est", sizeMan = 5)
```




```
# Model summary.
summary(model_ex_2_c_fit)
```

```
## lavaan 0.6-12 ended normally after 36 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                           372
##
## Model Test User Model:
##
##   Test statistic                                50.645
##   Degrees of freedom                                 8
##   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|)
##   F1 =~
##     ITEM1             1.000
##     ITEM2             0.861    0.050   17.150    0.000
##     ITEM3             0.860    0.057   15.139    0.000
##     ITEM6             0.569    0.056   10.219    0.000
```

```
##   F2 =~
##     ITEM5             1.000
##     ITEM10            0.917    0.182    5.028    0.000
##   F3 =~
##     F1                1.000
##     F2                1.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .ITEM1             0.620    0.096    6.446    0.000
##    .ITEM2             0.802    0.087    9.173    0.000
##    .ITEM3             1.417    0.125   11.315    0.000
##    .ITEM6             1.813    0.141   12.888    0.000
##    .ITEM5             1.417    0.188    7.531    0.000
##    .ITEM10            1.422    0.168    8.459    0.000
##    .F1                1.415    0.198    7.163    0.000
##    .F2                0.073    0.155    0.470    0.639
##     F3                0.718    0.127    5.641    0.000
```

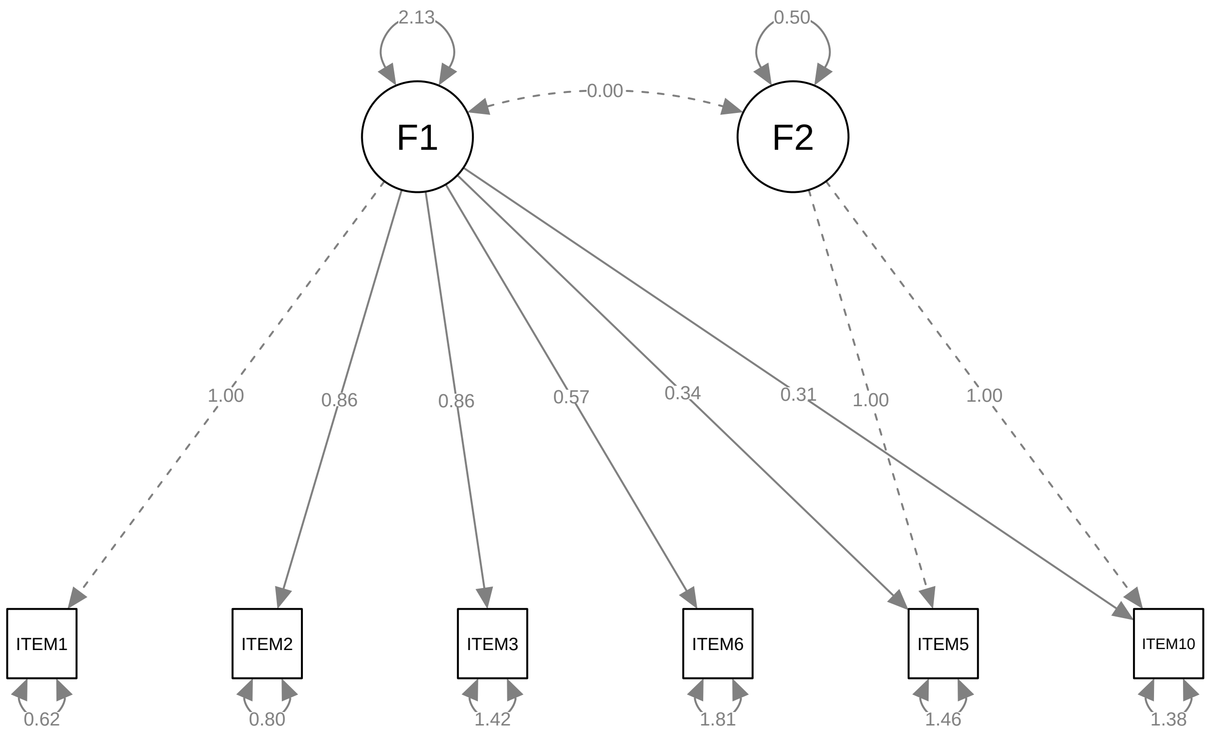
Model 2d

```
# Model syntax.
model_ex_2_d <- "
    # Measurement part.
    F1 =~ ITEM1 + ITEM2 + ITEM3 + ITEM6 + ITEM5 + ITEM10
    F2 =~ 1 * ITEM5 + 1 * ITEM10

    # Apply constraints.
    F1 ~~ 0 * F2
"

# Fit the model.
model_ex_2_d_fit <- cfa(model_ex_2_d, data = data, estimator = "ML")

# Visualize the model.
semPaths(model_ex_2_d_fit, what = "paths", whatLabels = "est", sizeMan = 5)
```




```r
# Model summary.
summary(model_ex_2_d_fit)
```

```
## lavaan 0.6-12 ended normally after 31 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        13
##
##   Number of observations                           372
##
## Model Test User Model:
##
##   Test statistic                                50.645
##   Degrees of freedom                                 8
##   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|)
##   F1 =~
##     ITEM1             1.000
##     ITEM2             0.861    0.050   17.150    0.000
##     ITEM3             0.860    0.057   15.139    0.000
##     ITEM6             0.569    0.056   10.219    0.000
```

```
##     ITEM5             0.337    0.055    6.147    0.000
##     ITEM10            0.309    0.053    5.778    0.000
##   F2 =~
##     ITEM5             1.000
##     ITEM10            1.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 ~~
##     F2                0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .ITEM1             0.620    0.096    6.446    0.000
##    .ITEM2             0.802    0.087    9.173    0.000
##    .ITEM3             1.417    0.125   11.315    0.000
##    .ITEM6             1.813    0.141   12.888    0.000
##    .ITEM5             1.462    0.145   10.076    0.000
##    .ITEM10            1.380    0.140    9.828    0.000
##     F1                2.134    0.214    9.961    0.000
##     F2                0.504    0.105    4.788    0.000
```

b. What do they have in common?

It turns out that all these models are… ***equivalent*** models in terms of model fit (i.e., check the χ^2 values and also the values for the parameter estimates).