Exercises multigroup LVMs

```r
library("lavaan")
```

```
## Warning: package 'lavaan' was built under R version 3.4.4
```

```
## This is lavaan 0.6-1
```

```
## lavaan is BETA software! Please report any bugs.
```

Exercise 4.1: Measurement invariance between the WISC and WISC-IV

We read in the data:

```r
WISC.names <- c("Compr", "Arith", "Simil", "Vocab", "DigSpan", "PictCompl", "BlockDes", "Cod")

WISC.cor <- lav_matrix_lower2full(c(
  1.00,
  0.31, 1.00,
  0.36, 0.40, 1.00,
  0.51, 0.46, 0.45, 1.00,
  0.29, 0.40, 0.33, 0.43, 1.00,
  0.39, 0.29, 0.27, 0.36, 0.33, 1.00,
  0.32, 0.27, 0.29, 0.33, 0.24, 0.28, 1.00,
  0.22, 0.32, 0.15, 0.22, 0.27, 0.12, 0.26, 1.00
))
WISC.means <- c(7.83, 5.50, 5.67, 21.50, 7.67, 8.00, 6.50, 34.83)
WISC.sds <- c(2.69, 1.50, 2.36, 6.06, 1.85, 2.18, 5.97, 9.94)
WISC.cov <- cor2cov(WISC.cor, sds=WISC.sds)

WISCIV.cor <- lav_matrix_lower2full(c(
  1.00,
  0.46, 1.00,
  0.58, 0.55, 1.00,
  0.63, 0.43, 0.73, 1.00,
  0.27, 0.51, 0.37, 0.33, 1.00,
  0.45, 0.38, 0.37, 0.43, 0.13, 1.00,
  0.33, 0.52, 0.49, 0.41, 0.29, 0.43, 1.00,
  0.15, 0.27, 0.16, 0.09, 0.12, 0.25, 0.23, 1.00
))
WISCIV.means <- c(15.17, 15.00, 11.83, 21.67, 12.17, 17.83, 18.67, 45.83)
WISCIV.sds <- c(4.93, 4.10, 5.20, 6.54, 2.72, 5.35, 9.36, 10.44)
WISCIV.cov <- cor2cov(WISCIV.cor, sds=WISCIV.sds)

names(WISC.means) <- names(WISCIV.sds) <- names(WISC.sds) <-
  names(WISCIV.sds) <- rownames(WISC.cov) <- colnames(WISC.cov) <-
  rownames(WISCIV.cov) <- colnames(WISCIV.cov) <- WISC.names

WISC.cov.list <- list(WISC.cov, WISCIV.cov)
WISC.mean.list <- list(WISC.means, WISCIV.means)
WISC.n.list <- list(WISC.n = 200, WISCIV.n = 200)
```

a) Fit a two-domensional model, with Verbal Comprehension (Similarities, Vocabulary and Comprehension), Working Memory (Artihmetic, Digit Span and Coding) and Perceptual Reasoning (Picture Completion and Block Design) to both covariance matrices. Perform a multigroup analyses, using edition as the grouping variable.

We fit the configural invariance model to the datasets:

```r
WISC.mod <- '
  ## verbal comprehension
  VC =~ Simil + Vocab + Compr

  ## Perceptual reasoning
  PR =~ PictCompl + BlockDes

  ## Working memory
  WM =~ Arith + DigSpan + Cod
'
fit.indices <- c("chisq", "df", "pvalue", "cfi", "rmsea", "srmr", "aic")
WISC.conf.fit <- cfa(WISC.mod, sample.cov = WISC.cov.list,
                     sample.mean = WISC.mean.list,
                     sample.nobs = WISC.n.list, meanstructure = TRUE)
summary(WISC.conf.fit, standardized = TRUE)
```

```
## lavaan (0.6-1) converged normally after 180 iterations
##
##   Number of observations per group
##   Group 1                                          200
##   Group 2                                          200
##
##   Estimator                                         ML
##   Model Fit Test Statistic                      62.187
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.002
##
## Chi-square for each group:
##
##   Group 1                                       18.998
##   Group 2                                       43.189
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
##
## Group 1 [Group 1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   VC =~
##     Simil             1.000                               1.393    0.592
##     Vocab             3.346    0.457    7.328    0.000    4.659    0.771
##     Compr             1.238    0.185    6.693    0.000    1.725    0.643
##   PR =~
##     PictCompl         1.000                               1.198    0.551
##     BlockDes          2.526    0.498    5.077    0.000    3.026    0.508
```

```
##   WM =~
##     Arith             1.000                               1.018    0.681
##     DigSpan           1.126    0.171    6.568    0.000    1.147    0.621
##     Cod               4.049    0.847    4.780    0.000    4.123    0.416
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   VC ~~
##     PR                1.513    0.310    4.876    0.000    0.907    0.907
##     WM                1.187    0.224    5.287    0.000    0.837    0.837
##   PR ~~
##     WM                1.000    0.211    4.751    0.000    0.820    0.820
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Simil             5.670    0.166   34.062    0.000    5.670    2.409
##    .Vocab            21.500    0.427   50.300    0.000   21.500    3.557
##    .Compr             7.830    0.190   41.268    0.000    7.830    2.918
##    .PictCompl         8.000    0.154   52.028    0.000    8.000    3.679
##    .BlockDes          6.500    0.421   15.436    0.000    6.500    1.092
##    .Arith             5.500    0.106   51.985    0.000    5.500    3.676
##    .DigSpan           7.670    0.130   58.780    0.000    7.670    4.156
##    .Cod              34.830    0.701   49.679    0.000   34.830    3.513
##     VC                0.000                               0.000    0.000
##     PR                0.000                               0.000    0.000
##     WM                0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Simil             3.602    0.419    8.603    0.000    3.602    0.650
##    .Vocab            14.834    2.454    6.044    0.000   14.834    0.406
##    .Compr             4.226    0.518    8.151    0.000    4.226    0.587
##    .PictCompl         3.293    0.451    7.308    0.000    3.293    0.696
##    .BlockDes         26.304    3.282    8.015    0.000   26.304    0.742
##    .Arith             1.202    0.179    6.722    0.000    1.202    0.537
##    .DigSpan           2.090    0.272    7.691    0.000    2.090    0.614
##    .Cod              81.310    8.768    9.273    0.000   81.310    0.827
##     VC                1.939    0.472    4.108    0.000    1.000    1.000
##     PR                1.436    0.458    3.134    0.002    1.000    1.000
##     WM                1.037    0.231    4.496    0.000    1.000    1.000
##
##
## Group 2 [Group 2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   VC =~
##     Simil             1.000                               4.460    0.860
##     Vocab             1.233    0.094   13.137    0.000    5.499    0.843
##     Compr             0.788    0.072   10.915    0.000    3.516    0.715
##   PR =~
##     PictCompl         1.000                               3.194    0.599
##     BlockDes          2.100    0.309    6.794    0.000    6.707    0.718
##   WM =~
```

```
##     Arith             1.000                               3.761    0.920
##     DigSpan           0.397    0.062    6.447    0.000    1.495    0.551
##     Cod               0.820    0.216    3.790    0.000    3.085    0.296
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   VC ~~
##     PR               10.806    1.934    5.586    0.000    0.758    0.758
##     WM               10.853    1.637    6.630    0.000    0.647    0.647
##   PR ~~
##     WM                9.022    1.594    5.661    0.000    0.751    0.751
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Simil            11.830    0.367   32.254    0.000   11.830    2.281
##    .Vocab            21.670    0.461   46.977    0.000   21.670    3.322
##    .Compr            15.170    0.348   43.626    0.000   15.170    3.085
##    .PictCompl        17.830    0.377   47.250    0.000   17.830    3.341
##    .BlockDes         18.670    0.660   28.280    0.000   18.670    2.000
##    .Arith            15.000    0.289   51.869    0.000   15.000    3.668
##    .DigSpan          12.170    0.192   63.434    0.000   12.170    4.485
##    .Cod              45.830    0.736   62.238    0.000   45.830    4.401
##     VC                0.000                               0.000    0.000
##     PR                0.000                               0.000    0.000
##     WM                0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Simil             7.014    1.237    5.670    0.000    7.014    0.261
##    .Vocab            12.323    1.991    6.190    0.000   12.323    0.290
##    .Compr            11.823    1.394    8.481    0.000   11.823    0.489
##    .PictCompl        18.276    2.247    8.133    0.000   18.276    0.642
##    .BlockDes         42.184    7.143    5.906    0.000   42.184    0.484
##    .Arith             2.580    1.642    1.571    0.116    2.580    0.154
##    .DigSpan           5.127    0.578    8.869    0.000    5.127    0.696
##    .Cod              98.930   10.059    9.835    0.000   98.930    0.912
##     VC               19.891    2.790    7.129    0.000    1.000    1.000
##     PR               10.204    2.546    4.008    0.000    1.000    1.000
##     WM               14.146    2.316    6.109    0.000    1.000    1.000
```

```
fitMeasures(WISC.conf.fit, fit.indices)
```

```
##     chisq        df    pvalue       cfi     rmsea      srmr       aic
##    62.187    34.000     0.002     0.968     0.064     0.037 17617.734
```

Note that the tests were made by the sampe individuals, so assuming that the latent means are similar between the two groups (e.g., identifying the scale by setting the latent means to 0) seems like a reasonable assumption.

b) Assess whether configural invariance between the WISC and WISC-IV is tenable.

The model fits well according to the CFI and SRMR, but not according to the χ^2 and RMSEA, though the latter does indicate acceptable fit. The model misfit seems stronger for the WISC-IV (higher χ^2 for group 2). All standardized loadings are substantial in both groups, although the loading for Coding is relatively low in the WISC-IV group, but it is still substantial and significant. The three factors correlate substantially in both groups, but more strongly in the WISC than in the WISC-IV group.

As this is just an exercise, we do not look further for potential sources of misfit, but conclude the configural invariance model fits the data. We continue our analysis by restricting loadings to be equal across the four groups:

c) Assess whether loadings, intercepts and residual variances are equal between the two WISC versions.

```
WISC.metr.fit <- cfa(WISC.mod, sample.cov = WISC.cov.list,
                     sample.mean = WISC.mean.list,
                     sample.nobs = WISC.n.list,
                     meanstructure = TRUE, group.equal = "loadings")
fitMeasures(WISC.metr.fit, fit.indices)
```

```
##     chisq        df    pvalue       cfi     rmsea      srmr       aic
##   132.004    39.000     0.000     0.896     0.109     0.093 17677.551
```

```
lavTestLRT(WISC.conf.fit, WISC.metr.fit)
```

```
## Chi Square Difference Test
##
##               Df   AIC   BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## WISC.conf.fit 34 17618 17833  62.187
## WISC.metr.fit 39 17678 17873 132.004     69.817       5  1.119e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We did not obtain full metric invariance, according to $\Delta\chi^2$ and ΔCFI. BIC, AIC and RMSEA also indicate loadings are not equal between WISC and WISC-IV.

```
lavTestScore(WISC.metr.fit)
```

```
## $test
##
## total score test:
##
##    test     X2 df p.value
## 1 score 65.047  5       0
##
## $uni
##
## univariate score tests:
##
##    lhs op   rhs     X2 df p.value
## 1 .p2. == .p35. 27.669  1   0.000
## 2 .p3. == .p36.  0.052  1   0.819
## 3 .p5. == .p38.  0.680  1   0.410
## 4 .p7. == .p40. 12.023  1   0.001
## 5 .p8. == .p41. 11.753  1   0.001
```

We see that three out of five restrictions are not tenable (i.e., yield a significant test statistic). Which parameters are those?

```
pars <- parameterestimates(WISC.metr.fit)
pars[pars$label %in% c(".p2.", ".p7.", ".p8."),]
```

```
##    lhs op     rhs block group label   est    se      z pvalue ci.lower
## 2   VC =~   Vocab     1     1  .p2. 1.498 0.107 13.969      0    1.288
## 7   WM =~ DigSpan     1     1  .p7. 0.529 0.063  8.351      0    0.405
## 8   WM =~     Cod     1     1  .p8. 1.248 0.239  5.229      0    0.780
## 35  VC =~   Vocab     2     2  .p2. 1.498 0.107 13.969      0    1.288
```

```
## 40  WM =~ DigSpan     2     2  .p7. 0.529 0.063  8.351      0    0.405
## 41  WM =~     Cod     2     2  .p8. 1.248 0.239  5.229      0    0.780
##    ci.upper
## 2     1.708
## 7     0.653
## 8     1.716
## 35    1.708
## 40    0.653
## 41    1.716
```

Vocabulary, Digit Span and Coding do not seem to have equal loadings in both subtests. We will lift those equality restricitons:

```
WISC.metr.fit2 <- cfa(WISC.mod, sample.cov = WISC.cov.list,
                     sample.mean = WISC.mean.list,
                     sample.nobs = WISC.n.list,
                     meanstructure = TRUE, group.equal = "loadings",
                     group.partial = c("VC =~ Vocab", "WM =~ DigSpan", "WM =~ Cod"))
fitMeasures(WISC.metr.fit2, fit.indices)
```

```
##     chisq        df    pvalue       cfi     rmsea      srmr       aic
##    69.187    36.000     0.001     0.963     0.068     0.047 17620.734
```

```
lavTestLRT(WISC.conf.fit, WISC.metr.fit2)
```

```
## Chi Square Difference Test
##
##                Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## WISC.conf.fit  34 17618 17833 62.187
## WISC.metr.fit2 36 17621 17828 69.187     7.0002       2    0.03019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Even after lifting three out of five restictions, the difference in model fit is significance, so equality of loadings does not seem tenable at all. Let's check whether the intercepts are equal:

```
WISC.scal.fit <- cfa(WISC.mod, sample.cov = WISC.cov.list,
                     sample.mean = WISC.mean.list,
                     sample.nobs = WISC.n.list, meanstructure = TRUE,
                     group.equal = "intercepts")
fitMeasures(WISC.scal.fit, fit.indices)
```

```
##     chisq        df    pvalue       cfi     rmsea      srmr       aic
##   211.365    39.000     0.000     0.807     0.149     0.109 17756.913
```

```
lavTestLRT(WISC.conf.fit, WISC.scal.fit)
```

```
## Chi Square Difference Test
##
##               Df   AIC   BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## WISC.conf.fit 34 17618 17833  62.187
## WISC.scal.fit 39 17757 17953 211.365     149.18       5  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lavTestScore(WISC.scal.fit)
```

```
## $test
##
```

```
## total score test:
##
##    test      X2 df p.value
## 1 score 117.749  8       0
##
## $uni
##
## univariate score tests:
##
##     lhs op   rhs     X2 df p.value
## 1 .p23. == .p56.  2.017  1   0.156
## 2 .p24. == .p57. 92.989  1   0.000
## 3 .p25. == .p58. 32.190  1   0.000
## 4 .p26. == .p59.  8.430  1   0.004
## 5 .p27. == .p60.  8.430  1   0.004
## 6 .p28. == .p61.  3.358  1   0.067
## 7 .p29. == .p62.  1.242  1   0.265
## 8 .p30. == .p63.  1.607  1   0.205
```

```
pars <- parameterestimates(WISC.scal.fit)
pars[pars$label %in% c(".p24.", ".p25.", ".p26.", ".p27."),]
```

```
##          lhs op rhs block group label    est    se      z pvalue ci.lower
## 24     Vocab ~1         1     1 .p24. 19.608 0.400 48.964      0   18.823
## 25     Compr ~1         1     1 .p25.  7.617 0.185 41.143      0    7.254
## 26 PictCompl ~1         1     1 .p26.  7.792 0.152 51.209      0    7.494
## 27  BlockDes ~1         1     1 .p27.  5.713 0.414 13.809      0    4.902
## 57     Vocab ~1         2     2 .p24. 19.608 0.400 48.964      0   18.823
## 58     Compr ~1         2     2 .p25.  7.617 0.185 41.143      0    7.254
## 59 PictCompl ~1         2     2 .p26.  7.792 0.152 51.209      0    7.494
## 60  BlockDes ~1         2     2 .p27.  5.713 0.414 13.809      0    4.902
##    ci.upper
## 24   20.392
## 25    7.980
## 26    8.090
## 27    6.524
## 57   20.392
## 58    7.980
## 59    8.090
## 60    6.524
```

Vocabulary, Comprehension, Picture Completion and subtests have different intercepts.

```
WISC.scal.fit2 <- cfa(WISC.mod, sample.cov = WISC.cov.list,
                      sample.mean = WISC.mean.list,
                      sample.nobs = WISC.n.list, meanstructure = TRUE,
                      group.equal = "intercepts",
                      group.partial = c("Compr ~ 1","Vocab ~ 1", "PictCompl ~ 1", "BlockDes ~ 1"))
fitMeasures(WISC.scal.fit2, fit.indices)
```

```
##     chisq        df    pvalue       cfi     rmsea      srmr       aic
##    64.711    35.000     0.002     0.967     0.065     0.040 17618.258
```

```
lavTestLRT(WISC.scal.fit2, WISC.conf.fit)
```

```
## Chi Square Difference Test
##
```

```
##                Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## WISC.conf.fit  34 17618 17833 62.187
## WISC.scal.fit2 35 17618 17830 64.711     2.5238       1     0.1121
```

Lifting these three equality restrictions still yields acceptable model fit. We continue with testing equality of measurement error variances:

```
WISC.uni.fit <- cfa(WISC.mod, sample.cov = WISC.cov.list,
                    sample.mean = WISC.mean.list,
                    sample.nobs = WISC.n.list, meanstructure = TRUE,
                    group.equal = c("intercepts", "residuals"),
                    group.partial = c("Compr ~ 1","Vocab ~ 1", "PictCompl ~ 1", "BlockDes ~ 1"))
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated lv
## variances are negative
```

```
fitMeasures(WISC.uni.fit, fit.indices)
```

```
##     chisq        df    pvalue       cfi     rmsea      srmr       aic
##   197.745    43.000     0.000     0.827     0.134     0.106 17735.292
```

```
lavTestLRT(WISC.uni.fit, WISC.scal.fit2)
```

```
## Chi Square Difference Test
##
##                Df   AIC   BIC   Chisq Chisq diff Df diff Pr(>Chisq)
## WISC.scal.fit2 35 17618 17830  64.711
## WISC.uni.fit   43 17735 17915 197.745     133.03       8  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lavTestScore(WISC.uni.fit)
```

```
## $test
##
## total score test:
##
##    test     X2 df p.value
## 1 score 96.059 12       0
##
## $uni
##
## univariate score tests:
##
##      lhs op   rhs     X2 df p.value
## 1    .p9. == .p42. 17.767  1   0.000
## 2   .p10. == .p43.  0.043  1   0.836
## 3   .p11. == .p44. 40.795  1   0.000
## 4   .p12. == .p45.  0.016  1   0.899
## 5   .p13. == .p46.  0.568  1   0.451
## 6   .p14. == .p47. 15.323  1   0.000
## 7   .p15. == .p48. 30.594  1   0.000
## 8   .p16. == .p49.  1.367  1   0.242
## 9   .p23. == .p56.  0.000  1   1.000
## 10  .p28. == .p61.  7.162  1   0.007
## 11  .p29. == .p62.  4.135  1   0.042
## 12  .p30. == .p63.  3.297  1   0.069
```

We get a warning about negative LV variances, which indicates a problem with the model. Obviously, restricting all measurement error variances to be equal is not a good idea. Let us stick with the configural invariant model, and look at the differences in parameter estimates to get an idea of the differences between the two versions of the WISC. Five out of eight equality restrictions on reisdual variances have modification indices > 5.

```
pars <- parameterestimates(WISC.conf.fit, standardized = TRUE)
col_names <- c("lhs", "op", "rhs", "group", "est", "se", "pvalue", "std.all")
pars[pars$op == "~~", colnames(pars) %in% col_names]
```

```
##          lhs op       rhs group    est     se pvalue std.all
## 9      Simil ~~     Simil     1  3.602  0.419  0.000   0.650
## 10     Vocab ~~     Vocab     1 14.834  2.454  0.000   0.406
## 11     Compr ~~     Compr     1  4.226  0.518  0.000   0.587
## 12 PictCompl ~~ PictCompl     1  3.293  0.451  0.000   0.696
## 13  BlockDes ~~  BlockDes     1 26.304  3.282  0.000   0.742
## 14     Arith ~~     Arith     1  1.202  0.179  0.000   0.537
## 15   DigSpan ~~   DigSpan     1  2.090  0.272  0.000   0.614
## 16       Cod ~~       Cod     1 81.310  8.768  0.000   0.827
## 17        VC ~~        VC     1  1.939  0.472  0.000   1.000
## 18        PR ~~        PR     1  1.436  0.458  0.002   1.000
## 19        WM ~~        WM     1  1.037  0.231  0.000   1.000
## 20        VC ~~        PR     1  1.513  0.310  0.000   0.907
## 21        VC ~~        WM     1  1.187  0.224  0.000   0.837
## 22        PR ~~        WM     1  1.000  0.211  0.000   0.820
## 42     Simil ~~     Simil     2  7.014  1.237  0.000   0.261
## 43     Vocab ~~     Vocab     2 12.323  1.991  0.000   0.290
## 44     Compr ~~     Compr     2 11.823  1.394  0.000   0.489
## 45 PictCompl ~~ PictCompl     2 18.276  2.247  0.000   0.642
## 46  BlockDes ~~  BlockDes     2 42.184  7.143  0.000   0.484
## 47     Arith ~~     Arith     2  2.580  1.642  0.116   0.154
## 48   DigSpan ~~   DigSpan     2  5.127  0.578  0.000   0.696
## 49       Cod ~~       Cod     2 98.930 10.059  0.000   0.912
## 50        VC ~~        VC     2 19.891  2.790  0.000   1.000
## 51        PR ~~        PR     2 10.204  2.546  0.000   1.000
## 52        WM ~~        WM     2 14.146  2.316  0.000   1.000
## 53        VC ~~        PR     2 10.806  1.934  0.000   0.758
## 54        VC ~~        WM     2 10.853  1.637  0.000   0.647
## 55        PR ~~        WM     2  9.022  1.594  0.000   0.751
```

We see that the residual variances for Similarities, Vocabulary, Picture Completion, Block Design and Arithmetic are lower in the second group (WISC-IV) thin in the first group (WISC). The residual variances for Digit Span and Coding are larger for the WISC-IV than for the WISC.

We also see stronger correlations between the latent factors in the WISC, than in the WISC-IV.

```
pars[pars$op == "=~", colnames(pars) %in% col_names]
```

```
##    lhs op       rhs group   est    se pvalue std.all
## 1   VC =~     Simil     1 1.000 0.000     NA   0.592
## 2   VC =~     Vocab     1 3.346 0.457      0   0.771
## 3   VC =~     Compr     1 1.238 0.185      0   0.643
## 4   PR =~ PictCompl     1 1.000 0.000     NA   0.551
## 5   PR =~  BlockDes     1 2.526 0.498      0   0.508
## 6   WM =~     Arith     1 1.000 0.000     NA   0.681
## 7   WM =~   DigSpan     1 1.126 0.171      0   0.621
```

```
## 8   WM =~      Cod     1 4.049 0.847      0    0.416
## 34  VC =~    Simil     2 1.000 0.000     NA    0.860
## 35  VC =~    Vocab     2 1.233 0.094      0    0.843
## 36  VC =~    Compr     2 0.788 0.072      0    0.715
## 37  PR =~ PictCompl    2 1.000 0.000     NA    0.599
## 38  PR =~  BlockDes    2 2.100 0.309      0    0.718
## 39  WM =~    Arith     2 1.000 0.000     NA    0.920
## 40  WM =~  DigSpan     2 0.397 0.062      0    0.551
## 41  WM =~      Cod     2 0.820 0.216      0    0.296
```

We see a similar (but reversed) pattern for the loadings: Standardized loadings are lower for Digit Span and Coding for the WISC-IV than for the WISC. At the same time, standardized loadings are higher for Similarities, Vocabulary, Picture Completion, Block Design and Arithmetic for the WISC-IV than for the WISC.

```
pars[pars$op == "~1", colnames(pars) %in% col_names]
```

```
##          lhs op rhs group   est    se pvalue std.all
## 23     Simil ~1          1  5.67 0.166      0   2.409
## 24     Vocab ~1          1 21.50 0.427      0   3.557
## 25     Compr ~1          1  7.83 0.190      0   2.918
## 26 PictCompl ~1          1  8.00 0.154      0   3.679
## 27  BlockDes ~1          1  6.50 0.421      0   1.092
## 28     Arith ~1          1  5.50 0.106      0   3.676
## 29   DigSpan ~1          1  7.67 0.130      0   4.156
## 30       Cod ~1          1 34.83 0.701      0   3.513
## 31        VC ~1          1  0.00 0.000     NA   0.000
## 32        PR ~1          1  0.00 0.000     NA   0.000
## 33        WM ~1          1  0.00 0.000     NA   0.000
## 56     Simil ~1          2 11.83 0.367      0   2.281
## 57     Vocab ~1          2 21.67 0.461      0   3.322
## 58     Compr ~1          2 15.17 0.348      0   3.085
## 59 PictCompl ~1          2 17.83 0.377      0   3.341
## 60  BlockDes ~1          2 18.67 0.660      0   2.000
## 61     Arith ~1          2 15.00 0.289      0   3.668
## 62   DigSpan ~1          2 12.17 0.192      0   4.485
## 63       Cod ~1          2 45.83 0.736      0   4.401
## 64        VC ~1          2  0.00 0.000     NA   0.000
## 65        PR ~1          2  0.00 0.000     NA   0.000
## 66        WM ~1          2  0.00 0.000     NA   0.000
```

We see that the subscale intercepts are higher for all subtests of the WISC-IV than of the WISC. Thus, the subtests of the WISC-IV may be easier. Such a difference could for example be accounted for by the scoring rules used to compute IQ scores based on the subtest (but we would require much larger samples to do that).

Exercise 4.2: Genetically informative design:

We read in the data:

```r
MZ <- lav_matrix_lower2full(c(
  .725,
  .589, .792
))
DZ <- lav_matrix_lower2full(c(
  .779,
  .246, .837
))
rownames(MZ) <- colnames(MZ) <- rownames(DZ)  <- colnames(DZ) <- c("P1", "P2")
bmi.cov <- list(MZ=MZ, DZ=DZ)
bmi.n <- list(MZ=534, DZ=328)
```

We fit the ACE model:

```r
bmi.ace.model <- '
  # build the factor model with group constraints:
  A1 =~ NA*P1 + c(a,a)*P1
  A2 =~ NA*P2 + c(a,a)*P2
  C =~ NA*P1 + c(c,c)*P1 + NA*P2 + c(c,c)*P2
  # constrain the factor variances:
  A1 ~~ 1*A1
  A2 ~~ 1*A2
  C ~~ 1*C
  P1 ~~ c(e,e)*P1
  P2 ~~ c(e,e)*P2
  # constrain the factor covariances:
  A1 ~~ c(1, .5)*A2
  A1 ~~ 0*C
  A2 ~~ 0*C
'
bmi.ace.fit <- cfa(bmi.ace.model, sample.cov = bmi.cov, sample.nobs = bmi.n)
summary(bmi.ace.fit, standardized = TRUE)
```

```
## lavaan (0.6-1) converged normally after  15 iterations
##
##   Number of observations per group
##   MZ                                               534
##   DZ                                               328
##
##   Estimator                                         ML
##   Model Fit Test Statistic                       8.040
##   Degrees of freedom                                 3
##   P-value (Chi-square)                           0.045
##
## Chi-square for each group:
##
##   MZ                                             3.382
##   DZ                                             4.658
##
## Parameter Estimates:
##
```

```
##   Information                                   Expected
##   Information saturated (h1) model            Structured
##   Standard Errors                               Standard
##
##
## Group 1 [MZ]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   A1 =~
##     P1         (a)    0.786    0.021   38.075    0.000    0.786    0.884
##   A2 =~
##     P2         (a)    0.786    0.021   38.075    0.000    0.786    0.884
##   C =~
##     P1         (c)   -0.000                              -0.000   -0.000
##     P2         (c)   -0.000                              -0.000   -0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   A1 ~~
##     A2                1.000                               1.000    1.000
##     C                 0.000                               0.000    0.000
##   A2 ~~
##     C                 0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     A1                1.000                               1.000    1.000
##     A2                1.000                               1.000    1.000
##     C                 1.000                               1.000    1.000
##    .P1         (e)    0.174    0.010   16.621    0.000    0.174    0.219
##    .P2         (e)    0.174    0.010   16.621    0.000    0.174    0.219
##
##
## Group 2 [DZ]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   A1 =~
##     P1         (a)    0.786    0.021   38.075    0.000    0.786    0.884
##   A2 =~
##     P2         (a)    0.786    0.021   38.075    0.000    0.786    0.884
##   C =~
##     P1         (c)   -0.000       NA                     -0.000   -0.000
##     P2         (c)   -0.000                              -0.000   -0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   A1 ~~
##     A2                0.500                               0.500    0.500
##     C                 0.000                               0.000    0.000
##   A2 ~~
##     C                 0.000                               0.000    0.000
##
```

```
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     A1                1.000                               1.000    1.000
##     A2                1.000                               1.000    1.000
##     C                 1.000                               1.000    1.000
##    .P1         (e)    0.174    0.010   16.621    0.000    0.174    0.219
##    .P2         (e)    0.174    0.010   16.621    0.000    0.174    0.219
```

```r
fitmeasures(bmi.ace.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##    8.040    3.000    0.045    0.990    0.062    0.058 3939.146
```

We cannot distinguish the effects of shared environment, as all twins pairs were raised in the same environment. All shared variance in BMI within twin pairs thus gets explained by additive genetic effects. Again, residual variance of BMI is about 22 percent.

We fit the CE model:

```r
bmi.ce.model <- '
  # build the factor model with group constraints
  C =~ NA*P1 + c(c,c)*P1
  # constrain the factor variances
  C ~~ 1*C
  P1 ~~ c(e,e)*P1
  P2 ~~ c(e,e)*P2
'
bmi.ce.fit <- cfa(bmi.ce.model, sample.cov = bmi.cov, sample.nobs = bmi.n)
summary(bmi.ce.fit, standardized = TRUE)
```

```
## lavaan (0.6-1) converged normally after  13 iterations
##
##   Number of observations per group
##   MZ                                               534
##   DZ                                               328
##
##   Estimator                                         ML
##   Model Fit Test Statistic                     529.111
##   Degrees of freedom                                 4
##   P-value (Chi-square)                           0.000
##
## Chi-square for each group:
##
##   MZ                                           496.260
##   DZ                                            32.851
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
##
## Group 1 [MZ]:
##
## Latent Variables:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   C =~
##     P1         (c)   -0.000       NA                      -0.000   -0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     C                 1.000                                1.000    1.000
##    .P1         (e)    0.776    0.026   29.360    0.000    0.776    1.000
##     P2         (e)    0.776    0.026   29.360    0.000    0.776    1.000
##
##
## Group 2 [DZ]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   C =~
##     P1         (c)   -0.000                               -0.000   -0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     C                 1.000                                1.000    1.000
##    .P1         (e)    0.776    0.026   29.360    0.000    0.776    1.000
##     P2         (e)    0.776    0.026   29.360    0.000    0.776    1.000
```

```
fitmeasures(bmi.ce.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##  529.111    4.000    0.000    0.000    0.552    0.348 4458.218
```

The CE model does not fit well, which is to be expected: we are trying to explain (co)variation in BMI by an environmental component that we cannot distinguish with our data (c) and residual error (e). According to the estimates of this model, all variation in BMI is due to non-shared environment effects and measurement erro (which is highly unlikely).

We fit the AE model:

```
bmi.ae.model <- '
  # build the factor model with group constraints
  A1=~ NA*P1 + c(a,a)*P1
  A2=~ NA*P2 + c(a,a)*P2
  # constrain the factor variances
  A1 ~~ 1*A1
  A2 ~~ 1*A2
  P1 ~~ c(e,e)*P1
  P2 ~~ c(e,e)*P2
  # constrain the factor covariances
  A1 ~~ c(1,.5)*A2
'
bmi.ae.fit <- cfa(bmi.ae.model, sample.cov = bmi.cov, sample.nobs = bmi.n)
summary(bmi.ae.fit, standardized = TRUE)
```

```
## lavaan (0.6-1) converged normally after  10 iterations
##
##   Number of observations per group
##   MZ                                               534
##   DZ                                               328
```

```
##
##   Estimator                                         ML
##   Model Fit Test Statistic                       8.040
##   Degrees of freedom                                 4
##   P-value (Chi-square)                           0.090
##
## Chi-square for each group:
##
##   MZ                                             3.382
##   DZ                                             4.658
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
##
## Group 1 [MZ]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   A1 =~
##     P1         (a)    0.786    0.021   38.075    0.000    0.786    0.884
##   A2 =~
##     P2         (a)    0.786    0.021   38.075    0.000    0.786    0.884
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   A1 ~~
##     A2                1.000                               1.000    1.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     A1                1.000                               1.000    1.000
##     A2                1.000                               1.000    1.000
##    .P1         (e)    0.174    0.010   16.621    0.000    0.174    0.219
##    .P2         (e)    0.174    0.010   16.621    0.000    0.174    0.219
##
##
## Group 2 [DZ]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   A1 =~
##     P1         (a)    0.786    0.021   38.075    0.000    0.786    0.884
##   A2 =~
##     P2         (a)    0.786    0.021   38.075    0.000    0.786    0.884
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   A1 ~~
##     A2                0.500                               0.500    0.500
##
```

```
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     A1                1.000                               1.000    1.000
##     A2                1.000                               1.000    1.000
##    .P1         (e)    0.174    0.010   16.621    0.000    0.174    0.219
##    .P2         (e)    0.174    0.010   16.621    0.000    0.174    0.219
```

```
fitmeasures(bmi.ae.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##    8.040    4.000    0.090    0.992    0.048    0.058 3937.146
```

The AE model fits well. In this model, we again see that $.884^2 = 78\%$ of BMI variance is explained by additive genetic effects, and 22% is explained by non-shared family effects and measurement error.

We can compare the fit of the ADE, ACE, AE and CE models:

```
bmi.ade.mod <- '

  # build the factor model with group constraints:
  A1=~ NA*P1 + c(a,a)*P1
  A2=~ NA*P2 + c(a,a)*P2
  D1 =~ NA*P1 + c(d,d)*P1
  D2 =~ NA*P2 + c(d,d)*P2

  # constrain the factor variances:
  A1 ~~ 1*A1
  A2 ~~ 1*A2
  D1 ~~ 1*D1
  D2 ~~ 1*D2
  P1 ~~ c(e2,e2)*P1
  P2 ~~ c(e2,e2)*P2

  # constrain the factor covariances:
  A1 ~~ c(1,.5)*A2
  A1 ~~ 0*D1 + 0*D2
  A2 ~~ 0*D1 + 0*D2
  D1 ~~ c(1,.25)*D2
'
bmi.ade.fit <- cfa(bmi.ade.mod, sample.cov=bmi.cov, sample.nobs=bmi.n)
fitmeasures(bmi.ade.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##    3.704    3.000    0.295    0.999    0.023    0.045 3934.811
```

We see that the ADE model fits best, and this model also gives us the most information about A (additive genetic effects), D (non-additive effects) and E (non-shared environmental effects and measurement error).

Exercise 4.3

```
ex43g1.cor <- lav_matrix_lower2full(c(
  1.000,
  0.759, 1.000,
  0.762, 0.787, 1.000,
  0.028, 0.010,-0.058, 1.000,
 -0.061,-0.061,-0.141, 0.785, 1.000,
```

```
 -0.022,-0.052,-0.102, 0.816, 0.816, 1.000
))
ex43g1.sds <- c(0.668, 0.685, 0.707, 0.714, 0.663, 0.653)
ex43g1.means <- c(3.135, 2.991, 3.069, 1.701, 1.527, 1.545)
ex43g1.cov <- cor2cov(ex43g1.cor, sds=ex43g1.sds)

ex43g2.cor <- lav_matrix_lower2full(c(
  1.000,
  0.813, 1.000,
  0.850, 0.835, 1.000,
 -0.188,-0.155,-0.215, 1.000,
 -0.289,-0.250,-0.338, 0.784, 1.000,
 -0.293,-0.210,-0.306, 0.800, 0.832, 1.000
))
ex43g2.sds <- c(0.703, 0.718, 0.762, 0.650, 0.602, 0.614)
ex43g2.means <- c(3.073, 2.847, 2.979, 1.717, 1.580, 1.550)
ex43g2.cov <- cor2cov(ex43g2.cor, sds=ex43g2.sds)

ex43.names <- c("Ind1", "Ind2", "Ind3", "Ind4", "Ind5", "Ind6")
names(ex43g1.means) <- names(ex43g2.means) <- ex43.names
names(ex43g1.sds) <- names(ex43g2.sds) <- ex43.names
rownames(ex43g1.cov) <- colnames(ex43g1.cov) <- rownames(ex43g2.cov) <- colnames(ex43g2.cov) <- ex43.nam

ex43.cov.list <- list(ex43g1.cov, ex43g2.cov)
ex43.mean.list <- list(ex43g1.means, ex43g2.means)
ex43.n.list <- list(ex43g1.n=380, ex43g2.n=379)

ex43.mod <- '
  F1 =~ Ind1 + Ind2 + Ind3
  F2 =~ Ind4 + Ind5 + Ind6
  Ind1 ~ 0*1
  Ind4 ~ 0*1
  F1 ~ NA*1
  F2 ~ NA*1
'
ex43.conf.fit <- cfa(ex43.mod, sample.cov = ex43.cov.list,
                     sample.nobs = ex43.n.list, sample.mean = ex43.mean.list,
                     meanstructure = TRUE)
summary(ex43.conf.fit, standardized = TRUE)
```

```
## lavaan (0.6-1) converged normally after  83 iterations
##
##   Number of observations per group
##   Group 1                                          380
##   Group 2                                          379
##
##   Estimator                                         ML
##   Model Fit Test Statistic                      46.249
##   Degrees of freedom                                16
##   P-value (Chi-square)                           0.000
##
## Chi-square for each group:
##
##   Group 1                                       17.356
```

```
##   Group 2                                     28.894
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
##
## Group 1 [Group 1]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 =~
##     Ind1              1.000                               0.571    0.857
##     Ind2              1.059    0.049   21.688    0.000    0.605    0.885
##     Ind3              1.100    0.050   21.817    0.000    0.628    0.890
##   F2 =~
##     Ind4              1.000                               0.631    0.885
##     Ind5              0.930    0.038   24.252    0.000    0.587    0.886
##     Ind6              0.951    0.037   25.643    0.000    0.601    0.921
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 ~~
##     F2               -0.025    0.020   -1.252    0.211   -0.070   -0.070
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Ind1              0.000                               0.000    0.000
##    .Ind4              0.000                               0.000    0.000
##     F1                3.135    0.034   91.606    0.000    5.486    5.486
##     F2                1.701    0.037   46.502    0.000    2.694    2.694
##    .Ind2             -0.329    0.155   -2.124    0.034   -0.329   -0.482
##    .Ind3             -0.378    0.160   -2.363    0.018   -0.378   -0.536
##    .Ind5             -0.054    0.069   -0.789    0.430   -0.054   -0.082
##    .Ind6             -0.073    0.066   -1.105    0.269   -0.073   -0.113
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Ind1              0.118    0.012    9.699    0.000    0.118    0.266
##    .Ind2              0.102    0.012    8.322    0.000    0.102    0.217
##    .Ind3              0.104    0.013    8.030    0.000    0.104    0.208
##    .Ind4              0.110    0.012    9.427    0.000    0.110    0.216
##    .Ind5              0.094    0.010    9.372    0.000    0.094    0.214
##    .Ind6              0.064    0.009    7.157    0.000    0.064    0.152
##     F1                0.327    0.032   10.108    0.000    1.000    1.000
##     F2                0.399    0.037   10.771    0.000    1.000    1.000
##
##
## Group 2 [Group 2]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
```

```
##   F1 =~
##     Ind1              1.000                               0.638    0.909
##     Ind2              1.001    0.038   26.642    0.000    0.639    0.891
##     Ind3              1.117    0.038   29.450    0.000    0.713    0.937
##   F2 =~
##     Ind4              1.000                               0.561    0.865
##     Ind5              0.970    0.040   24.085    0.000    0.544    0.906
##     Ind6              1.007    0.041   24.635    0.000    0.565    0.922
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   F1 ~~
##     F2               -0.115    0.021   -5.487    0.000   -0.321   -0.321
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Ind1              0.000                               0.000    0.000
##    .Ind4              0.000                               0.000    0.000
##     F1                3.073    0.036   85.212    0.000    4.814    4.814
##     F2                1.717    0.033   51.493    0.000    3.059    3.059
##    .Ind2             -0.229    0.118   -1.945    0.052   -0.229   -0.319
##    .Ind3             -0.454    0.119   -3.828    0.000   -0.454   -0.596
##    .Ind5             -0.085    0.072   -1.179    0.238   -0.085   -0.142
##    .Ind6             -0.178    0.073   -2.438    0.015   -0.178   -0.291
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Ind1              0.085    0.009    9.013    0.000    0.085    0.173
##    .Ind2              0.106    0.011   10.056    0.000    0.106    0.206
##    .Ind3              0.071    0.010    6.867    0.000    0.071    0.122
##    .Ind4              0.106    0.010   10.413    0.000    0.106    0.252
##    .Ind5              0.065    0.008    8.311    0.000    0.065    0.180
##    .Ind6              0.057    0.008    7.197    0.000    0.057    0.151
##     F1                0.407    0.036   11.322    0.000    1.000    1.000
##     F2                0.315    0.030   10.379    0.000    1.000    1.000
```

```r
fitMeasures(ex43.conf.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   46.249   16.000    0.000    0.992    0.071    0.029 5872.392
```

The configural invariant model fits well according to CFI and SRMR. RMSEA indicates acceptable model fit. Let's check the standardized residuals (I have to use a bit of hacky code, because the residuals() function gave an error):

```r
ex43g1.cor - cov2cor(fitted(ex43.conf.fit)[[1]]$cov)
```

```
##        Ind1   Ind2   Ind3   Ind4   Ind5   Ind6
## Ind1  0.000
## Ind2  0.001  0.000
## Ind3  0.000  0.000  0.000
## Ind4  0.081  0.065 -0.003  0.000
## Ind5 -0.008 -0.006 -0.086  0.000  0.000
## Ind6  0.033  0.005 -0.045  0.000 -0.001  0.000
```

```r
ex43g2.cor - cov2cor(fitted(ex43.conf.fit)[[2]]$cov)
```

```
##       Ind1   Ind2   Ind3   Ind4   Ind5   Ind6
## Ind1  0.000
## Ind2  0.003  0.000
## Ind3 -0.002  0.000  0.000
## Ind4  0.064  0.092  0.045  0.000
## Ind5 -0.025  0.009 -0.066  0.001  0.000
## Ind6 -0.024  0.054 -0.029  0.003 -0.003  0.000
```

The absolute values of the standardized residuals are all smaller than .1, so we'll keep the model as it is and continue with testing the equality of loadings:

```r
ex43.weak.fit <- cfa(ex43.mod, sample.cov = ex43.cov.list,
                     sample.nobs = ex43.n.list, sample.mean=ex43.mean.list,
                     meanstructure = TRUE, group.equal = "loadings")
fitMeasures(ex43.weak.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   49.036   20.000    0.000    0.992    0.062    0.032 5867.179
```

```r
lavTestLRT(ex43.conf.fit, ex43.weak.fit)
```

```
## Chi Square Difference Test
##
##               Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## ex43.conf.fit 16 5872.4 6048.4 46.249
## ex43.weak.fit 20 5867.2 6024.7 49.036     2.7867       4     0.5941
```

Equality restrictions on loadings are tenable. We continue with testing the equality of intercepts:

```r
ex43.strong.fit <- cfa(ex43.mod, sample.cov = ex43.cov.list,
                       sample.nobs = ex43.n.list, sample.mean = ex43.mean.list,
                       meanstructure = TRUE,
                       group.equal = c("loadings","intercepts"))
fitMeasures(ex43.strong.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   58.808   24.000    0.000    0.990    0.062    0.033 5868.951
```

```r
lavTestLRT(ex43.strong.fit, ex43.weak.fit)
```

```
## Chi Square Difference Test
##
##                 Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## ex43.weak.fit   20 5867.2 6024.7 49.036
## ex43.strong.fit 24 5869.0 6007.9 58.808     9.7721       4    0.04445 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lavTestScore(ex43.strong.fit)
```

```
## $test
##
## total score test:
##
##    test     X2 df p.value
## 1 score 12.497  8    0.13
```

```
##
## $uni
##
## univariate score tests:
##
##     lhs op   rhs    X2 df p.value
## 1  .p2. == .p25. 6.929  1   0.008
## 2  .p3. == .p26. 0.909  1   0.340
## 3  .p5. == .p28. 3.140  1   0.076
## 4  .p6. == .p29. 1.169  1   0.280
## 5 .p20. == .p43. 6.006  1   0.014
## 6 .p21. == .p44. 0.668  1   0.414
## 7 .p22. == .p45. 3.220  1   0.073
## 8 .p23. == .p46. 1.823  1   0.177
```

Equality restrictions on intercepts are tenable according to CFI, RMSEA and AIC, but not according to the $\Delta\chi^2$. If there is any non-invariance, its in parameters .p20. and .p2., which are the intercept and loading of Ind2. For now, we go by CFI, RMSEA and AIC and conclude equality of item intercepts. We continue with testing the equality of residual variances:

```
ex43.strict.fit <- cfa(ex43.mod, sample.cov = ex43.cov.list,
                       sample.nobs = ex43.n.list, sample.mean = ex43.mean.list,
                       meanstructure = TRUE,
                       group.equal = c("loadings", "intercepts", "residuals"))
fitMeasures(ex43.strict.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   77.943   30.000    0.000    0.987    0.065    0.033 5876.086
```

```
lavTestLRT(ex43.strong.fit, ex43.strict.fit)
```

```
## Chi Square Difference Test
##
##                 Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## ex43.strong.fit 24 5869.0 6007.9 58.808
## ex43.strict.fit 30 5876.1 5987.3 77.943     19.135       6   0.003942 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to CFI, there is not significant deterioration of model fit, but according to the χ^2, there is, and according to RMSEA and AIC, the fit has deteriorated as well.

```
lavTestScore(ex43.strict.fit)
```

```
## $test
##
## total score test:
##
##    test     X2 df p.value
## 1 score 31.527 14   0.005
##
## $uni
##
## univariate score tests:
##
##      lhs op   rhs    X2 df p.value
## 1   .p2. == .p25. 6.434  1   0.011
```

```
## 2   .p3. == .p26. 0.819  1   0.365
## 3   .p5. == .p28. 2.567  1   0.109
## 4   .p6. == .p29. 0.955  1   0.328
## 5  .p11. == .p34. 7.486  1   0.006
## 6  .p12. == .p35. 0.735  1   0.391
## 7  .p13. == .p36. 5.891  1   0.015
## 8  .p14. == .p37. 0.860  1   0.354
## 9  .p15. == .p38. 7.375  1   0.007
## 10 .p16. == .p39. 1.595  1   0.207
## 11 .p20. == .p43. 5.983  1   0.014
## 12 .p21. == .p44. 0.664  1   0.415
## 13 .p22. == .p45. 3.218  1   0.073
## 14 .p23. == .p46. 1.814  1   0.178
```

Most problematic equality restrictions is on parameter .p11., which is the residual variance of Ind1, let's release its equality restriction:

```r
ex43.strict.fit2 <- cfa(ex43.mod, sample.cov = ex43.cov.list,
                        sample.nobs = ex43.n.list, sample.mean = ex43.mean.list,
                        meanstructure = TRUE,
                        group.equal = c("loadings", "intercepts", "residuals"),
                        group.partial = c("Ind1 ~~ Ind1"))
fitMeasures(ex43.strict.fit2, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   70.278   29.000    0.000    0.988    0.061    0.034 5870.420
```

```r
lavTestLRT(ex43.strong.fit, ex43.strict.fit2)
```

```
## Chi Square Difference Test
##
##                  Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## ex43.strong.fit  24 5869.0 6007.9 58.808
## ex43.strict.fit2 29 5870.4 5986.2 70.278      11.47       5    0.04283 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

RMSEA indicates improvement of fit, compared to the strong invariance model. Also, the difference in CFI values is $< .01$. AIC and χ^2 indicate a deterioration of model fit, let's check out the modification indices:

```r
lavTestScore(ex43.strict.fit2)
```

```
## $test
##
## total score test:
##
##    test     X2 df p.value
## 1 score 23.797 13   0.033
##
## $uni
##
## univariate score tests:
##
##      lhs op   rhs    X2 df p.value
## 1   .p2. == .p25. 6.671  1   0.010
## 2   .p3. == .p26. 0.684  1   0.408
## 3   .p5. == .p28. 2.566  1   0.109
```

```
## 4    .p6. == .p29. 0.955  1   0.328
## 5   .p12. == .p35. 0.288  1   0.592
## 6   .p13. == .p36. 3.364  1   0.067
## 7   .p14. == .p37. 0.858  1   0.354
## 8   .p15. == .p38. 7.358  1   0.007
## 9   .p16. == .p39. 1.608  1   0.205
## 10  .p20. == .p43. 5.996  1   0.014
## 11  .p21. == .p44. 0.654  1   0.419
## 12  .p22. == .p45. 3.219  1   0.073
## 13  .p23. == .p46. 1.815  1   0.178
```

Most problematic equality restrictions are on parameters .p11. and .p15., which are the residual variance of Ind1 and Ind5.

Do we get the same result with standardizing the LVs?

```
ex43.mod.stdLV <- '
  F1 =~ Ind1 + Ind2 + Ind3
  F2 =~ Ind4 + Ind5 + Ind6
'
ex43.conf.fit.stdLV <- cfa(ex43.mod, sample.cov = ex43.cov.list,
                           sample.nobs = ex43.n.list,
                           sample.mean = ex43.mean.list,
                           std.lv = TRUE, meanstructure = TRUE)
fitMeasures(ex43.conf.fit.stdLV, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   46.249   16.000    0.000    0.992    0.071    0.029 5872.392
```

```
fitMeasures(ex43.conf.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   46.249   16.000    0.000    0.992    0.071    0.029 5872.392
```

Unsurprisingly, a different identification method yields the exact same model fit.

```
ex43.weak.fit.stdLV <- cfa(ex43.mod, sample.cov = ex43.cov.list, std.lv = TRUE,
                           sample.nobs = ex43.n.list, sample.mean=ex43.mean.list,
                           meanstructure = TRUE, group.equal = "loadings")
```

```
## Warning in lav_partable_flat(FLAT, blocks = "group", meanstructure =
## meanstructure, : lavaan WARNING: std.lv = TRUE forces all variances to be
## unity in all groups, despite group.equal = "loadings"
```

```
fitMeasures(ex43.weak.fit.stdLV, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   54.602   22.000    0.000    0.991    0.062    0.058 5868.745
```

```
fitMeasures(ex43.weak.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   49.036   20.000    0.000    0.992    0.062    0.032 5867.179
```

Here, we do see a difference in model fit: Whereas the marker-variable identification method allows the variances of the LVs to be different between groups, the standardized-LV identification method restricts loadings as well as LV variances to be equal across groups. However, the conclusion is the same: equality restrictions on loadings are tenable. We continue with testing the equality of intercepts:

```r
ex43.strong.fit.stdLV <- cfa(ex43.mod, sample.cov = ex43.cov.list, std.lv = TRUE,
                             sample.nobs = ex43.n.list,
                             sample.mean = ex43.mean.list,
                             meanstructure = TRUE,
                             group.equal = c("loadings","intercepts"))
```

```
## Warning in lav_partable_flat(FLAT, blocks = "group", meanstructure =
## meanstructure, : lavaan WARNING: std.lv = TRUE forces all variances to be
## unity in all groups, despite group.equal = "loadings"
```

```r
fitMeasures(ex43.strong.fit.stdLV, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   64.399   26.000    0.000    0.989    0.062    0.059 5870.542
```

```r
fitMeasures(ex43.strong.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   58.808   24.000    0.000    0.990    0.062    0.033 5868.951
```

```r
lavTestLRT(ex43.strong.fit.stdLV, ex43.weak.fit)
```

```
## Chi Square Difference Test
##
##                       Df    AIC    BIC  Chisq Chisq diff Df diff
## ex43.weak.fit         20 5867.2 6024.7 49.036
## ex43.strong.fit.stdLV 26 5870.5 6000.2 64.399     15.363       6
##                       Pr(>Chisq)
## ex43.weak.fit
## ex43.strong.fit.stdLV    0.01761 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lavTestScore(ex43.strong.fit.stdLV)
```

```
## $test
##
## total score test:
##
##    test     X2 df p.value
## 1 score 18.079 10   0.054
##
## $uni
##
## univariate score tests:
##
##      lhs op   rhs    X2 df p.value
## 1   .p1. == .p24. 2.876  1   0.090
## 2   .p2. == .p25. 6.509  1   0.011
## 3   .p3. == .p26. 1.133  1   0.287
## 4   .p4. == .p27. 0.581  1   0.446
## 5   .p5. == .p28. 2.687  1   0.101
## 6   .p6. == .p29. 1.526  1   0.217
## 7  .p20. == .p43. 6.027  1   0.014
## 8  .p21. == .p44. 0.669  1   0.413
## 9  .p22. == .p45. 3.223  1   0.073
## 10 .p23. == .p46. 1.826  1   0.177
```

The model fit is again slightly different: Whereas the marker-variable identification method allows the means of the LVs to be different between groups, the standardized-LV identification method restricts item intercepts as well as LV means to be equal across groups. We do obtain the same conlclusion, though. If any parameters are different betweern groups, its .p2 and .p20.

```
ex43.strict.fit.stdLV <- cfa(ex43.mod, sample.cov = ex43.cov.list,
                             sample.nobs = ex43.n.list,
                             sample.mean = ex43.mean.list,
                             meanstructure = TRUE, std.lv = TRUE,
                             group.equal = c("loadings", "intercepts", "residuals"))
```

```
## Warning in lav_partable_flat(FLAT, blocks = "group", meanstructure =
## meanstructure, : lavaan WARNING: std.lv = TRUE forces all variances to be
## unity in all groups, despite group.equal = "loadings"
```

```
fitMeasures(ex43.strict.fit.stdLV, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   83.316   32.000    0.000    0.986    0.065    0.059 5877.459
```

```
fitMeasures(ex43.strict.fit, fit.indices)
```

```
##    chisq       df   pvalue      cfi    rmsea     srmr      aic
##   77.943   30.000    0.000    0.987    0.065    0.033 5876.086
```

```
lavTestLRT(ex43.strong.fit.stdLV, ex43.strict.fit)
```

```
## Chi Square Difference Test
##
##                       Df    AIC    BIC  Chisq Chisq diff Df diff
## ex43.strong.fit.stdLV 26 5870.5 6000.2 64.399
## ex43.strict.fit       30 5876.1 5987.3 77.943     13.544       4
##                       Pr(>Chisq)
## ex43.strong.fit.stdLV
## ex43.strict.fit         0.008902 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lavTestScore(ex43.strict.fit.stdLV)
```

```
## $test
##
## total score test:
##
##    test     X2 df p.value
## 1 score 36.863 16   0.002
##
## $uni
##
## univariate score tests:
##
##      lhs op   rhs    X2 df p.value
## 1   .p1. == .p24. 2.731  1   0.098
## 2   .p2. == .p25. 6.047  1   0.014
## 3   .p3. == .p26. 1.015  1   0.314
## 4   .p4. == .p27. 0.496  1   0.481
## 5   .p5. == .p28. 2.144  1   0.143
## 6   .p6. == .p29. 1.314  1   0.252
```

```
## 7  .p11. == .p34. 7.190  1   0.007
## 8  .p12. == .p35. 0.653  1   0.419
## 9  .p13. == .p36. 5.454  1   0.020
## 10 .p14. == .p37. 0.935  1   0.334
## 11 .p15. == .p38. 7.635  1   0.006
## 12 .p16. == .p39. 1.826  1   0.177
## 13 .p20. == .p43. 5.984  1   0.014
## 14 .p21. == .p44. 0.664  1   0.415
## 15 .p22. == .p45. 3.220  1   0.073
## 16 .p23. == .p46. 1.818  1   0.178
```

Again, we see that the most problematic equality restrictions are on parameters .p11. and .p15.

Note that I did not closely follow Beaujeans assignment: for the marker identification method, we had to use items 1 and 5. That would have yielded different results.

Additional exercise: HADS

In the HADS anxiety subscale exercise in week 3 (IRT), we used a unidimensional model. That model did not fit very well. Therefore, we are going to use a two-dimensional model, suggested by Barth and Martin (2005). It consists of a Psychomotor Agitation (PAG) and a Psychic Anxiety (ANX) factor.

These are the items of the HADS: 1. I feel tense or wound up 2. I get a sort of frightened feeling as if something bad is about to happen 3. Worrying thoughts go through my mind 4. I can sit at ease and feel relaxed 5. I get a sort of frightened feeling like butterflies in the stomach 6. I feel restless and have to be on the move 7. I get sudden feelings of panic

a) Assess measurement invariance of the HADS Anxiety items with respect to gender ('geslacht'). Describe and interpret any differences you found.

```
library(foreign)
HADS <- read.spss("HADS.sav", use.value.labels = TRUE, to.data.frame = TRUE)
```

```
## re-encoding from UTF-8
```

```
HADS.mod <- '
  PAG =~ HADS1 + HADS4 + HADS6
  ANX =~ HADS2 + HADS3 + HADS5 + HADS7
'
HADS.fit.conf <- cfa(HADS.mod, data = HADS, group="geslacht", ordered = paste0("HADS", 1:7))
summary(HADS.fit.conf, standardized = TRUE)
```

```
## lavaan (0.6-1) converged normally after  25 iterations
##
##   Number of observations per group
##   een vrouw                                        285
##   een man                                          217
##
##   Estimator                                       DWLS      Robust
##   Model Fit Test Statistic                      48.206      91.152
##   Degrees of freedom                                26          26
##   P-value (Chi-square)                           0.005       0.000
##   Scaling correction factor                                  0.548
##   Shift parameter for each group:
##     een vrouw                                                1.817
##     een man                                                  1.383
##     for simple second-order correction (Mplus variant)
##
## Chi-square for each group:
##
##   een vrouw                                     31.345      59.005
##   een man                                       16.861      32.146
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##   Standard Errors                           Robust.sem
##
##
## Group 1 [een vrouw ]:
##
```

```
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   PAG =~
##     HADS1             1.000                               0.894    0.894
##     HADS4             0.777    0.049   15.968    0.000    0.695    0.695
##     HADS6             0.901    0.043   20.863    0.000    0.805    0.805
##   ANX =~
##     HADS2             1.000                               0.836    0.836
##     HADS3             0.989    0.048   20.522    0.000    0.826    0.826
##     HADS5             0.781    0.057   13.709    0.000    0.653    0.653
##     HADS7             0.944    0.051   18.590    0.000    0.789    0.789
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   PAG ~~
##     ANX               0.592    0.039   15.350    0.000    0.792    0.792
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .HADS1             0.000                               0.000    0.000
##    .HADS4             0.000                               0.000    0.000
##    .HADS6             0.000                               0.000    0.000
##    .HADS2             0.000                               0.000    0.000
##    .HADS3             0.000                               0.000    0.000
##    .HADS5             0.000                               0.000    0.000
##    .HADS7             0.000                               0.000    0.000
##     PAG               0.000                               0.000    0.000
##     ANX               0.000                               0.000    0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     HADS1|t1         -1.355    0.105  -12.857    0.000   -1.355   -1.355
##     HADS1|t2         -0.317    0.076   -4.190    0.000   -0.317   -0.317
##     HADS1|t3          0.893    0.086   10.354    0.000    0.893    0.893
##     HADS4|t1         -1.501    0.114  -13.110    0.000   -1.501   -1.501
##     HADS4|t2         -0.734    0.082   -8.932    0.000   -0.734   -0.734
##     HADS4|t3          0.392    0.077    5.129    0.000    0.392    0.392
##     HADS6|t1         -1.063    0.092  -11.571    0.000   -1.063   -1.063
##     HADS6|t2          0.119    0.075    1.596    0.111    0.119    0.119
##     HADS6|t3          1.048    0.091   11.475    0.000    1.048    1.048
##     HADS2|t1         -0.226    0.075   -3.013    0.003   -0.226   -0.226
##     HADS2|t2          0.550    0.079    6.993    0.000    0.550    0.550
##     HADS2|t3          1.529    0.116   13.130    0.000    1.529    1.529
##     HADS3|t1         -1.095    0.093  -11.760    0.000   -1.095   -1.095
##     HADS3|t2         -0.084    0.074   -1.123    0.261   -0.084   -0.084
##     HADS3|t3          1.048    0.091   11.475    0.000    1.048    1.048
##     HADS5|t1         -0.440    0.077   -5.714    0.000   -0.440   -0.440
##     HADS5|t2          0.560    0.079    7.109    0.000    0.560    0.560
##     HADS5|t3          1.529    0.116   13.130    0.000    1.529    1.529
##     HADS7|t1         -0.199    0.075   -2.659    0.008   -0.199   -0.199
##     HADS7|t2          0.769    0.083    9.266    0.000    0.769    0.769
##     HADS7|t3          1.910    0.152   12.543    0.000    1.910    1.910
##
## Variances:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .HADS1            0.202                                0.202    0.202
##    .HADS4            0.518                                0.518    0.518
##    .HADS6            0.351                                0.351    0.351
##    .HADS2            0.302                                0.302    0.302
##    .HADS3            0.317                                0.317    0.317
##    .HADS5            0.574                                0.574    0.574
##    .HADS7            0.377                                0.377    0.377
##     PAG              0.798    0.044   18.094    0.000    1.000    1.000
##     ANX              0.698    0.045   15.562    0.000    1.000    1.000
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     HADS1            1.000                                1.000    1.000
##     HADS4            1.000                                1.000    1.000
##     HADS6            1.000                                1.000    1.000
##     HADS2            1.000                                1.000    1.000
##     HADS3            1.000                                1.000    1.000
##     HADS5            1.000                                1.000    1.000
##     HADS7            1.000                                1.000    1.000
##
##
## Group 2 [een man]:
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   PAG =~
##     HADS1            1.000                                0.888    0.888
##     HADS4            0.703    0.069   10.169    0.000    0.624    0.624
##     HADS6            0.814    0.053   15.214    0.000    0.722    0.722
##   ANX =~
##     HADS2            1.000                                0.805    0.805
##     HADS3            1.052    0.062   16.992    0.000    0.848    0.848
##     HADS5            0.760    0.073   10.351    0.000    0.612    0.612
##     HADS7            0.980    0.052   18.858    0.000    0.789    0.789
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   PAG ~~
##     ANX              0.613    0.044   14.072    0.000    0.858    0.858
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .HADS1            0.000                                0.000    0.000
##    .HADS4            0.000                                0.000    0.000
##    .HADS6            0.000                                0.000    0.000
##    .HADS2            0.000                                0.000    0.000
##    .HADS3            0.000                                0.000    0.000
##    .HADS5            0.000                                0.000    0.000
##    .HADS7            0.000                                0.000    0.000
##     PAG              0.000                                0.000    0.000
##     ANX              0.000                                0.000    0.000
##
## Thresholds:
```

```
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     HADS1|t1        -1.386    0.123  -11.283    0.000   -1.386   -1.386
##     HADS1|t2        -0.145    0.086   -1.693    0.090   -0.145   -0.145
##     HADS1|t3         0.832    0.097    8.585    0.000    0.832    0.832
##     HADS4|t1        -1.596    0.139  -11.465    0.000   -1.596   -1.596
##     HADS4|t2        -0.800    0.096   -8.335    0.000   -0.800   -0.800
##     HADS4|t3         0.435    0.088    4.929    0.000    0.435    0.435
##     HADS6|t1        -1.176    0.111  -10.638    0.000   -1.176   -1.176
##     HADS6|t2         0.075    0.085    0.881    0.379    0.075    0.075
##     HADS6|t3         1.274    0.116   10.997    0.000    1.274    1.274
##     HADS2|t1        -0.133    0.086   -1.558    0.119   -0.133   -0.133
##     HADS2|t2         0.678    0.093    7.310    0.000    0.678    0.678
##     HADS2|t3         1.449    0.127   11.386    0.000    1.449    1.449
##     HADS3|t1        -0.971    0.102   -9.555    0.000   -0.971   -0.971
##     HADS3|t2         0.098    0.085    1.152    0.249    0.098    0.098
##     HADS3|t3         1.088    0.106   10.229    0.000    1.088    1.088
##     HADS5|t1        -0.275    0.086   -3.180    0.001   -0.275   -0.275
##     HADS5|t2         0.448    0.088    5.063    0.000    0.448    0.448
##     HADS5|t3         1.356    0.121   11.221    0.000    1.356    1.356
##     HADS7|t1        -0.348    0.087   -3.989    0.000   -0.348   -0.348
##     HADS7|t2         0.693    0.093    7.439    0.000    0.693    0.693
##     HADS7|t3         1.849    0.166   11.114    0.000    1.849    1.849
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .HADS1            0.212                               0.212    0.212
##    .HADS4            0.611                               0.611    0.611
##    .HADS6            0.478                               0.478    0.478
##    .HADS2            0.351                               0.351    0.351
##    .HADS3            0.282                               0.282    0.282
##    .HADS5            0.625                               0.625    0.625
##    .HADS7            0.377                               0.377    0.377
##     PAG              0.788    0.060   13.139    0.000    1.000    1.000
##     ANX              0.649    0.055   11.806    0.000    1.000    1.000
##
## Scales y*:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     HADS1            1.000                               1.000    1.000
##     HADS4            1.000                               1.000    1.000
##     HADS6            1.000                               1.000    1.000
##     HADS2            1.000                               1.000    1.000
##     HADS3            1.000                               1.000    1.000
##     HADS5            1.000                               1.000    1.000
##     HADS7            1.000                               1.000    1.000
```

The female group contributes more strongly to the χ^2 value, which is to be expected, as this is also the largest group. In both female and male groups, we see substantial and significant loadings for all items. Also, the correlations between the PAG and ANX factors are significant and substantial.

```r
indices <- c("chisq.scaled", "df", "pvalue.scaled", "cfi.scaled", "srmr",
             "rmsea.scaled", "rmsea.ci.lower.scaled", "rmsea.ci.upper.scaled")
fitMeasures(HADS.fit.conf, indices)
```

```
##          chisq.scaled                    df         pvalue.scaled
##                91.152                26.000                 0.000
```

```
##            cfi.scaled                  srmr          rmsea.scaled
##                 0.982                 0.047                 0.100
## rmsea.ci.lower.scaled rmsea.ci.upper.scaled
##                 0.078                 0.123
```

```r
modificationindices(HADS.fit.conf, sort = TRUE)[1:10,]
```

```
##       lhs op   rhs block group level    mi    epc sepc.lv sepc.all
## 113   ANX =~ HADS1     1     1     1 7.602  0.634   0.530    0.530
## 122 HADS4 ~~ HADS6     1     1     1 7.601  0.156   0.156    0.366
## 119 HADS1 ~~ HADS3     1     1     1 7.294  0.145   0.145    0.575
## 109   PAG =~ HADS2     1     1     1 7.009 -0.477  -0.426   -0.426
## 111   PAG =~ HADS5     1     1     1 6.904  0.430   0.384    0.384
## 133 HADS2 ~~ HADS7     1     1     1 6.242  0.139   0.139    0.411
## 138   PAG =~ HADS3     2     2     1 5.467  0.660   0.586    0.586
## 164 HADS5 ~~ HADS7     2     2     1 5.148  0.141   0.141    0.291
## 134 HADS3 ~~ HADS5     1     1     1 4.768 -0.145  -0.145   -0.341
## 132 HADS2 ~~ HADS5     1     1     1 4.519 -0.144  -0.144   -0.346
##     sepc.nox
## 113    0.530
## 122    0.366
## 119    0.575
## 109   -0.426
## 111    0.384
## 133    0.411
## 138    0.586
## 164    0.291
## 134   -0.341
## 132   -0.346
```

CFI and SRMR indicate a well-fitting model, RMSEA does not. Graded Response Models are not very parsimonious by definition: a loading and multiple thresholds are estimated for every item. This often yields a relatively high RMSEA in these models. Modification indices do not indicate the same parameters should be added for males and females. So we proceed by assessing the equality of loadings:

```r
HADS.fit.metr <- cfa(HADS.mod, data = HADS, group = "geslacht",
                     ordered = paste0("HADS", 1:7),
                     group.equal = "loadings")
fitMeasures(HADS.fit.metr, indices)
```

```
##           chisq.scaled                    df          pvalue.scaled
##                 78.449                31.000                  0.000
##            cfi.scaled                  srmr          rmsea.scaled
##                 0.987                 0.049                 0.078
## rmsea.ci.lower.scaled rmsea.ci.upper.scaled
##                 0.057                 0.100
```

```r
lavTestLRT(HADS.fit.metr, HADS.fit.conf)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##               Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## HADS.fit.conf 26         48.206
## HADS.fit.metr 31         51.437     2.7994       5     0.7309
```

Model fit according to RMSEA has substantially improved, model fit has also improved according to CFI. The difference in χ^2 values is also not significant.

```r
HADS.fit.scal <- cfa(HADS.mod, data = HADS, group = "geslacht",
                     ordered = paste0("HADS", 1:7),
                     group.equal = c("loadings", "thresholds"))
fitMeasures(HADS.fit.scal, indices)
```

```
##          chisq.scaled                   df        pvalue.scaled
##               105.304               43.000                0.000
##            cfi.scaled                 srmr         rmsea.scaled
##                 0.982                0.048                0.076
## rmsea.ci.lower.scaled rmsea.ci.upper.scaled
##                 0.058                0.095
```

```r
lavTestLRT(HADS.fit.metr, HADS.fit.scal)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##               Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## HADS.fit.metr 31         51.437
## HADS.fit.scal 43         61.991     16.765      12     0.1587
```

The difference in model fit is not significant. Also, CFI and SRMR indicate a well-fitting model, RMSEA value approaches an acceptable level.

We conclude that factor loadings and item thresholds (i.e., discrimination and difficulty parameters are equal across gender.

b) Assess structural invariance of the HADS Anxiety factor with respect to gender (‘geslacht’). Describe and interpret any differences you found.

```r
HADS.fit.var <- cfa(HADS.mod, data = HADS, group = "geslacht",
                     ordered = paste0("HADS", 1:7),
                     group.equal = c("loadings", "thresholds", "lv.variances"))
fitMeasures(HADS.fit.var, indices)
```

```
##          chisq.scaled                   df        pvalue.scaled
##                98.009               45.000                0.000
##            cfi.scaled                 srmr         rmsea.scaled
##                 0.985                0.049                0.069
## rmsea.ci.lower.scaled rmsea.ci.upper.scaled
##                 0.050                0.087
```

```r
lavTestLRT(HADS.fit.var, HADS.fit.scal)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##               Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## HADS.fit.scal 43         61.991
## HADS.fit.var  45         67.979     2.4066       2     0.3002
```

Equal latent variances seems tenable.

```r
HADS.fit.covar <- cfa(HADS.mod, data = HADS, group = "geslacht",
                     ordered = paste0("HADS", 1:7),
                     group.equal = c("loadings", "thresholds", "lv.variances",
                                     "lv.covariances"))
fitMeasures(HADS.fit.covar, indices)
```

```
##          chisq.scaled                   df        pvalue.scaled
##                95.409               46.000                0.000
```

```
##            cfi.scaled                  srmr          rmsea.scaled
##                 0.986                 0.049                 0.066
## rmsea.ci.lower.scaled rmsea.ci.upper.scaled
##                 0.047                 0.084
```

```
lavTestLRT(HADS.fit.var, HADS.fit.covar)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##                 Df AIC BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## HADS.fit.var   45         67.979
## HADS.fit.covar 46         69.570    0.95187       1     0.3292
```

Equal latent covariances seems tenable also.

```
HADS.fit.means <- cfa(HADS.mod, data = HADS, group = "geslacht",
                      ordered = paste0("HADS", 1:7),
                      group.equal = c("loadings", "thresholds", "lv.variances",
                                      "lv.covariances", "means"))
fitMeasures(HADS.fit.means, indices)
```

```
##          chisq.scaled                    df          pvalue.scaled
##                79.670                48.000                 0.003
##            cfi.scaled                  srmr          rmsea.scaled
##                 0.991                 0.049                 0.051
## rmsea.ci.lower.scaled rmsea.ci.upper.scaled
##                 0.030                 0.071
```

```
lavTestLRT(HADS.fit.means, HADS.fit.covar)
```

```
## Scaled Chi Square Difference Test (method = "satorra.2000")
##
##                 Df AIC BIC Chisq Chisq diff Df diff Pr(>Chisq)
## HADS.fit.covar 46         69.57
## HADS.fit.means 48         70.60    0.59397       2     0.7431
```

Equal latent means seems tenable also.

c) Fit one single model, in which you assess the main and interaction effects of gender ('geslacht') and age ('leeftijd') on Psychomotor Agitation and Psychic Anxiety levels.

First we create a dummay variable for gender and the interaction between gender and age:

```
HADS$geslacht <- as.numeric(HADS$geslacht) - 1
HADS$interact <- HADS$geslacht * HADS$leeftijd
HADS.mod.main <- '
  PAG =~ HADS1 + HADS4 + HADS6
  ANX =~ HADS2 + HADS3 + HADS5 + HADS7
  PAG ~ geslacht + leeftijd
  ANX ~ geslacht + leeftijd
'
HADS.fit.main <- cfa(HADS.mod.main, data = HADS, ordered = paste0("HADS", 1:7))
summary(HADS.fit.main)
```

```
## lavaan (0.6-1) converged normally after  29 iterations
##
##   Number of observations                           502
##
##   Estimator                                       DWLS      Robust
```

```
##   Model Fit Test Statistic                        47.026      90.963
##   Degrees of freedom                                  23          23
##   P-value (Chi-square)                             0.002       0.000
##   Scaling correction factor                                    0.530
##   Shift parameter                                              2.247
##     for simple second-order correction (Mplus variant)
##
## Parameter Estimates:
##
##   Information                                   Expected
##   Information saturated (h1) model          Unstructured
##   Standard Errors                             Robust.sem
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   PAG =~
##     HADS1             1.000
##     HADS4             0.741    0.041   17.990    0.000
##     HADS6             0.860    0.035   24.606    0.000
##   ANX =~
##     HADS2             1.000
##     HADS3             1.022    0.039   26.215    0.000
##     HADS5             0.767    0.046   16.714    0.000
##     HADS7             0.960    0.036   26.324    0.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   PAG ~
##     geslacht          0.010    0.093    0.108    0.914
##     leeftijd         -0.003    0.003   -0.822    0.411
##   ANX ~
##     geslacht          0.007    0.082    0.090    0.928
##     leeftijd         -0.007    0.003   -2.275    0.023
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .PAG ~~
##    .ANX               0.600    0.029   20.876    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .HADS1             0.000
##    .HADS4             0.000
##    .HADS6             0.000
##    .HADS2             0.000
##    .HADS3             0.000
##    .HADS5             0.000
##    .HADS7             0.000
##    .PAG               0.000
##    .ANX               0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     HADS1|t1         -1.549    0.187   -8.288    0.000
```

```
##     HADS1|t2        -0.421    0.175   -2.409    0.016
##     HADS1|t3         0.689    0.176    3.927    0.000
##     HADS4|t1        -1.629    0.200   -8.158    0.000
##     HADS4|t2        -0.850    0.188   -4.530    0.000
##     HADS4|t3         0.322    0.186    1.738    0.082
##     HADS6|t1        -1.125    0.182   -6.166    0.000
##     HADS6|t2         0.086    0.180    0.476    0.634
##     HADS6|t3         1.124    0.187    6.010    0.000
##     HADS2|t1        -0.429    0.193   -2.221    0.026
##     HADS2|t2         0.363    0.194    1.874    0.061
##     HADS2|t3         1.257    0.188    6.696    0.000
##     HADS3|t1        -1.332    0.185   -7.204    0.000
##     HADS3|t2        -0.291    0.181   -1.604    0.109
##     HADS3|t3         0.787    0.182    4.318    0.000
##     HADS5|t1        -0.582    0.183   -3.173    0.002
##     HADS5|t2         0.298    0.183    1.627    0.104
##     HADS5|t3         1.237    0.190    6.499    0.000
##     HADS7|t1        -0.668    0.193   -3.458    0.001
##     HADS7|t2         0.336    0.192    1.746    0.081
##     HADS7|t3         1.490    0.207    7.193    0.000
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .HADS1            0.201
##    .HADS4            0.561
##    .HADS6            0.409
##    .HADS2            0.329
##    .HADS3            0.300
##    .HADS5            0.605
##    .HADS7            0.382
##    .PAG              0.799    0.037   21.429    0.000
##    .ANX              0.671    0.035   19.097    0.000
##
## Scales y*:
##                   Estimate  Std.Err  z-value  P(>|z|)
##     HADS1            1.000
##     HADS4            1.000
##     HADS6            1.000
##     HADS2            1.000
##     HADS3            1.000
##     HADS5            1.000
##     HADS7            1.000
```

```
fitMeasures(HADS.fit.main, indices)
```

```
##          chisq.scaled                    df         pvalue.scaled
##                90.963                23.000                 0.000
##            cfi.scaled                  srmr          rmsea.scaled
##                 0.981                 0.045                 0.077
## rmsea.ci.lower.scaled rmsea.ci.upper.scaled
##                 0.061                 0.094
```

We see a positive significant effect of age on ANX, but it is very small. The model fits well according to CFI and SRMR, and acceptable according to RMSEA.

```r
HADS.mod.int <- '
  PAG =~ HADS1 + HADS4 + HADS6
  ANX =~ HADS2 + HADS3 + HADS5 + HADS7
  PAG ~ interact + geslacht + leeftijd
  ANX ~ interact + geslacht + leeftijd
'
HADS.fit.int <- cfa(HADS.mod.int, data = HADS, ordered = paste0("HADS", 1:7))
summary(HADS.fit.int)
```

```
## lavaan (0.6-1) converged normally after  42 iterations
##
##   Number of observations                           502
##
##   Estimator                                       DWLS      Robust
##   Model Fit Test Statistic                      55.151      88.563
##   Degrees of freedom                                28          28
##   P-value (Chi-square)                           0.002       0.000
##   Scaling correction factor                                  0.690
##   Shift parameter                                            8.667
##     for simple second-order correction (Mplus variant)
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Unstructured
##   Standard Errors                           Robust.sem
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   PAG =~
##     HADS1             1.000
##     HADS4             0.740    0.041   17.967    0.000
##     HADS6             0.859    0.035   24.665    0.000
##   ANX =~
##     HADS2             1.000
##     HADS3             1.025    0.039   26.345    0.000
##     HADS5             0.774    0.046   16.763    0.000
##     HADS7             0.963    0.037   26.218    0.000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   PAG ~
##     interact          0.015    0.007    2.079    0.038
##     geslacht         -0.623    0.317   -1.969    0.049
##     leeftijd         -0.012    0.005   -2.170    0.030
##   ANX ~
##     interact          0.014    0.006    2.153    0.031
##     geslacht         -0.591    0.294   -2.009    0.045
##     leeftijd         -0.015    0.005   -3.122    0.002
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##  .PAG ~~
##    .ANX               0.596    0.029   20.623    0.000
```

```
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .HADS1             0.000
##    .HADS4             0.000
##    .HADS6             0.000
##    .HADS2             0.000
##    .HADS3             0.000
##    .HADS5             0.000
##    .HADS7             0.000
##    .PAG               0.000
##    .ANX               0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     HADS1|t1         -1.939    0.268   -7.242    0.000
##     HADS1|t2         -0.804    0.256   -3.142    0.002
##     HADS1|t3          0.309    0.257    1.203    0.229
##     HADS4|t1         -1.902    0.275   -6.907    0.000
##     HADS4|t2         -1.119    0.270   -4.152    0.000
##     HADS4|t3          0.057    0.266    0.213    0.832
##     HADS6|t1         -1.497    0.291   -5.149    0.000
##     HADS6|t2         -0.281    0.286   -0.984    0.325
##     HADS6|t3          0.759    0.290    2.620    0.009
##     HADS2|t1         -0.625    0.276   -2.263    0.024
##     HADS2|t2          0.167    0.277    0.605    0.545
##     HADS2|t3          1.061    0.277    3.826    0.000
##     HADS3|t1         -1.870    0.286   -6.539    0.000
##     HADS3|t2         -0.819    0.278   -2.945    0.003
##     HADS3|t3          0.264    0.278    0.948    0.343
##     HADS5|t1         -1.201    0.279   -4.303    0.000
##     HADS5|t2         -0.314    0.278   -1.128    0.259
##     HADS5|t3          0.635    0.278    2.288    0.022
##     HADS7|t1         -0.786    0.294   -2.671    0.008
##     HADS7|t2          0.218    0.295    0.740    0.459
##     HADS7|t3          1.371    0.314    4.372    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .HADS1             0.203
##    .HADS4             0.563
##    .HADS6             0.412
##    .HADS2             0.333
##    .HADS3             0.299
##    .HADS5             0.600
##    .HADS7             0.382
##    .PAG               0.797    0.037   21.368    0.000
##    .ANX               0.667    0.035   18.987    0.000
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     HADS1             1.000
##     HADS4             1.000
##     HADS6             1.000
```

```
##      HADS2              1.000
##      HADS3              1.000
##      HADS5              1.000
##      HADS7              1.000
```

Magically, after adding the interaction effect, the effect of gender has now become significant. Males appear to have lower PAG and ANX. The interaction effect is also significant: for females, age has a positive association with ANX and PAG. This effect was not observed for males. The effects are very modest. Also, we are taking a forward selection approach to regression here, yielding incorrect p-values, so we should take these results with a grain of salt.