Appendix B) Example 2: Testing equality of direct effects in a path model (random-effects analysis)

Contents

Install the metaSEM package

- R can be downloaded at http://www.r-project.org/.
- We only need to install the metaSEM-package once.

```
install.packages("metaSEM")
```

Read in the data and load metaSEM package

```
library(metaSEM)
```

```
## Warning: package 'metaSEM' was built under R version 3.3.3
```

```
## Warning: package 'OpenMx' was built under R version 3.3.3
```

```
## Load the functions to facilitate analysis
source("http://www.suzannejak.nl/subgroup.functions.R")

head(Roorda11$data)
```

```
## [[1]]
##          pos   neg enga achiev
## pos     1.00 -0.54   NA   0.18
## neg    -0.54  1.00   NA  -0.29
## enga      NA    NA   NA     NA
## achiev  0.18 -0.29   NA   1.00
##
## [[2]]
##         pos neg enga achiev
## pos    1.00  NA 0.64   0.29
## neg      NA  NA   NA     NA
## enga   0.64  NA 1.00   0.23
## achiev 0.29  NA 0.23   1.00
##
## [[3]]
##         pos neg enga achiev
```

```
## pos    1.00  NA 0.29     NA
## neg      NA  NA   NA     NA
## enga   0.29  NA 1.00     NA
## achiev   NA  NA   NA     NA
##
## [[4]]
##         pos neg enga achiev
## pos    1.00  NA 0.29     NA
## neg      NA  NA   NA     NA
## enga   0.29  NA 1.00     NA
## achiev   NA  NA   NA     NA
##
## [[5]]
##         pos   neg  enga achiev
## pos      NA    NA  0.22   0.08
## neg      NA  1.00 -0.45  -0.24
## enga   0.22 -0.45    NA     NA
## achiev 0.08 -0.24    NA   1.00
##
## [[6]]
##          pos neg enga achiev
## pos     1.00  NA 0.06  -0.09
## neg       NA  NA   NA     NA
## enga    0.06  NA 1.00   0.20
## achiev -0.09  NA 0.20   1.00
```

```
head(Roorda11$n)
```

```
## [1] 1310  427  123   66  179   93
```

```
head(Roorda11$SES)
```

```
## [1] 70 78 83 30 27 39
```

Stage 1 random-effects model on all data

```
## Stage 1 analysis overall (random)
stage1random <- tssem1(my.df = Roorda11$data, n = Roorda11$n, method = "REM", RE.type = "Diag")
summary(stage1random)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(x = paste(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es, sep = "")), RE.lbound = RE.lbound,
##     I2 = I2, model.name = model.name, suppressWarnings = TRUE,
##     silent = silent, run = run)
##
## 95% confidence intervals: z statistic approximation
## Coefficients:
##               Estimate   Std.Error      lbound      ubound z value
## Intercept1 -0.24259253  0.04093281 -0.32281935 -0.16236571 -5.9266
## Intercept2  0.31920502  0.03782141  0.24507642  0.39333362  8.4398
## Intercept3  0.14298554  0.02053265  0.10274229  0.18322879  6.9638
## Intercept4 -0.30866862  0.04448889 -0.39586524 -0.22147200 -6.9381
```

```
## Intercept5 -0.18052178  0.02565064 -0.23079611 -0.13024745 -7.0377
## Intercept6  0.27907254  0.03861529  0.20338796  0.35475713  7.2270
## Tau2_1_1    0.01919055  0.00861594  0.00230362  0.03607747  2.2273
## Tau2_2_2    0.02538388  0.00886824  0.00800245  0.04276532  2.8623
## Tau2_3_3    0.00670507  0.00281928  0.00117938  0.01223076  2.3783
## Tau2_4_4    0.01240653  0.00747080 -0.00223597  0.02704902  1.6607
## Tau2_5_5    0.00706886  0.00371117 -0.00020489  0.01434261  1.9048
## Tau2_6_6    0.01593875  0.00724252  0.00174368  0.03013382  2.2007
##             Pr(>|z|)
## Intercept1 3.093e-09 ***
## Intercept2 < 2.2e-16 ***
## Intercept3 3.312e-12 ***
## Intercept4 3.974e-12 ***
## Intercept5 1.954e-12 ***
## Intercept6 4.938e-13 ***
## Tau2_1_1    0.025925 *
## Tau2_2_2    0.004205 **
## Tau2_3_3    0.017393 *
## Tau2_4_4    0.096780 .
## Tau2_5_5    0.056812 .
## Tau2_6_6    0.027756 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 921.8621
## Degrees of freedom of the Q statistic: 95
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                              Estimate
## Intercept1: I2 (Q statistic)   0.9202
## Intercept2: I2 (Q statistic)   0.9428
## Intercept3: I2 (Q statistic)   0.7932
## Intercept4: I2 (Q statistic)   0.8777
## Intercept5: I2 (Q statistic)   0.8025
## Intercept6: I2 (Q statistic)   0.9027
##
## Number of studies (or clusters): 45
## Number of observed statistics: 101
## Number of estimated parameters: 12
## Degrees of freedom: 89
## -2 log likelihood: -124.9153
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

Stage 2 random-effects model on all data

- Create a matrix with direct effects (A) and a matrix with variances and covariances (S)

```
varnames <- c("pos","neg","enga","achiev")

A <- create.mxMatrix(c( 0,0,0,0,
                        0,0,0,0,
```

```
                        "0.1*b31","0.1*b32",0,0,
                        0,0,"0.1*b43",0),
                      type = "Full", nrow = 4, ncol = 4, byrow = TRUE,
                      name = "A", as.mxMatrix = FALSE)
dimnames(A) <- list(varnames, varnames)
A
```

```
##        pos       neg       enga      achiev
## pos    "0"       "0"       "0"       "0"
## neg    "0"       "0"       "0"       "0"
## enga   "0.1*b31" "0.1*b32" "0"       "0"
## achiev "0"       "0"       "0.1*b43" "0"
```

```
S <- create.mxMatrix(c(1,
                       ".5*p21",1,
                       0,0,"0.6*p33",
                       0,0,0,"0.6*p44"),
                     type="Symm", byrow = TRUE,
                     name="S", as.mxMatrix = FALSE)
dimnames(S) <- list(varnames, varnames)
S
```

```
##        pos      neg      enga      achiev
## pos    "1"      ".5*p21" "0"       "0"
## neg    ".5*p21" "1"      "0"       "0"
## enga   "0"      "0"      "0.6*p33" "0"
## achiev "0"      "0"      "0"       "0.6*p44"
```

- Fitting the Stage 2 model on the pooled correlation matrix from the random effects Stage 1 analysis

```
# Run the Stage 2 model
Stage2 <- tssem2(stage1random, Amatrix=A, Smatrix=S,
                 mx.algebras = list(Indirect_pos=mxAlgebra(b31*b43, name = "Indirect_pos"),
                                    Indirect_neg=mxAlgebra(b32*b43, name = "Indirect_neg")))
summary(Stage2)
```

```
##
## Call:
## wls(Cov = pooledS, asyCov = asyCov, n = tssem1.obj$total.n, Amatrix = Amatrix,
##     Smatrix = Smatrix, Fmatrix = Fmatrix, diag.constraints = diag.constraints,
##     cor.analysis = cor.analysis, intervals.type = intervals.type,
##     mx.algebras = mx.algebras, model.name = model.name, suppressWarnings = suppressWarnings,
##     silent = silent, run = run)
##
## 95% confidence intervals: z statistic approximation
## Coefficients:
##      Estimate Std.Error    lbound    ubound z value  Pr(>|z|)
## b43  0.348639  0.031005  0.287871  0.409407 11.2447 < 2.2e-16 ***
## b32 -0.298846  0.042218 -0.381592 -0.216101 -7.0787 1.456e-12 ***
## b31  0.274312  0.036934  0.201923  0.346702  7.4271 1.110e-13 ***
## p21 -0.239683  0.040928 -0.319899 -0.159466 -5.8563 4.734e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## mxAlgebras objects:
## Indirect_pos Indirect_neg
```

```
##   0.09563584  -0.10418938
##
## Goodness-of-fit indices:
##                                                Value
## Sample size                                29438.0000
## Chi-square of target model                    11.1613
## DF of target model                             2.0000
## p value of target model                        0.0038
## Number of constraints imposed on "Smatrix"     0.0000
## DF manually adjusted                           0.0000
## Chi-square of independence model             277.0853
## DF of independence model                       6.0000
## RMSEA                                          0.0125
## RMSEA lower 95% CI                             0.0061
## RMSEA upper 95% CI                             0.0200
## SRMR                                           0.0448
## TLI                                            0.8986
## CFI                                            0.9662
## AIC                                            7.1613
## BIC                                           -9.4187
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

Subgroup analysis

- We split the studies based on the % SES

```
# Data for studies with majority low SES
data_low <- Roorda11$data[Roorda11$SES>50]
n_low <- Roorda11$n[Roorda11$SES>50]

# Data for studies with majority high SES
data_high <- Roorda11$data[Roorda11$SES<=50]
n_high <- Roorda11$n[Roorda11$SES<=50]
```

Fitting a random-effects Stage 1 model in both subgroups

```
# Stage 1 analysis per subgroup (random-effects analysis)
stage1_low.fit <- tssem1(my.df = data_low, n = n_low, method = "REM", RE.type = "Diag")
stage1_high.fit <- tssem1(my.df = data_high, n = n_high, method = "REM", RE.type = "Diag")

summary(stage1_low.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(x = paste(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es, sep = "")), RE.lbound = RE.lbound,
##     I2 = I2, model.name = model.name, suppressWarnings = TRUE,
##     silent = silent, run = run)
##
## 95% confidence intervals: z statistic approximation
## Coefficients:
```

```
##              Estimate  Std.Error     lbound     ubound z value  Pr(>|z|)
## Intercept1 -0.3297964  0.0469541 -0.4218247 -0.2377680 -7.0238 2.159e-12
## Intercept2  0.3535758  0.0448531  0.2656654  0.4414862  7.8830 3.109e-15
## Intercept3  0.1204494  0.0266097  0.0682954  0.1726035  4.5265 5.996e-06
## Intercept4 -0.3518625  0.0538236 -0.4573548 -0.2463703 -6.5373 6.263e-11
## Intercept5 -0.1777484  0.0351393 -0.2466202 -0.1088767 -5.0584 4.228e-07
## Intercept6  0.2259883  0.0480624  0.1317877  0.3201890  4.7020 2.577e-06
## Tau2_1_1    0.0111794  0.0069318 -0.0024066  0.0247654  1.6128   0.10679
## Tau2_2_2    0.0269074  0.0107447  0.0058481  0.0479666  2.5042   0.01227
## Tau2_3_3    0.0051245  0.0031918 -0.0011313  0.0113802  1.6055   0.10838
## Tau2_4_4    0.0105147  0.0088957 -0.0069206  0.0279501  1.1820   0.23721
## Tau2_5_5    0.0055861  0.0047332 -0.0036907  0.0148630  1.1802   0.23792
## Tau2_6_6    0.0130146  0.0081686 -0.0029955  0.0290247  1.5933   0.11110
##
## Intercept1 ***
## Intercept2 ***
## Intercept3 ***
## Intercept4 ***
## Intercept5 ***
## Intercept6 ***
## Tau2_1_1
## Tau2_2_2   *
## Tau2_3_3
## Tau2_4_4
## Tau2_5_5
## Tau2_6_6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 578.0407
## Degrees of freedom of the Q statistic: 48
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                               Estimate
## Intercept1: I2 (Q statistic)   0.8546
## Intercept2: I2 (Q statistic)   0.9417
## Intercept3: I2 (Q statistic)   0.7113
## Intercept4: I2 (Q statistic)   0.8389
## Intercept5: I2 (Q statistic)   0.7304
## Intercept6: I2 (Q statistic)   0.8636
##
## Number of studies (or clusters): 24
## Number of observed statistics: 54
## Number of estimated parameters: 12
## Degrees of freedom: 42
## -2 log likelihood: -73.27951
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```
summary(stage1_high.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(x = paste(RE.startvalues,
```

```
##     "*Tau2_", 1:no.es, "_", 1:no.es, sep = "")), RE.lbound = RE.lbound,
##     I2 = I2, model.name = model.name, suppressWarnings = TRUE,
##     silent = silent, run = run)
##
## 95% confidence intervals: z statistic approximation
## Coefficients:
##               Estimate   Std.Error      lbound      ubound z value
## Intercept1 -0.16585168  0.05551542 -0.27465991 -0.05704345 -2.9875
## Intercept2  0.22857479  0.05498427  0.12080759  0.33634198  4.1571
## Intercept3  0.16188262  0.02948168  0.10409959  0.21966564  5.4910
## Intercept4 -0.23410428  0.05327516 -0.33852167 -0.12968688 -4.3942
## Intercept5 -0.18110950  0.03685927 -0.25335235 -0.10886666 -4.9135
## Intercept6  0.33814683  0.05212185  0.23598988  0.44030378  6.4876
## Tau2_1_1    0.01730394  0.01106641 -0.00438583  0.03899371  1.5636
## Tau2_2_2    0.01008418  0.00856552 -0.00670393  0.02687228  1.1773
## Tau2_3_3    0.00719333  0.00417741 -0.00099425  0.01538091  1.7220
## Tau2_4_4    0.00381664  0.00760538 -0.01108963  0.01872292  0.5018
## Tau2_5_5    0.00772284  0.00527512 -0.00261621  0.01806190  1.4640
## Tau2_6_6    0.01253071  0.00900561 -0.00511996  0.03018138  1.3914
##             Pr(>|z|)
## Intercept1  0.002813 **
## Intercept2 3.223e-05 ***
## Intercept3 3.998e-08 ***
## Intercept4 1.112e-05 ***
## Intercept5 8.945e-07 ***
## Intercept6 8.720e-11 ***
## Tau2_1_1    0.117901
## Tau2_2_2    0.239076
## Tau2_3_3    0.085077 .
## Tau2_4_4    0.615784
## Tau2_5_5    0.143191
## Tau2_6_6    0.164094
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 194.1947
## Degrees of freedom of the Q statistic: 41
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                               Estimate
## Intercept1: I2 (Q statistic)   0.8997
## Intercept2: I2 (Q statistic)   0.8439
## Intercept3: I2 (Q statistic)   0.7900
## Intercept4: I2 (Q statistic)   0.6644
## Intercept5: I2 (Q statistic)   0.7996
## Intercept6: I2 (Q statistic)   0.8716
##
## Number of studies (or clusters): 21
## Number of observed statistics: 47
## Number of estimated parameters: 12
## Degrees of freedom: 35
## -2 log likelihood: -65.26855
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
```

```
## Other values may indicate problems.)
```

Fitting the Stage 2 model in both subgroups

```
# Stage 2 analysis per subgroup (random-effect analysis)

stage2_low.fit <- tssem2(stage1_low.fit, Amatrix=A, Smatrix=S)

stage2_high.fit <- tssem2(stage1_high.fit, Amatrix=A, Smatrix=S)
summary(stage2_low.fit)
```

```
##
## Call:
## wls(Cov = pooledS, asyCov = asyCov, n = tssem1.obj$total.n, Amatrix = Amatrix,
##     Smatrix = Smatrix, Fmatrix = Fmatrix, diag.constraints = diag.constraints,
##     cor.analysis = cor.analysis, intervals.type = intervals.type,
##     mx.algebras = mx.algebras, model.name = model.name, suppressWarnings = suppressWarnings,
##     silent = silent, run = run)
##
## 95% confidence intervals: z statistic approximation
## Coefficients:
##      Estimate Std.Error    lbound    ubound z value  Pr(>|z|)
## b43  0.287634  0.038743  0.211698  0.363569  7.4241 1.137e-13 ***
## b32 -0.306873  0.057622 -0.419811 -0.193935 -5.3256 1.006e-07 ***
## b31  0.265198  0.050430  0.166356  0.364039  5.2587 1.451e-07 ***
## p21 -0.329102  0.046956 -0.421135 -0.237069 -7.0087 2.406e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                                Value
## Sample size                                11883.0000
## Chi-square of target model                     6.2786
## DF of target model                             2.0000
## p value of target model                        0.0433
## Number of constraints imposed on "Smatrix"     0.0000
## DF manually adjusted                           0.0000
## Chi-square of independence model             201.6100
## DF of independence model                       6.0000
## RMSEA                                          0.0134
## RMSEA lower 95% CI                             0.0020
## RMSEA upper 95% CI                             0.0259
## SRMR                                           0.0411
## TLI                                            0.9344
## CFI                                            0.9781
## AIC                                            2.2786
## BIC                                          -12.4871
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```
summary(stage2_high.fit)
```

```
##
## Call:
```

```
## wls(Cov = pooledS, asyCov = asyCov, n = tssem1.obj$total.n, Amatrix = Amatrix,
##     Smatrix = Smatrix, Fmatrix = Fmatrix, diag.constraints = diag.constraints,
##     cor.analysis = cor.analysis, intervals.type = intervals.type,
##     mx.algebras = mx.algebras, model.name = model.name, suppressWarnings = suppressWarnings,
##     silent = silent, run = run)
##
## 95% confidence intervals: z statistic approximation
## Coefficients:
##      Estimate Std.Error    lbound    ubound z value  Pr(>|z|)
## b43  0.413663  0.044789  0.325879  0.501447  9.2359 < 2.2e-16 ***
## b32 -0.249265  0.049543 -0.346367 -0.152163 -5.0313 4.871e-07 ***
## b31  0.250022  0.047337  0.157243  0.342802  5.2817 1.280e-07 ***
## p21 -0.159492  0.055493 -0.268256 -0.050728 -2.8741  0.004052 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                                  Value
## Sample size                                  17555.0000
## Chi-square of target model                       9.5002
## DF of target model                               2.0000
## p value of target model                          0.0087
## Number of constraints imposed on "Smatrix"       0.0000
## DF manually adjusted                             0.0000
## Chi-square of independence model               123.4797
## DF of independence model                         6.0000
## RMSEA                                            0.0146
## RMSEA lower 95% CI                               0.0063
## RMSEA upper 95% CI                               0.0245
## SRMR                                             0.0549
## TLI                                              0.8085
## CFI                                              0.9362
## AIC                                              5.5002
## BIC                                            -10.0460
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

Testing the equality of regression coefficients

- We create and fit a model with equal direct effects (we use the same matrix A for both groups),but different variances and covariances, so we create an S matrix with different labels for the 'high' group

```
S_high <- create.mxMatrix(c(1,
                            ".1*p21_high",1,
                             0,0,"0.6*p33_high",
                             0,0,0,"0.6*p44_high"),
                      type="Symm", byrow = TRUE,
                      name="S", as.mxMatrix = FALSE)
S_high
```

```
##      [,1]          [,2]          [,3]           [,4]
## [1,] "1"           ".1*p21_high" "0"            "0"
## [2,] ".1*p21_high" "1"           "0"            "0"
## [3,] "0"           "0"           "0.6*p33_high" "0"
```

```
## [4,] "0"           "0"           "0"           "0.6*p44_high"
```

```
# Create the models for the two groups, make sure to set the argument run=FALSE
stage2_low <- tssem2(stage1_low.fit, Amatrix=A, Smatrix=S, run=FALSE, model.name="low")

stage2_high <- tssem2(stage1_high.fit, Amatrix=A, Smatrix=S_high, run=FALSE, model.name="high")

# Create the multigroup model
stage2_constrained <- mxModel(model="same_regression_coef", stage2_low, stage2_high,
                              mxFitFunctionMultigroup(c("low", "high")))

# Fit multigroup model with equality constraints
Stage2_constrained.fit <- mxRun(stage2_constrained, intervals=TRUE)
```

```
## Running same_regression_coef with 5 parameters
```

```
# first make a list of the fitted models in the separate groups
submodels.fit <- list(stage2_low.fit,stage2_high.fit)

subgroup.summary(submodels.fit,Stage2_constrained.fit)
```

```
## # # # # # # # # # # # # # # # # # # #
##  Output for subgroup MASEM analysis
## # # # # # # # # # # # # # # # # # # #
##
##  Total sample size: 29438
##
##  Parameter estimates of the constrained model
##
## [1] "Set 'print.est=TRUE' to print the parameter estimates of the constrained model"
##
##  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Fit indices of the free model:
##
##           Statistic Free_m1
##                  df   4.000
##         Chi-square  15.779
##                   p   0.003
##               RMSEA   0.014
##  RMSEA lower 95% CI   0.006
##  RMSEA upper 95% CI   0.023
##                 CFI   0.962
##                 TLI   0.887
##                 AIC  47.779
##                 BIC 180.419
##                SRMR   0.050
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Fit indices of the model with equality constraints:
##
##           Statistic Constrained_m2
##                  df          7.000
##         Chi-square         20.959
##                   p          0.004
##               RMSEA          0.012
##  RMSEA lower 95% CI          0.005
##  RMSEA upper 95% CI          0.019
```

```
##                 CFI          0.955
##                 TLI          0.924
##                 AIC         46.959
##                 BIC        154.730
##                SRMR          0.057
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Chi-square difference between free and constrained model:
##
##   Statistic Diff_m1_m2
##          df      3.000
##  Chi-square      5.180
##           p      0.159
##
##  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```