MASEM on Nohe et al. (2015) data

Suzanne Jak and Mike Cheung

May 08, 2019

Contents

TSSEM (with complete data)

A model without any moderator

```
library(metaSEM)

## Proposed model in lavaan syntax
model1 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ w1WITHs1*S1
           W2 ~~ w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ Errw2*W2
           S2 ~~ Errs2*S2'

RAM1 <- lavaan2RAM(model1, obs.variables=c("W1", "S1", "W2", "S2"))
RAM1
```

```
## $A
##    W1      S1      W2  S2
## W1 "0"     "0"     "0" "0"
## S1 "0"     "0"     "0" "0"
```

```
## W2 "0*w2w" "0*s2w" "0" "0"
## S2 "0*w2s" "0*s2s" "0" "0"
##
## $S
##    W1           S1           W2           S2
## W1 "1"          "0*w1WITHs1" "0"          "0"
## S1 "0*w1WITHs1" "1"          "0"          "0"
## W2 "0"          "0"          "0*Errw2"    "0*w2WITHs2"
## S2 "0"          "0"          "0*w2WITHs2" "0*Errs2"
##
## $F
##    W1 S1 W2 S2
## W1  1  0  0  0
## S1  0  1  0  0
## W2  0  0  1  0
## S2  0  0  0  1
##
## $M
##   W1 S1 W2 S2
## 1  0  0  0  0
```

```
## Display the number of data points
pattern.na(Nohe15A1$data, show.na=FALSE)
```

```
##    W1 S1 W2 S2
## W1 32 32 32 32
## S1 32 32 32 32
## W2 32 32 32 32
## S2 32 32 32 32
```

```
## Stage 1 analysis
random1 <- tssem1(Nohe15A1$data, Nohe15A1$n, method="REM", RE.type="Diag")
summary(random1)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(paste0(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es)), 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 0.3804522 0.0225616 0.3362323 0.4246720 16.8629 < 2.2e-16 ***
## Intercept2 0.6051298 0.0180362 0.5697794 0.6404802 33.5508 < 2.2e-16 ***
## Intercept3 0.3032290 0.0178803 0.2681842 0.3382738 16.9588 < 2.2e-16 ***
## Intercept4 0.3036392 0.0178408 0.2686718 0.3386066 17.0194 < 2.2e-16 ***
## Intercept5 0.6166503 0.0166427 0.5840312 0.6492694 37.0523 < 2.2e-16 ***
## Intercept6 0.3954085 0.0216645 0.3529470 0.4378701 18.2515 < 2.2e-16 ***
## Tau2_1_1   0.0134777 0.0038704 0.0058919 0.0210635  3.4823 0.0004972 ***
## Tau2_2_2   0.0087592 0.0025260 0.0038083 0.0137102  3.4676 0.0005252 ***
## Tau2_3_3   0.0071123 0.0022470 0.0027082 0.0115163  3.1652 0.0015496 **
## Tau2_4_4   0.0070585 0.0022121 0.0027229 0.0113941  3.1909 0.0014183 **
## Tau2_5_5   0.0072634 0.0021092 0.0031293 0.0113974  3.4436 0.0005740 ***
## Tau2_6_6   0.0122813 0.0034848 0.0054513 0.0191114  3.5243 0.0004246 ***
```

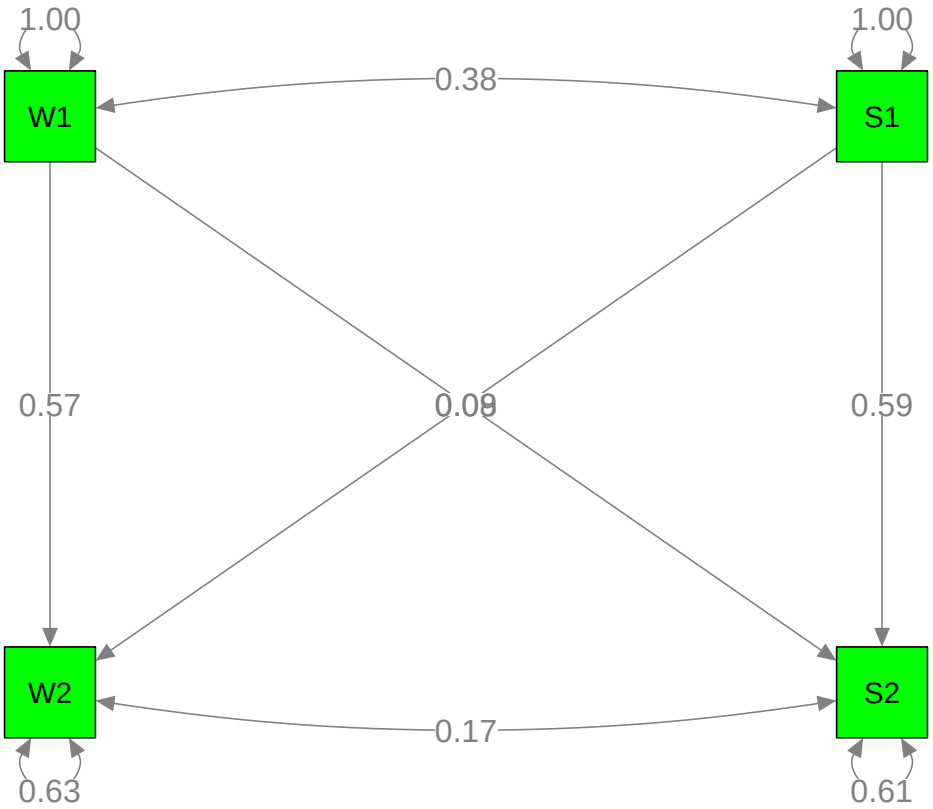
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 1466.159
## Degrees of freedom of the Q statistic: 186
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                               Estimate
## Intercept1: I2 (Q statistic)   0.8829
## Intercept2: I2 (Q statistic)   0.8973
## Intercept3: I2 (Q statistic)   0.7743
## Intercept4: I2 (Q statistic)   0.7718
## Intercept5: I2 (Q statistic)   0.8810
## Intercept6: I2 (Q statistic)   0.8748
##
## Number of studies (or clusters): 32
## Number of observed statistics: 192
## Number of estimated parameters: 12
## Degrees of freedom: 180
## -2 log likelihood: -300.1701
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```
## Stage 2 analysis
random2 <- tssem2(random1, Amatrix=RAM1$A, Smatrix=RAM1$S)
summary(random2)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s      0.586124  0.020790 0.545376 0.626872 28.1926 < 2.2e-16 ***
## w2s      0.080237  0.024842 0.031547 0.128927  3.2299 0.0012385 **
## s2w      0.085841  0.024796 0.037242 0.134440  3.4619 0.0005364 ***
## w2w      0.572471  0.022265 0.528834 0.616109 25.7122 < 2.2e-16 ***
## w1WITHs1 0.380452  0.022562 0.336232 0.424672 16.8629 < 2.2e-16 ***
## w2WITHs2 0.168885  0.025232 0.119431 0.218338  6.6933 2.182e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                           Value
## Sample size                               12906
## Chi-square of target model                    0
## DF of target model                            0
## p value of target model                       0
## Number of constraints imposed on "Smatrix"    0
```

```
## DF manually adjusted                          0
## Chi-square of independence model           3079
## DF of independence model                      6
## RMSEA                                         0
## RMSEA lower 95% CI                            0
## RMSEA upper 95% CI                            0
## SRMR                                          0
## TLI                                        -Inf
## CFI                                           1
## AIC                                           0
## BIC                                           0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```
## Plot the model
plot(random2, col="green")
```



Models with three subgroup analysis

```
## Get the necessary functions
source("http://www.suzannejak.nl/subgroup.functions.R")

data <- Nohe15A1$data
n <- Nohe15A1$n
Lag <- Nohe15A1$Lag

# Data for studies with short Lag
```

```r
data_g1 <- data[Lag<7]
n_g1 <- n[Lag<7]

# Data for studies with medium Lag
data_g2 <- data[Lag>=7&Lag<13]
n_g2 <- n[Lag>=7&Lag<13]

# Data for studies with long Lag
data_g3 <- data[Lag>=13]
n_g3 <- n[Lag>=13]
```

Fitting a random-effects Stage 1 model in three subgroups

```r
## Stage 1 analysis per subgroup (random-effects analysis)
stage1_g1.fit <- tssem1(Cov = data_g1, n = n_g1, method = "REM", RE.type = "Diag")
stage1_g2.fit <- tssem1(Cov = data_g2, n = n_g2, method = "REM", RE.type = "Diag")
stage1_g3.fit <- tssem1(Cov = data_g3, n = n_g3, method = "REM", RE.type = "Diag")

## Rerun it to remove the error code
stage1_g3.fit <- rerun(stage1_g3.fit)
```

```r
## Results
summary(stage1_g1.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(paste0(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es)), 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  0.42860576  0.03711507  0.35586157  0.50134995 11.5480  < 2e-16 ***
## Intercept2  0.64893778  0.02491946  0.60009654  0.69777902 26.0414  < 2e-16 ***
## Intercept3  0.34725162  0.03476027  0.27912275  0.41538049  9.9899  < 2e-16 ***
## Intercept4  0.35445085  0.03474154  0.28635868  0.42254303 10.2025  < 2e-16 ***
## Intercept5  0.69211029  0.02527053  0.64258095  0.74163963 27.3880  < 2e-16 ***
## Intercept6  0.42483613  0.04413725  0.33832870  0.51134355  9.6253  < 2e-16 ***
## Tau2_1_1    0.01057581  0.00550629 -0.00021631  0.02136794  1.9207  0.05477 .
## Tau2_2_2    0.00454042  0.00259602 -0.00054769  0.00962853  1.7490  0.08029 .
## Tau2_3_3    0.00841150  0.00460648 -0.00061704  0.01744003  1.8260  0.06785 .
## Tau2_4_4    0.00843305  0.00464811 -0.00067708  0.01754318  1.8143  0.06963 .
## Tau2_5_5    0.00502724  0.00281689 -0.00049375  0.01054824  1.7847  0.07431 .
## Tau2_6_6    0.01606559  0.00804791  0.00029199  0.03183920  1.9962  0.04591 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 341.1965
## Degrees of freedom of the Q statistic: 54
## P value of the Q statistic: 0
##
```

```
## Heterogeneity indices (based on the estimated Tau2):
##                              Estimate
## Intercept1: I2 (Q statistic)   0.8161
## Intercept2: I2 (Q statistic)   0.7993
## Intercept3: I2 (Q statistic)   0.7506
## Intercept4: I2 (Q statistic)   0.7546
## Intercept5: I2 (Q statistic)   0.8423
## Intercept6: I2 (Q statistic)   0.8662
##
## Number of studies (or clusters): 10
## Number of observed statistics: 60
## Number of estimated parameters: 12
## Degrees of freedom: 48
## -2 log likelihood: -97.84918
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```
summary(stage1_g2.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(paste0(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es)), 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 0.34829723 0.03716906 0.27544722 0.42114724  9.3706  < 2e-16 ***
## Intercept2 0.61072347 0.02048942 0.57056494 0.65088199 29.8068  < 2e-16 ***
## Intercept3 0.28588838 0.02572111 0.23547593 0.33630084 11.1149  < 2e-16 ***
## Intercept4 0.28841182 0.02673802 0.23600626 0.34081737 10.7866  < 2e-16 ***
## Intercept5 0.58850378 0.02457855 0.54033071 0.63667684 23.9438  < 2e-16 ***
## Intercept6 0.36861766 0.02911075 0.31156164 0.42567368 12.6626  < 2e-16 ***
## Tau2_1_1   0.01809559 0.00728838 0.00381062 0.03238055  2.4828  0.01304 *
## Tau2_2_2   0.00485932 0.00231300 0.00032593 0.00939272  2.1009  0.03565 *
## Tau2_3_3   0.00705198 0.00311184 0.00095289 0.01315107  2.2662  0.02344 *
## Tau2_4_4   0.00784269 0.00345167 0.00107755 0.01460784  2.2721  0.02308 *
## Tau2_5_5   0.00753169 0.00312393 0.00140889 0.01365449  2.4110  0.01591 *
## Tau2_6_6   0.01018393 0.00414085 0.00206802 0.01829984  2.4594  0.01392 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 714.1101
## Degrees of freedom of the Q statistic: 84
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                              Estimate
## Intercept1: I2 (Q statistic)   0.9053
## Intercept2: I2 (Q statistic)   0.8238
## Intercept3: I2 (Q statistic)   0.7673
## Intercept4: I2 (Q statistic)   0.7853
## Intercept5: I2 (Q statistic)   0.8730
```

```
## Intercept6: I2 (Q statistic)   0.8455
##
## Number of studies (or clusters): 15
## Number of observed statistics: 90
## Number of estimated parameters: 12
## Degrees of freedom: 78
## -2 log likelihood: -150.9746
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```
summary(stage1_g3.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(paste0(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es)), 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  3.9659e-01  4.0660e-02  3.1690e-01  4.7628e-01  9.7538 < 2.2e-16 ***
## Intercept2  5.3658e-01  4.8889e-02  4.4076e-01  6.3240e-01 10.9754 < 2.2e-16 ***
## Intercept3  2.9657e-01  2.9449e-02  2.3885e-01  3.5428e-01 10.0706 < 2.2e-16 ***
## Intercept4  2.7917e-01  6.7692e-02  1.4650e-01  4.1185e-01  4.1242  3.72e-05 ***
## Intercept5  5.8098e-01  2.7737e-02  5.2662e-01  6.3535e-01 20.9464 < 2.2e-16 ***
## Intercept6  4.2588e-01  4.8654e-02  3.3052e-01  5.2124e-01  8.7532 < 2.2e-16 ***
## Tau2_1_1    3.2548e-03  3.4114e-03 -3.4314e-03  9.9409e-03  0.9541    0.3400
## Tau2_2_2    1.3288e-02  8.7629e-03 -3.8868e-03  3.0463e-02  1.5164    0.1294
## Tau2_3_3    1.1556e-03  2.7021e-03 -4.1404e-03  6.4516e-03  0.4277    0.6689
## Tau2_4_4    1.0000e-10  4.6898e-03 -9.1919e-03  9.1919e-03  0.0000    1.0000
## Tau2_5_5    5.5237e-04  1.7507e-03 -2.8790e-03  3.9838e-03  0.3155    0.7524
## Tau2_6_6    8.1111e-03  5.6552e-03 -2.9728e-03  1.9195e-02  1.4343    0.1515
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 254.7399
## Degrees of freedom of the Q statistic: 36
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                              Estimate
## Intercept1: I2 (Q statistic)   0.6753
## Intercept2: I2 (Q statistic)   0.9303
## Intercept3: I2 (Q statistic)   0.3817
## Intercept4: I2 (Q statistic)   0.0000
## Intercept5: I2 (Q statistic)   0.3672
## Intercept6: I2 (Q statistic)   0.8458
##
## Number of studies (or clusters): 7
## Number of observed statistics: 42
## Number of estimated parameters: 12
## Degrees of freedom: 30
## -2 log likelihood: -91.83662
```

```
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

Fitting the Stage 2 model in three subgroups

```
## Stage 2 analysis per subgroup (random-effect analysis)
stage2_g1.fit <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)
stage2_g2.fit <- tssem2(stage1_g2.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)
stage2_g3.fit <- tssem2(stage1_g3.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)

## Results
summary(stage2_g1.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s       0.66553740  0.03569841  0.59556980  0.73550500 18.6433 < 2.2e-16 ***
## w2s       0.06199846  0.04876834 -0.03358573  0.15758265  1.2713  0.203627
## s2w       0.09348604  0.04754748  0.00029469  0.18667738  1.9662  0.049280 *
## w2w       0.60886913  0.03478416  0.54069342  0.67704484 17.5042 < 2.2e-16 ***
## w1WITHs1  0.42860576  0.03711507  0.35586157  0.50134995 11.5480 < 2.2e-16 ***
## w2WITHs2  0.14870269  0.05135547  0.04804782  0.24935755  2.8956  0.003785 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                            Value
## Sample size                               2845.0
## Chi-square of target model                   0.0
## DF of target model                           0.0
## p value of target model                      0.0
## Number of constraints imposed on "Smatrix"   0.0
## DF manually adjusted                         0.0
## Chi-square of independence model          1561.7
## DF of independence model                     6.0
## RMSEA                                        0.0
## RMSEA lower 95% CI                           0.0
## RMSEA upper 95% CI                           0.0
## SRMR                                         0.0
## TLI                                         -Inf
## CFI                                          1.0
## AIC                                          0.0
## BIC                                          0.0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```r
summary(stage2_g2.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s      0.556431  0.029435 0.498739 0.614123 18.9036 < 2.2e-16 ***
## w2s      0.092085  0.035876 0.021770 0.162400  2.5668   0.01026 *
## s2w      0.086149  0.036990 0.013651 0.158648  2.3290   0.01986 *
## w2w      0.580718  0.025338 0.531056 0.630379 22.9189 < 2.2e-16 ***
## w1WITHs1 0.348297  0.037169 0.275447 0.421147  9.3706 < 2.2e-16 ***
## w2WITHs2 0.151898  0.035147 0.083011 0.220785  4.3218 1.548e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                            Value
## Sample size                               5991.0
## Chi-square of target model                   0.0
## DF of target model                           0.0
## p value of target model                      0.0
## Number of constraints imposed on "Smatrix"   0.0
## DF manually adjusted                         0.0
## Chi-square of independence model          1662.6
## DF of independence model                     6.0
## RMSEA                                        0.0
## RMSEA lower 95% CI                           0.0
## RMSEA upper 95% CI                           0.0
## SRMR                                         0.0
## TLI                                         -Inf
## CFI                                          1.0
## AIC                                          0.0
## BIC                                          0.0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```r
summary(stage2_g3.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s       0.549852  0.030205  0.490651  0.609053 18.2040 < 2.2e-16 ***
## w2s       0.078499  0.028028  0.023565  0.133433  2.8007  0.005099 **
## s2w       0.078761  0.061399 -0.041579  0.199102  1.2828  0.199571
## w2w       0.505340  0.055375  0.396807  0.613872  9.1258 < 2.2e-16 ***
## w1WITHs1  0.396591  0.040660  0.316899  0.476284  9.7538 < 2.2e-16 ***
## w2WITHs2  0.230256  0.039857  0.152138  0.308374  5.7771   7.6e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                                Value
## Sample size                                   4070.0
## Chi-square of target model                       0.0
## DF of target model                               0.0
## p value of target model                          0.0
## Number of constraints imposed on "Smatrix"       0.0
## DF manually adjusted                             0.0
## Chi-square of independence model              1319.6
## DF of independence model                         6.0
## RMSEA                                            0.0
## RMSEA lower 95% CI                               0.0
## RMSEA upper 95% CI                               0.0
## SRMR                                             0.0
## TLI                                             -Inf
## CFI                                              1.0
## AIC                                              0.0
## BIC                                              0.0
## 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 group 2 and group 3.

```
## Proposed model g2
model2 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ g2w1WITHs1*S1
           W2 ~~ g2w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g2Errw2*W2
           S2 ~~ g2Errs2*S2'

RAM2 <- lavaan2RAM(model2, obs.variables=c("W1", "S1", "W2", "S2"))

## Proposed model g3
model3 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ g3w1WITHs1*S1
```

```
          W2 ~~ g3w2WITHs2*S2
          W1 ~~ 1*W1
          S1 ~~ 1*S1
          W2 ~~ g3Errw2*W2
          S2 ~~ g3Errs2*S2'

RAM3 <- lavaan2RAM(model3, obs.variables=c("W1", "S1", "W2", "S2"))

## Create the models for the two groups, make sure to set the argument run=FALSE
stage2_g1 <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S, run=FALSE, model.name="g1")

stage2_g2 <- tssem2(stage1_g2.fit, Amatrix=RAM1$A, Smatrix=RAM2$S, run=FALSE, model.name="g2")

stage2_g3 <- tssem2(stage1_g3.fit, Amatrix=RAM1$A, Smatrix=RAM3$S, run=FALSE, model.name="g3")

## Create the multigroup model
stage2_constrained <- mxModel(model="same_regression_coef", stage2_g1, stage2_g2,stage2_g3,
                              mxFitFunctionMultigroup(c("g1", "g2", "g3")))

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

## first make a list of the fitted models in the separate groups
submodels.fit <- list(stage2_g1.fit, stage2_g2.fit, stage2_g3.fit)

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

```
## # # # # # # # # # # # # # # # # # #
##  Output for subgroup MASEM analysis
## # # # # # # # # # # # # # # # # # #
##
##  Total sample size: 12906
##
##  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   0.000
##         Chi-square   0.000
##                   p   0.000
##               RMSEA     Inf
##  RMSEA lower 95% CI     Inf
##  RMSEA upper 95% CI     Inf
##                 CFI   1.000
##                 TLI    -Inf
##                 AIC  60.000
##                 BIC 283.963
##                SRMR   0.000
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Fit indices of the model with equality constraints:
```

```
##
##           Statistic Constrained_m2
##                  df          8.000
##          Chi-square         14.738
##                   p          0.064
##               RMSEA          0.014
##  RMSEA lower 95% CI          0.000
##  RMSEA upper 95% CI          0.027
##                 CFI          0.999
##                 TLI          0.997
##                 AIC         58.738
##                 BIC        222.978
##                SRMR          0.033
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Chi-square difference between free and constrained model:
##
##   Statistic Diff_m1_m2
##          df      8.000
##  Chi-square     14.738
##           p      0.064
##
##  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

Testing the equality of one regression coefficient (w2w)

- 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 group 2 and group 3.

```
## Proposed model g2
model2 <- 'W2 ~ w2w*W1 + g2s2w*S1
           S2 ~ g2w2s*W1 + g2s2s*S1
           W1 ~~ g2w1WITHs1*S1
           W2 ~~ g2w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g2Errw2*W2
           S2 ~~ g2Errs2*S2'

RAM2 <- lavaan2RAM(model2, obs.variables=c("W1", "S1", "W2", "S2"))

## Proposed model g3
model3 <- 'W2 ~ w2w*W1 + g3s2w*S1
           S2 ~ g3w2s*W1 + g3s2s*S1
           W1 ~~ g3w1WITHs1*S1
           W2 ~~ g3w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g3Errw2*W2
           S2 ~~ g3Errs2*S2'

RAM3 <- lavaan2RAM(model3, obs.variables=c("W1", "S1", "W2", "S2"))

## Create the models for the two groups, make sure to set the argument run=FALSE
```

```r
stage2_g1 <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S, run=FALSE, model.name="g1")

stage2_g2 <- tssem2(stage1_g2.fit, Amatrix=RAM2$A, Smatrix=RAM2$S, run=FALSE, model.name="g2")

stage2_g3 <- tssem2(stage1_g3.fit, Amatrix=RAM3$A, Smatrix=RAM3$S, run=FALSE, model.name="g3")

## Create the multigroup model
stage2_constrained <- mxModel(model="same_regression_coef", stage2_g1, stage2_g2,stage2_g3,
                              mxFitFunctionMultigroup(c("g1", "g2", "g3")))

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

## First make a list of the fitted models in the separate groups
submodels.fit <- list(stage2_g1.fit,stage2_g2.fit,stage2_g3.fit)

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

```
## # # # # # # # # # # # # # # # # # # #
##  Output for subgroup MASEM analysis
## # # # # # # # # # # # # # # # # # # #
##
##  Total sample size: 12906
##
##  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   0.000
##         Chi-square   0.000
##                   p   0.000
##               RMSEA     Inf
##  RMSEA lower 95% CI     Inf
##  RMSEA upper 95% CI     Inf
##                 CFI   1.000
##                 TLI    -Inf
##                 AIC  60.000
##                 BIC 283.963
##                SRMR   0.000
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Fit indices of the model with equality constraints:
##
##           Statistic Constrained_m2
##                  df          2.000
##         Chi-square          2.527
##                   p          0.283
##               RMSEA          0.008
##  RMSEA lower 95% CI          0.000
##  RMSEA upper 95% CI          0.036
##                 CFI          1.000
```

```
##                 TLI          0.999
##                 AIC         58.527
##                 BIC        267.560
##                SRMR          0.015
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Chi-square difference between free and constrained model:
##
##   Statistic Diff_m1_m2
##          df      2.000
##  Chi-square      2.527
##           p      0.283
##
##  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

Models with two subgroup analysis

```
# Data for studies with short Lag
data_g1 <- data[Lag<12]
n_g1 <- n[Lag<12]

# Data for studies with long Lag
data_g2 <- data[Lag>=12]
n_g2 <- n[Lag>=12]
```

Fitting a random-effects Stage 1 model in two subgroups

```
## Stage 1 analysis per subgroup (random-effects analysis)
stage1_g1.fit <- tssem1(Cov = data_g1, n = n_g1, method = "REM", RE.type = "Diag")
stage1_g2.fit <- tssem1(Cov = data_g2, n = n_g2, method = "REM", RE.type = "Diag")
## Rerun the analysis
stage1_g2.fit <- rerun(stage1_g2.fit)
```

```
summary(stage1_g1.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(paste0(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es)), 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  4.6331e-01  3.4802e-02  3.9510e-01  5.3152e-01 13.3127  < 2e-16 ***
## Intercept2  6.5871e-01  1.8886e-02  6.2169e-01  6.9572e-01 34.8783  < 2e-16 ***
## Intercept3  3.6924e-01  3.0810e-02  3.0886e-01  4.2963e-01 11.9846  < 2e-16 ***
## Intercept4  3.7718e-01  3.1986e-02  3.1449e-01  4.3987e-01 11.7923  < 2e-16 ***
## Intercept5  6.7273e-01  2.7748e-02  6.1834e-01  7.2711e-01 24.2446  < 2e-16 ***
## Intercept6  4.4613e-01  3.9848e-02  3.6803e-01  5.2423e-01 11.1959  < 2e-16 ***
## Tau2_1_1    1.3297e-02  5.7560e-03  2.0154e-03  2.4578e-02  2.3101  0.02088 *
## Tau2_2_2    3.3131e-03  1.7387e-03 -9.4594e-05  6.7208e-03  1.9056  0.05671 .
```

```
## Tau2_3_3    9.3795e-03  4.2626e-03  1.0249e-03  1.7734e-02  2.2004  0.02778 *
## Tau2_4_4    1.0324e-02  4.7309e-03  1.0512e-03  1.9596e-02  2.1822  0.02910 *
## Tau2_5_5    8.7112e-03  3.8165e-03  1.2310e-03  1.6191e-02  2.2825  0.02246 *
## Tau2_6_6    1.8025e-02  7.6104e-03  3.1090e-03  3.2941e-02  2.3685  0.01786 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 734.0562
## Degrees of freedom of the Q statistic: 72
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                              Estimate
## Intercept1: I2 (Q statistic)   0.8943
## Intercept2: I2 (Q statistic)   0.7986
## Intercept3: I2 (Q statistic)   0.8257
## Intercept4: I2 (Q statistic)   0.8407
## Intercept5: I2 (Q statistic)   0.9166
## Intercept6: I2 (Q statistic)   0.9149
##
## Number of studies (or clusters): 13
## Number of observed statistics: 78
## Number of estimated parameters: 12
## Degrees of freedom: 66
## -2 log likelihood: -122.837
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```
summary(stage1_g2.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(paste0(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es)), 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  0.32573374  0.02286304  0.28092302  0.37054447 14.2472 < 2.2e-16 ***
## Intercept2  0.57069916  0.02385056  0.52395292  0.61744540 23.9281 < 2.2e-16 ***
## Intercept3  0.26146828  0.01444459  0.23315742  0.28977915 18.1015 < 2.2e-16 ***
## Intercept4  0.25259612  0.01201123  0.22905454  0.27613769 21.0300 < 2.2e-16 ***
## Intercept5  0.57722170  0.01668253  0.54452454  0.60991885 34.6004 < 2.2e-16 ***
## Intercept6  0.36349460  0.02093829  0.32245630  0.40453290 17.3603 < 2.2e-16 ***
## Tau2_1_1    0.00721522  0.00304222  0.00125258  0.01317787  2.3717  0.017707 *
## Tau2_2_2    0.00903672  0.00346802  0.00223953  0.01583392  2.6057  0.009168 **
## Tau2_3_3    0.00125766  0.00110733 -0.00091267  0.00342798  1.1358  0.256059
## Tau2_4_4    0.00032457  0.00050817 -0.00067143  0.00132057  0.6387  0.523016
## Tau2_5_5    0.00361660  0.00161473  0.00045179  0.00678141  2.2398  0.025107 *
## Tau2_6_6    0.00576799  0.00241771  0.00102936  0.01050661  2.3857  0.017046 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Q statistic on the homogeneity of effect sizes: 580.9571
## Degrees of freedom of the Q statistic: 108
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                              Estimate
## Intercept1: I2 (Q statistic)   0.7859
## Intercept2: I2 (Q statistic)   0.8876
## Intercept3: I2 (Q statistic)   0.3641
## Intercept4: I2 (Q statistic)   0.1274
## Intercept5: I2 (Q statistic)   0.7614
## Intercept6: I2 (Q statistic)   0.7548
##
## Number of studies (or clusters): 19
## Number of observed statistics: 114
## Number of estimated parameters: 12
## Degrees of freedom: 102
## -2 log likelihood: -240.0904
## 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_g1.fit <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)
stage2_g2.fit <- tssem2(stage1_g2.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)

summary(stage2_g1.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s       0.6387695  0.0397122  0.5609350  0.7166039 16.0850 < 2.2e-16 ***
## w2s       0.0732976  0.0472910 -0.0193911  0.1659863  1.5499 0.1211592
## s2w       0.0916772  0.0464278  0.0006804  0.1826740  1.9746 0.0483114 *
## w2w       0.6162324  0.0302086  0.5570246  0.6754403 20.3992 < 2.2e-16 ***
## w1WITHs1  0.4633081  0.0348019  0.3950977  0.5315186 13.3127 < 2.2e-16 ***
## w2WITHs2  0.1569152  0.0475635  0.0636924  0.2501379  3.2991 0.0009701 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                           Value
## Sample size                              4863.0
## Chi-square of target model                  0.0
```

```
## DF of target model                              0.0
## p value of target model                         0.0
## Number of constraints imposed on "Smatrix"      0.0
## DF manually adjusted                            0.0
## Chi-square of independence model             2025.5
## DF of independence model                        6.0
## RMSEA                                           0.0
## RMSEA lower 95% CI                              0.0
## RMSEA upper 95% CI                              0.0
## SRMR                                            0.0
## TLI                                            -Inf
## CFI                                             1.0
## AIC                                             0.0
## BIC                                             0.0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```
summary(stage2_g2.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s      0.550458  0.018924 0.513368 0.587547 29.0885 < 2.2e-16 ***
## w2s      0.082166  0.019219 0.044498 0.119834  4.2753 1.909e-05 ***
## s2w      0.074617  0.018300 0.038749 0.110485  4.0774 4.554e-05 ***
## w2w      0.546394  0.026741 0.493983 0.598805 20.4330 < 2.2e-16 ***
## w1WITHs1 0.325734  0.022863 0.280923 0.370544 14.2472 < 2.2e-16 ***
## w2WITHs2 0.177559  0.021402 0.135612 0.219506  8.2964 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                              Value
## Sample size                                 8043.0
## Chi-square of target model                     0.0
## DF of target model                             0.0
## p value of target model                        0.0
## Number of constraints imposed on "Smatrix"     0.0
## DF manually adjusted                           0.0
## Chi-square of independence model            2241.6
## DF of independence model                       6.0
## RMSEA                                          0.0
## RMSEA lower 95% CI                             0.0
## RMSEA upper 95% CI                             0.0
## SRMR                                           0.0
## TLI                                           -Inf
## CFI                                            1.0
```

```
## AIC                                                 0.0
## BIC                                                 0.0
## 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 group 2.

```
## Proposed model g2
model2 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ g2w1WITHs1*S1
           W2 ~~ g2w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g2Errw2*W2
           S2 ~~ g2Errs2*S2'

RAM2 <- lavaan2RAM(model2, obs.variables=c("W1", "S1", "W2", "S2"))

# Create the models for the two groups, make sure to set the argument run=FALSE
stage2_g1 <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S, run=FALSE, model.name="g1")

stage2_g2 <- tssem2(stage1_g2.fit, Amatrix=RAM1$A, Smatrix=RAM2$S, run=FALSE, model.name="g2")

# Create the multigroup model
stage2_constrained <- mxModel(model="same_regression_coef", stage2_g1, stage2_g2,
                              mxFitFunctionMultigroup(c("g1", "g2")))

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

# first make a list of the fitted models in the separate groups
submodels.fit <- list(stage2_g1.fit,stage2_g2.fit)

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

```
## # # # # # # # # # # # # # # # # # # #
##  Output for subgroup MASEM analysis
## # # # # # # # # # # # # # # # # # # #
##
##  Total sample size: 12906
##
##  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   0.000
```

```
##          Chi-square   0.000
##                   p   0.000
##               RMSEA     Inf
##  RMSEA lower 95% CI     Inf
##  RMSEA upper 95% CI     Inf
##                 CFI   1.000
##                 TLI    -Inf
##                 AIC  40.000
##                 BIC 189.309
##                SRMR   0.000
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Fit indices of the model with equality constraints:
##
##           Statistic Constrained_m2
##                  df          4.000
##          Chi-square         13.247
##                   p          0.010
##               RMSEA          0.019
##  RMSEA lower 95% CI          0.006
##  RMSEA upper 95% CI          0.033
##                 CFI          0.998
##                 TLI          0.993
##                 AIC         45.247
##                 BIC        164.694
##                SRMR          0.025
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Chi-square difference between free and constrained model:
##
##   Statistic Diff_m1_m2
##          df      4.000
##  Chi-square     13.247
##           p      0.010
##
##  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

OSMASEM (with complete data)

Data preparation

```
## Get the data
data <- Nohe15A1$data
n <- Nohe15A1$n
Lag <- Nohe15A1$Lag

## Calculate the sampling covariance matrix of the correlations
my.df <- Cor2DataFrame(data, n, acov = "weighted")

## Add standardized Lag as a moderator.
## Standardization of the moderator improves the convergence.
my.df$data <- data.frame(my.df$data, Lag=scale(Nohe15A1$Lag),
                         check.names=FALSE)
head(my.df$data)
```

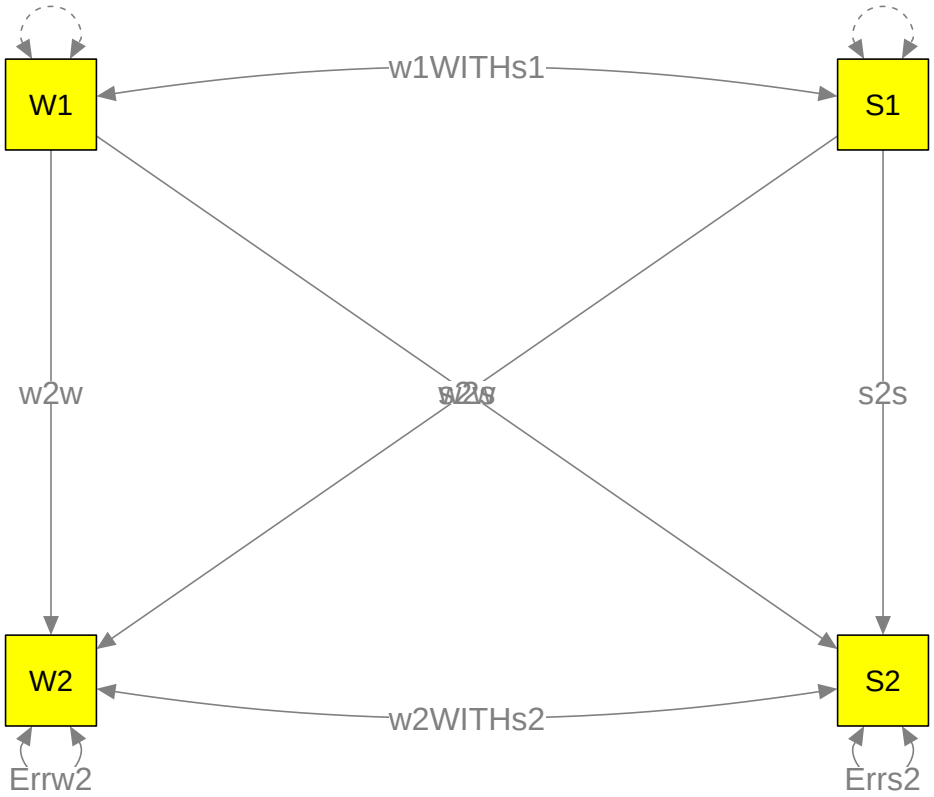
```
##                                       S1_W1 W2_W1 S2_W1 W2_S1 S2_S1 S2_W2 C(S1_W1 S1_W1)
## Britt...Dawson..2005.                  0.29  0.58  0.22  0.24  0.57  0.27   0.0014224800
## Demerouti.et.al...2004.                0.53  0.57  0.41  0.41  0.68  0.54   0.0020763943
## Ford..2010.                            0.35  0.75  0.32  0.26  0.74  0.30   0.0021207097
## Hammer.et.al...2005...female.subsample 0.32  0.57  0.22  0.30  0.43  0.30   0.0029726174
## Hammer.et.al...2005...male.subsample   0.19  0.54  0.17  0.21  0.60  0.30   0.0029726196
## Innstrand.et.al...2008.                0.42  0.63  0.31  0.30  0.62  0.44   0.0003112271
##                                       C(W2_W1 S1_W1) C(S2_W1 S1_W1) C(W2_S1 S1_W1) C(S2_S1 S1_W1)
## Britt...Dawson..2005.                   2.033647e-04   0.0008791262   0.0008736627   2.047359e-04
## Demerouti.et.al...2004.                 2.968495e-04   0.0012832584   0.0012752839   2.988512e-04
## Ford..2010.                             3.031867e-04   0.0013106479   0.0013025035   3.052310e-04
## Hammer.et.al...2005...female.subsample  4.249786e-04   0.0018371458   0.0018257284   4.278432e-04
## Hammer.et.al...2005...male.subsample    4.249785e-04   0.0018371480   0.0018257314   4.278449e-04
## Innstrand.et.al...2008.                 4.449445e-05   0.0001923455   0.0001911501   4.479433e-05
##                                       C(S2_W2 S1_W1) C(W2_W1 W2_W1) C(S2_W1 W2_W1) C(W2_S1 W2_W1)
## Britt...Dawson..2005.                   0.0004890910   0.0007981111   3.750406e-04   3.671032e-04
## Demerouti.et.al...2004.                 0.0007139237   0.0011650031   5.474443e-04   5.358586e-04
## Ford..2010.                             0.0007291625   0.0011898670   5.591302e-04   5.472970e-04
## Hammer.et.al...2005...female.subsample  0.0010220715   0.0016678467   7.837363e-04   7.671493e-04
## Hammer.et.al...2005...male.subsample    0.0010220747   0.0016678468   7.837364e-04   7.671497e-04
## Innstrand.et.al...2008.                 0.0001070089   0.0001746203   8.205576e-05   8.031912e-05
##                                       C(S2_S1 W2_W1) C(S2_W2 W2_W1) C(S2_W1 S2_W1) C(W2_S1 S2_W1)
## Britt...Dawson..2005.                   1.042820e-04   2.035394e-04   0.0016496839   0.0005769124
## Demerouti.et.al...2004.                 1.522189e-04   2.971039e-04   0.0024080415   0.0008421151
## Ford..2010.                             1.554689e-04   3.034468e-04   0.0024594362   0.0008600915
## Hammer.et.al...2005...female.subsample  2.179207e-04   4.253423e-04   0.0034474117   0.0012055943
## Hammer.et.al...2005...male.subsample    2.179212e-04   4.253429e-04   0.0034474147   0.0012055978
## Innstrand.et.al...2008.                 2.281594e-05   4.453263e-05   0.0003609372   0.0001262235
##                                       C(S2_S1 S2_W1) C(S2_W2 S2_W1) C(W2_S1 W2_S1) C(S2_S1 W2_S1)
## Britt...Dawson..2005.                   3.592792e-04   0.0008607809   0.0016601427   3.727429e-04
## Demerouti.et.al...2004.                 5.244373e-04   0.0012564787   0.0024233093   5.440908e-04
## Ford..2010.                             5.356320e-04   0.0012832975   0.0024750297   5.557048e-04
## Hammer.et.al...2005...female.subsample  7.507976e-04   0.0017988063   0.0034692678   7.789334e-04
## Hammer.et.al...2005...male.subsample    7.508000e-04   0.0017988099   0.0034692719   7.789361e-04
## Innstrand.et.al...2008.                 7.860707e-05   0.0001883316   0.0003632255   8.155286e-05
##                                       C(S2_W2 W2_S1) C(S2_S1 S2_S1) C(S2_W2 S2_S1) C(S2_W2 S2_W2)
## Britt...Dawson..2005.                   0.0008739044   0.0007805647   1.951881e-04   0.0013981165
## Demerouti.et.al...2004.                 0.0012756357   0.0011393902   2.849136e-04   0.0020408295
## Ford..2010.                             0.0013028629   0.0011637074   2.909961e-04   0.0020843866
## Hammer.et.al...2005...female.subsample  0.0018262311   0.0016311783   4.078888e-04   0.0029217006
## Hammer.et.al...2005...male.subsample    0.0018262351   0.0016311796   4.078917e-04   0.0029217041
## Innstrand.et.al...2008.                 0.0001912029   0.0001707811   4.270526e-05   0.0003058964
##                                              Lag
## Britt...Dawson..2005.                  -0.6794521
## Demerouti.et.al...2004.                -0.7711151
## Ford..2010.                            -0.8016694
## Hammer.et.al...2005...female.subsample -0.1294740
## Hammer.et.al...2005...male.subsample   -0.1294740
## Innstrand.et.al...2008.                 0.6038301
```

```
## Check the number of studies
pattern.na(Nohe15A1$data, show.na = FALSE)
```

```
##    W1 S1 W2 S2
## W1 32 32 32 32
## S1 32 32 32 32
## W2 32 32 32 32
## S2 32 32 32 32
```

```
## Proposed model
model1 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ w1WITHs1*S1
           W2 ~~ w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ Errw2*W2
           S2 ~~ Errs2*S2'

plot(model1, col="yellow")
```




```
## Convert the lavaan syntax into the RAM specification
RAM1 <- lavaan2RAM(model1, obs.variables=c("W1", "S1", "W2", "S2"))
RAM1
```

```
## $A
##    W1       S1       W2  S2
## W1 "0"      "0"      "0" "0"
## S1 "0"      "0"      "0" "0"
## W2 "0*w2w" "0*s2w" "0" "0"
## S2 "0*w2s" "0*s2s" "0" "0"
##
## $S
```

```
##    W1           S1           W2           S2
## W1 "1"          "0*w1WITHs1" "0"          "0"
## S1 "0*w1WITHs1" "1"          "0"          "0"
## W2 "0"          "0"          "0*Errw2"    "0*w2WITHs2"
## S2 "0"          "0"          "0*w2WITHs2" "0*Errs2"
##
## $F
##    W1 S1 W2 S2
## W1  1  0  0  0
## S1  0  1  0  0
## W2  0  0  1  0
## S2  0  0  0  1
##
## $M
##   W1 S1 W2 S2
## 1  0  0  0  0
```

Model without any moderator

```r
## Create the model implied correlation structure with implicit diagonal constraints
M0 <- create.vechsR(A0=RAM1$A, S0=RAM1$S)

## Create the heterogeneity variance-covariance matrix
## RE.type= either "Diag" or "Symm"
## Transform= either "expLog" or "sqSD" for better estimation on variances
T0 <- create.Tau2(RAM=RAM1, RE.type="Diag", Transform="expLog", RE.startvalues=0.05)

mx.fit0 <- osmasem(model.name="No moderator", Mmatrix=M0, Tmatrix=T0, data=my.df)
summary(mx.fit0)
```

```
## Summary of No moderator
##
## free parameters:
##        name  matrix row col    Estimate  Std.Error A    z value     Pr(>|z|)
## 1       w2w      A0  W2  W1  0.57247128 0.02226456    25.712223 0.000000e+00
## 2       w2s      A0  S2  W1  0.08023681 0.02484213     3.229868 1.238472e-03
## 3       s2w      A0  W2  S1  0.08584124 0.02479589     3.461915 5.363474e-04
## 4       s2s      A0  S2  S1  0.58612399 0.02079000    28.192596 0.000000e+00
## 5  w1WITHs1      S0  S1  W1  0.38045213 0.02256155    16.862851 0.000000e+00
## 6  w2WITHs2      S0  S2  W2  0.16888459 0.02523191     6.693294 2.182010e-11
## 7    Tau1_1 vecTau1   1   1 -2.15335897 0.14358420   -14.997186 0.000000e+00
## 8    Tau1_2 vecTau1   2   1 -2.36882272 0.14419316   -16.428122 0.000000e+00
## 9    Tau1_3 vecTau1   3   1 -2.47296573 0.15796633   -15.655018 0.000000e+00
## 10   Tau1_4 vecTau1   4   1 -2.47676234 0.15669552   -15.806210 0.000000e+00
## 11   Tau1_5 vecTau1   5   1 -2.46245709 0.14519713   -16.959406 0.000000e+00
## 12   Tau1_6 vecTau1   6   1 -2.19983833 0.14187317   -15.505669 0.000000e+00
##
## Model Statistics:
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:             12                   180             -300.1701
##    Saturated:             27                   165                    NA
## Independence:             12                   180                    NA
## Number of observations/statistics: 12906/192
```

```
##
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:      -660.1701              -276.1701               -276.1459
## BIC:     -2003.9507              -186.5848               -224.7196
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2019-05-08 12:55:43
## Wall clock time: 0.2900579 secs
## optimizer:  SLSQP
## OpenMx version number: 2.12.2
## Need help?  See help(mxSummary)
```

```
## The variance-covariance matrix in mx.fit0 is based on the untransformed matrix
## Extract the heterogeneity variance-covariance matrix
VarCorr(mx.fit0)
```

```
##            Tau2_1      Tau2_2      Tau2_3      Tau2_4     Tau2_5     Tau2_6
## Tau2_1 0.01347771 0.000000000 0.000000000 0.000000000 0.00000000 0.00000000
## Tau2_2 0.00000000 0.008759246 0.000000000 0.000000000 0.00000000 0.00000000
## Tau2_3 0.00000000 0.000000000 0.007112287 0.000000000 0.00000000 0.00000000
## Tau2_4 0.00000000 0.000000000 0.000000000 0.007058486 0.00000000 0.00000000
## Tau2_5 0.00000000 0.000000000 0.000000000 0.000000000 0.00726335 0.00000000
## Tau2_6 0.00000000 0.000000000 0.000000000 0.000000000 0.00000000 0.01228131
```

Model with Lag as a moderator on the A matrix

```
Ax <- matrix(c(0,0,0,0,
               0,0,0,0,
               "0*data.Lag","0*data.Lag",0,0,
               "0*data.Lag","0*data.Lag",0,0),
             nrow=4, ncol=4, byrow=TRUE)
Ax
```

```
##      [,1]          [,2]          [,3] [,4]
## [1,] "0"           "0"           "0"  "0"
## [2,] "0"           "0"           "0"  "0"
## [3,] "0*data.Lag" "0*data.Lag" "0"  "0"
## [4,] "0*data.Lag" "0*data.Lag" "0"  "0"
```

```
## When there are more than one moderators
## Ax <- list(A1, A2, A3)

## Create the model implied correlation structure with the standardized Lag as the moderator
M1 <- create.vechsR(A0=RAM1$A, S0=RAM1$S, Ax=Ax)

mx.fit1 <- osmasem(model.name="Ax as moderator", Mmatrix=M1, Tmatrix=T0, data=my.df)
summary(mx.fit1)
```

```
## Summary of Ax as moderator
##
## free parameters:
##        name  matrix row col     Estimate  Std.Error A     z value     Pr(>|z|)
## 1       w2w      A0  W2  W1  0.573039769 0.01839765    31.1474465 0.000000e+00
## 2       w2s      A0  S2  W1  0.079844939 0.02419488     3.3000759 9.665869e-04
## 3       s2w      A0  W2  S1  0.085391017 0.02393611     3.5674553 3.604649e-04
```

```
## 4       s2s      A0  S2  S1  0.586234257 0.01962560    29.8708924 0.000000e+00
## 5  w1WITHs1      S0  S1  W1  0.381183684 0.02282277    16.7019054 0.000000e+00
## 6  w2WITHs2      S0  S2  W2  0.166974820 0.02500438     6.6778225 2.425193e-11
## 7     w2w_1      A1  W2  W1 -0.062015588 0.01850573    -3.3511563 8.047487e-04
## 8     w2s_1      A1  S2  W1 -0.025933704 0.02096723    -1.2368683 2.161360e-01
## 9     s2w_1      A1  W2  S1 -0.002382803 0.02055896    -0.1159009 9.077311e-01
## 10    s2s_1      A1  S2  S1 -0.027809761 0.01974171    -1.4086802 1.589298e-01
## 11   Tau1_1 vecTau1   1   1 -2.138190752 0.14360103   -14.8898011 0.000000e+00
## 12   Tau1_2 vecTau1   2   1 -2.630518447 0.16155263   -16.2827334 0.000000e+00
## 13   Tau1_3 vecTau1   3   1 -2.524194286 0.16007543   -15.7687804 0.000000e+00
## 14   Tau1_4 vecTau1   4   1 -2.519909006 0.15983967   -15.7652295 0.000000e+00
## 15   Tau1_5 vecTau1   5   1 -2.537475830 0.14712277   -17.2473353 0.000000e+00
## 16   Tau1_6 vecTau1   6   1 -2.198863585 0.14144100   -15.5461537 0.000000e+00
##
## Model Statistics:
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:             16                   176             -323.6921
##    Saturated:             27                   165                    NA
## Independence:             12                   180                    NA
## Number of observations/statistics: 12906/192
##
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:      -675.6921              -291.6921               -291.6499
## BIC:     -1989.6109              -172.2450               -223.0914
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2019-05-08 12:55:44
## Wall clock time: 0.5262439 secs
## optimizer:  SLSQP
## OpenMx version number: 2.12.2
## Need help?  See help(mxSummary)
```

```
## Extract the residual heterogeneity variance-covariance matrix
VarCorr(mx.fit1)
```

```
##            Tau2_1      Tau2_2     Tau2_3      Tau2_4      Tau2_5     Tau2_6
## Tau2_1 0.01389284 0.000000000 0.00000000 0.000000000 0.000000000 0.00000000
## Tau2_2 0.00000000 0.005189921 0.00000000 0.000000000 0.000000000 0.00000000
## Tau2_3 0.00000000 0.000000000 0.00641967 0.000000000 0.000000000 0.00000000
## Tau2_4 0.00000000 0.000000000 0.00000000 0.006474927 0.000000000 0.00000000
## Tau2_5 0.00000000 0.000000000 0.00000000 0.000000000 0.006251389 0.00000000
## Tau2_6 0.00000000 0.000000000 0.00000000 0.000000000 0.000000000 0.01230528
```

```
## Calculate the R2
## Tau2.0: Heterogeneity variances without the predictors
## Tau2.1: Heterogeneity variances with the predictors
## R2: (Tau2.0-Tau2.1)/Tau2.0
osmasemR2(mx.fit1, mx.fit0)
```

```
## $Tau2.0
##    Tau2_1_1    Tau2_2_2    Tau2_3_3    Tau2_4_4    Tau2_5_5    Tau2_6_6
## 0.013477712 0.008759246 0.007112287 0.007058486 0.007263350 0.012281310
##
## $Tau2.1
##    Tau2_1_1    Tau2_2_2    Tau2_3_3    Tau2_4_4    Tau2_5_5    Tau2_6_6
## 0.013892842 0.005189921 0.006419670 0.006474927 0.006251389 0.012305276
```

```
##
## $R2
##   Tau2_1_1   Tau2_2_2   Tau2_3_3   Tau2_4_4   Tau2_5_5   Tau2_6_6
## 0.00000000 0.40749232 0.09738313 0.08267489 0.13932428 0.00000000
```

```
## Compare the models with and without the moderator
anova(mx.fit1, mx.fit0)
```

```
##              base   comparison ep  minus2LL  df       AIC   diffLL diffdf            p
## 1 Ax as moderator         <NA> 16 -323.6921 176 -675.6921       NA     NA           NA
## 2 Ax as moderator No moderator 12 -300.1701 180 -660.1701 23.52199      4 9.957461e-05
```

```
## Get the estimated A0 and A1
A0 <- mxEval(A0, mx.fit1$mx.fit)
A0
```

```
##            W1         S1 W2 S2
## W1 0.00000000 0.00000000  0  0
## S1 0.00000000 0.00000000  0  0
## W2 0.57303977 0.08539102  0  0
## S2 0.07984494 0.58623426  0  0
```

```
A1 <- mxEval(A1, mx.fit1$mx.fit)
A1
```

```
##             W1           S1 W2 S2
## W1  0.00000000  0.000000000  0  0
## S1  0.00000000  0.000000000  0  0
## W2 -0.06201559 -0.002382803  0  0
## S2 -0.02593370 -0.027809761  0  0
```

```
## Compute the estimated A matrix at -1SD (-1) of the standardized Lag
A0 - A1
```

```
##           W1         S1 W2 S2
## W1 0.0000000 0.00000000  0  0
## S1 0.0000000 0.00000000  0  0
## W2 0.6350554 0.08777382  0  0
## S2 0.1057786 0.61404402  0  0
```

```
## Compute the estimated A matrix at 0 (mean) of the standardized Lag
A0
```

```
##            W1         S1 W2 S2
## W1 0.00000000 0.00000000  0  0
## S1 0.00000000 0.00000000  0  0
## W2 0.57303977 0.08539102  0  0
## S2 0.07984494 0.58623426  0  0
```

```
## Compute the estimated A matrix at +1SD (+1) of the standardized Lag
A0 + A1
```

```
##            W1         S1 W2 S2
## W1 0.00000000 0.00000000  0  0
## S1 0.00000000 0.00000000  0  0
## W2 0.51102418 0.08300821  0  0
## S2 0.05391124 0.55842450  0  0
```

TSSEM (with 1/4 variables (3/6 correlations) per study randomly deleted)

A model without any moderator

```
## Set seed for reproducibility
set.seed(345678)

## A function to create missing data: 1/4 of the variables were randomly deleted.
del_rand <- function(x, pattern=c(TRUE, TRUE, TRUE, FALSE)) {
  filter <- sample(pattern)
  x[!filter, ] <- NA
  x[, !filter] <- NA
  diag(x) <- 1
  x
}

data.missing <- lapply(Nohe15A1$data, del_rand)

## Display the number of data points
pattern.na(data.missing, show.na=FALSE)
```

```
##    W1 S1 W2 S2
## W1 32 17 16 19
## S1 17 32 13 16
## W2 16 13 32 15
## S2 19 16 15 32
```

```
## Proposed model in lavaan syntax
model1 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ w1WITHs1*S1
           W2 ~~ w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ Errw2*W2
           S2 ~~ Errs2*S2'

RAM1 <- lavaan2RAM(model1, obs.variables=c("W1", "S1", "W2", "S2"))
RAM1
```

```
## $A
##    W1      S1      W2  S2
## W1 "0"     "0"     "0" "0"
## S1 "0"     "0"     "0" "0"
## W2 "0*w2w" "0*s2w" "0" "0"
## S2 "0*w2s" "0*s2s" "0" "0"
##
## $S
##    W1           S1           W2           S2
## W1 "1"          "0*w1WITHs1" "0"          "0"
## S1 "0*w1WITHs1" "1"          "0"          "0"
## W2 "0"          "0"          "0*Errw2"    "0*w2WITHs2"
## S2 "0"          "0"          "0*w2WITHs2" "0*Errs2"
```

```
##
## $F
##    W1 S1 W2 S2
## W1  1  0  0  0
## S1  0  1  0  0
## W2  0  0  1  0
## S2  0  0  0  1
##
## $M
##   W1 S1 W2 S2
## 1  0  0  0  0
```

```
## Stage 1 analysis
random1 <- tssem1(data.missing, Nohe15A1$n, method="REM", RE.type="Diag")
summary(random1)
```

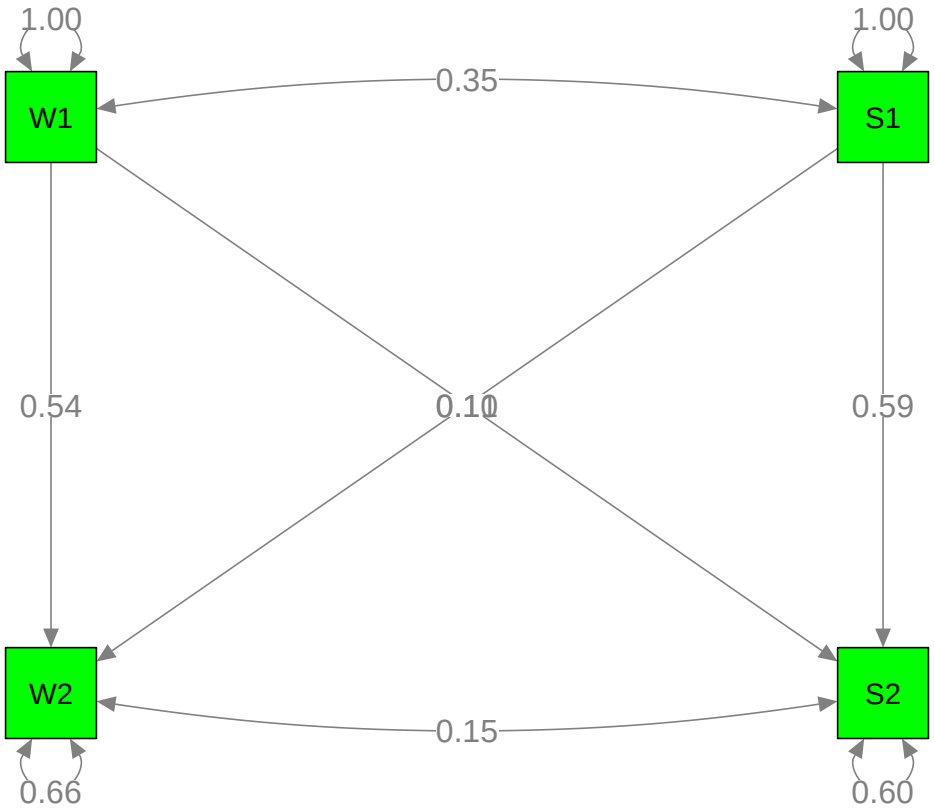
```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = Diag(paste0(RE.startvalues,
##     "*Tau2_", 1:no.es, "_", 1:no.es)), RE.lbound = RE.lbound,
##     I2 = I2, model.name = model.name, suppressWarnings = TRUE,
##     silent = silent, run = run)
##
## 95% confidence intervals: z statistic approximation
## Coefficients:
##               Estimate   Std.Error      lbound      ubound z value  Pr(>|z|)
## Intercept1  0.35496214  0.03601063  0.28438260  0.42554167  9.8571 < 2.2e-16 ***
## Intercept2  0.57373355  0.02846038  0.51795223  0.62951487 20.1590 < 2.2e-16 ***
## Intercept3  0.32254576  0.02960718  0.26451675  0.38057476 10.8942 < 2.2e-16 ***
## Intercept4  0.29302761  0.02610461  0.24186352  0.34419170 11.2251 < 2.2e-16 ***
## Intercept5  0.62645060  0.02893555  0.56973797  0.68316324 21.6499 < 2.2e-16 ***
## Intercept6  0.38735948  0.03150429  0.32561221  0.44910675 12.2955 < 2.2e-16 ***
## Tau2_1_1    0.01830452  0.00710088  0.00438705  0.03222200  2.5778  0.009944 **
## Tau2_2_2    0.00918413  0.00425451  0.00084544  0.01752281  2.1587  0.030875 *
## Tau2_3_3    0.01251147  0.00479304  0.00311728  0.02190566  2.6103  0.009045 **
## Tau2_4_4    0.00567331  0.00315636 -0.00051303  0.01185965  1.7974  0.072268 .
## Tau2_5_5    0.01048377  0.00435938  0.00193955  0.01902800  2.4049  0.016178 *
## Tau2_6_6    0.01098569  0.00506233  0.00106371  0.02090766  2.1701  0.030000 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 576.2832
## Degrees of freedom of the Q statistic: 90
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                               Estimate
## Intercept1: I2 (Q statistic)   0.8908
## Intercept2: I2 (Q statistic)   0.8088
## Intercept3: I2 (Q statistic)   0.8392
## Intercept4: I2 (Q statistic)   0.6967
## Intercept5: I2 (Q statistic)   0.8366
## Intercept6: I2 (Q statistic)   0.8165
##
## Number of studies (or clusters): 32
```

```
## Number of observed statistics: 96
## Number of estimated parameters: 12
## Degrees of freedom: 84
## -2 log likelihood: -137.0305
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```
## Stage 2 analysis
random2 <- tssem2(random1, Amatrix=RAM1$A, Smatrix=RAM1$S)
summary(random2)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s      0.585764  0.035132 0.516907 0.654621 16.6734 < 2.2e-16 ***
## w2s      0.114622  0.041565 0.033155 0.196088  2.7576 0.0058224 **
## s2w      0.102258  0.037398 0.028960 0.175557  2.7343 0.0062505 **
## w2w      0.537436  0.034106 0.470589 0.604283 15.7577 < 2.2e-16 ***
## w1WITHs1 0.354962  0.036011 0.284383 0.425542  9.8571 < 2.2e-16 ***
## w2WITHs2 0.149952  0.039310 0.072906 0.226999  3.8146 0.0001364 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                                Value
## Sample size                                  12906.0
## Chi-square of target model                       0.0
## DF of target model                               0.0
## p value of target model                          0.0
## Number of constraints imposed on "Smatrix"       0.0
## DF manually adjusted                             0.0
## Chi-square of independence model              1275.2
## DF of independence model                         6.0
## RMSEA                                            0.0
## RMSEA lower 95% CI                               0.0
## RMSEA upper 95% CI                               0.0
## SRMR                                             0.0
## TLI                                             -Inf
## CFI                                              1.0
## AIC                                              0.0
## BIC                                              0.0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```
## Plot the model
plot(random2, col="green")
```



Models with three subgroup analysis

```
## Get the necessary functions
## source("http://www.suzannejak.nl/subgroup.functions.R")

data <- data.missing
n <- Nohe15A1$n
Lag <- Nohe15A1$Lag

# Data for studies with short Lag
data_g1 <- data[Lag<7]
n_g1 <- n[Lag<7]

# Data for studies with medium Lag
data_g2 <- data[Lag>=7&Lag<13]
n_g2 <- n[Lag>=7&Lag<13]

# Data for studies with long Lag
data_g3 <- data[Lag>=13]
n_g3 <- n[Lag>=13]
```

Fitting a fix-effects Stage 1 model in three subgroups as there is not enough data

```
## Stage 1 analysis per subgroup (random-effects analysis)
stage1_g1.fit <- tssem1(Cov = data_g1, n = n_g1, method = "REM", RE.type = "Zero")
```

```r
stage1_g2.fit <- tssem1(Cov = data_g2, n = n_g2, method = "REM", RE.type = "Zero")
stage1_g3.fit <- tssem1(Cov = data_g3, n = n_g3, method = "REM", RE.type = "Zero")

## Results
summary(stage1_g1.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = matrix(0, ncol = no.es,
##     nrow = no.es), 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 0.442386  0.022220 0.398836 0.485935 19.9096 < 2.2e-16 ***
## Intercept2 0.629767  0.030345 0.570293 0.689242 20.7538 < 2.2e-16 ***
## Intercept3 0.355986  0.018227 0.320262 0.391710 19.5308 < 2.2e-16 ***
## Intercept4 0.349565  0.045162 0.261049 0.438081  7.7402 9.992e-15 ***
## Intercept5 0.739007  0.017366 0.704969 0.773044 42.5541 < 2.2e-16 ***
## Intercept6 0.412774  0.027228 0.359409 0.466140 15.1601 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 128.2161
## Degrees of freedom of the Q statistic: 24
## P value of the Q statistic: 3.330669e-16
##
## Heterogeneity indices (based on the estimated Tau2):
##                               Estimate
## Intercept1: I2 (Q statistic)        0
## Intercept2: I2 (Q statistic)        0
## Intercept3: I2 (Q statistic)        0
## Intercept4: I2 (Q statistic)        0
## Intercept5: I2 (Q statistic)        0
## Intercept6: I2 (Q statistic)        0
##
## Number of studies (or clusters): 10
## Number of observed statistics: 30
## Number of estimated parameters: 6
## Degrees of freedom: 24
## -2 log likelihood: 13.79085
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```r
summary(stage1_g2.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = matrix(0, ncol = no.es,
##     nrow = no.es), 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 0.408242  0.015638 0.377592 0.438893  26.105 < 2.2e-16 ***
## Intercept2 0.588740  0.024557 0.540609 0.636871  23.974 < 2.2e-16 ***
## Intercept3 0.348951  0.016754 0.316114 0.381788  20.828 < 2.2e-16 ***
## Intercept4 0.292873  0.017981 0.257630 0.328116  16.288 < 2.2e-16 ***
## Intercept5 0.589811  0.012008 0.566277 0.613346  49.119 < 2.2e-16 ***
## Intercept6 0.289545  0.019185 0.251943 0.327148  15.092 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 275.8132
## Degrees of freedom of the Q statistic: 39
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                               Estimate
## Intercept1: I2 (Q statistic)        0
## Intercept2: I2 (Q statistic)        0
## Intercept3: I2 (Q statistic)        0
## Intercept4: I2 (Q statistic)        0
## Intercept5: I2 (Q statistic)        0
## Intercept6: I2 (Q statistic)        0
##
## Number of studies (or clusters): 15
## Number of observed statistics: 45
## Number of estimated parameters: 6
## Degrees of freedom: 39
## -2 log likelihood: 91.67945
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```
summary(stage1_g3.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = matrix(0, ncol = no.es,
##     nrow = no.es), 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 0.393140  0.017151 0.359524 0.426756  22.922 < 2.2e-16 ***
## Intercept2 0.581720  0.011511 0.559159 0.604281  50.536 < 2.2e-16 ***
## Intercept3 0.344147  0.029203 0.286911 0.401383  11.785 < 2.2e-16 ***
## Intercept4 0.284446  0.017365 0.250411 0.318481  16.381 < 2.2e-16 ***
## Intercept5 0.634615  0.056856 0.523179 0.746050  11.162 < 2.2e-16 ***
## Intercept6 0.496591  0.027686 0.442328 0.550854  17.937 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 141.0071
## Degrees of freedom of the Q statistic: 15
## P value of the Q statistic: 0
```

```
##
## Heterogeneity indices (based on the estimated Tau2):
##                               Estimate
## Intercept1: I2 (Q statistic)        0
## Intercept2: I2 (Q statistic)        0
## Intercept3: I2 (Q statistic)        0
## Intercept4: I2 (Q statistic)        0
## Intercept5: I2 (Q statistic)        0
## Intercept6: I2 (Q statistic)        0
##
## Number of studies (or clusters): 7
## Number of observed statistics: 21
## Number of estimated parameters: 6
## Degrees of freedom: 15
## -2 log likelihood: 50.65641
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

Fitting the Stage 2 model in three subgroups

```r
## Stage 2 analysis per subgroup (random-effect analysis)
stage2_g1.fit <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)
stage2_g2.fit <- tssem2(stage1_g2.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)
stage2_g3.fit <- tssem2(stage1_g3.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)

## Results
summary(stage2_g1.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s       0.723023  0.024077  0.675833  0.770213 30.0297 < 2.2e-16 ***
## w2s       0.036131  0.025974 -0.014777  0.087038  1.3910 0.1642116
## s2w       0.088233  0.058739 -0.026893  0.203358  1.5021 0.1330653
## w2w       0.590734  0.043612  0.505257  0.676212 13.5453 < 2.2e-16 ***
## w1WITHs1  0.442386  0.022220  0.398836  0.485935 19.9096 < 2.2e-16 ***
## w2WITHs2  0.137277  0.037886  0.063022  0.211531  3.6235 0.0002907 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                           Value
## Sample size                              2845.0
## Chi-square of target model                  0.0
## DF of target model                          0.0
```

```
## p value of target model                          0.0
## Number of constraints imposed on "Smatrix"       0.0
## DF manually adjusted                             0.0
## Chi-square of independence model              2509.9
## DF of independence model                         6.0
## RMSEA                                            0.0
## RMSEA lower 95% CI                               0.0
## RMSEA upper 95% CI                               0.0
## SRMR                                             0.0
## TLI                                             -Inf
## CFI                                              1.0
## AIC                                              0.0
## BIC                                              0.0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```
summary(stage2_g2.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s      0.536823  0.015511 0.506422 0.567223 34.6094 < 2.2e-16 ***
## w2s      0.129798  0.019390 0.091794 0.167801  6.6941  2.17e-11 ***
## s2w      0.063029  0.025349 0.013346 0.112713  2.4864   0.01290 *
## w2w      0.563009  0.030004 0.504203 0.621815 18.7647 < 2.2e-16 ***
## w1WITHs1 0.408242  0.015638 0.377592 0.438893 26.1051 < 2.2e-16 ***
## w2WITHs2 0.055907  0.020551 0.015628 0.096187  2.7204   0.00652 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                            Value
## Sample size                                 5991
## Chi-square of target model                     0
## DF of target model                             0
## p value of target model                        0
## Number of constraints imposed on "Smatrix"     0
## DF manually adjusted                           0
## Chi-square of independence model            3599
## DF of independence model                       6
## RMSEA                                          0
## RMSEA lower 95% CI                             0
## RMSEA upper 95% CI                             0
## SRMR                                           0
## TLI                                         -Inf
## CFI                                            1
## AIC                                            0
```

```
## BIC                                                 0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```
summary(stage2_g3.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s      0.590599  0.067734 0.457844 0.723355  8.7194 < 2.2e-16 ***
## w2s      0.111959  0.043474 0.026750 0.197167  2.5753 0.0100161 *
## s2w      0.065940  0.017859 0.030938 0.100943  3.6923 0.0002222 ***
## w2w      0.555796  0.014296 0.527777 0.583815 38.8790 < 2.2e-16 ***
## w1WITHs1 0.393140  0.017151 0.359524 0.426756 22.9219 < 2.2e-16 ***
## w2WITHs2 0.263469  0.026183 0.212152 0.314786 10.0627 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                             Value
## Sample size                                4070.0
## Chi-square of target model                    0.0
## DF of target model                            0.0
## p value of target model                       0.0
## Number of constraints imposed on "Smatrix"    0.0
## DF manually adjusted                          0.0
## Chi-square of independence model           3223.7
## DF of independence model                      6.0
## RMSEA                                         0.0
## RMSEA lower 95% CI                            0.0
## RMSEA upper 95% CI                            0.0
## SRMR                                          0.0
## TLI                                          -Inf
## CFI                                           1.0
## AIC                                           0.0
## BIC                                           0.0
## 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 group 2 and group 3.

```
## Proposed model g2
model2 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ g2w1WITHs1*S1
           W2 ~~ g2w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g2Errw2*W2
           S2 ~~ g2Errs2*S2'

RAM2 <- lavaan2RAM(model2, obs.variables=c("W1", "S1", "W2", "S2"))

## Proposed model g3
model3 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ g3w1WITHs1*S1
           W2 ~~ g3w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g3Errw2*W2
           S2 ~~ g3Errs2*S2'

RAM3 <- lavaan2RAM(model3, obs.variables=c("W1", "S1", "W2", "S2"))

## Create the models for the two groups, make sure to set the argument run=FALSE
stage2_g1 <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S, run=FALSE, model.name="g1")

stage2_g2 <- tssem2(stage1_g2.fit, Amatrix=RAM1$A, Smatrix=RAM2$S, run=FALSE, model.name="g2")

stage2_g3 <- tssem2(stage1_g3.fit, Amatrix=RAM1$A, Smatrix=RAM3$S, run=FALSE, model.name="g3")

## Create the multigroup model
stage2_constrained <- mxModel(model="same_regression_coef", stage2_g1, stage2_g2,stage2_g3,
                              mxFitFunctionMultigroup(c("g1", "g2", "g3")))

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

## first make a list of the fitted models in the separate groups
submodels.fit <- list(stage2_g1.fit, stage2_g2.fit, stage2_g3.fit)

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

```
## # # # # # # # # # # # # # # # # # # #
##  Output for subgroup MASEM analysis
## # # # # # # # # # # # # # # # # # # #
##
##  Total sample size: 12906
##
##  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   0.000
##          Chi-square   0.000
##                   p   0.000
##               RMSEA     Inf
##  RMSEA lower 95% CI     Inf
##  RMSEA upper 95% CI     Inf
##                 CFI   1.000
##                 TLI    -Inf
##                 AIC  60.000
##                 BIC 283.963
##                SRMR   0.000
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Fit indices of the model with equality constraints:
##
##           Statistic Constrained_m2
##                  df          8.000
##          Chi-square         52.163
##                   p          0.000
##               RMSEA          0.036
##  RMSEA lower 95% CI          0.025
##  RMSEA upper 95% CI          0.047
##                 CFI          0.995
##                 TLI          0.989
##                 AIC         96.163
##                 BIC        260.403
##                SRMR          0.026
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Chi-square difference between free and constrained model:
##
##   Statistic Diff_m1_m2
##          df      8.000
##  Chi-square     52.163
##           p      0.000
##
##  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

Testing the equality of one regression coefficient (w2w)

- 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 group 2 and group 3.

```
## Proposed model g2
model2 <- 'W2 ~ w2w*W1 + g2s2w*S1
           S2 ~ g2w2s*W1 + g2s2s*S1
           W1 ~~ g2w1WITHs1*S1
           W2 ~~ g2w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g2Errw2*W2
           S2 ~~ g2Errs2*S2'
```

```
RAM2 <- lavaan2RAM(model2, obs.variables=c("W1", "S1", "W2", "S2"))

## Proposed model g3
model3 <- 'W2 ~ w2w*W1 + g3s2w*S1
           S2 ~ g3w2s*W1 + g3s2s*S1
           W1 ~~ g3w1WITHs1*S1
           W2 ~~ g3w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g3Errw2*W2
           S2 ~~ g3Errs2*S2'

RAM3 <- lavaan2RAM(model3, obs.variables=c("W1", "S1", "W2", "S2"))

## Create the models for the two groups, make sure to set the argument run=FALSE
stage2_g1 <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S, run=FALSE, model.name="g1")

stage2_g2 <- tssem2(stage1_g2.fit, Amatrix=RAM2$A, Smatrix=RAM2$S, run=FALSE, model.name="g2")

stage2_g3 <- tssem2(stage1_g3.fit, Amatrix=RAM3$A, Smatrix=RAM3$S, run=FALSE, model.name="g3")

## Create the multigroup model
stage2_constrained <- mxModel(model="same_regression_coef", stage2_g1, stage2_g2,stage2_g3,
                              mxFitFunctionMultigroup(c("g1", "g2", "g3")))

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

## First make a list of the fitted models in the separate groups
submodels.fit <- list(stage2_g1.fit,stage2_g2.fit,stage2_g3.fit)

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

```
## # # # # # # # # # # # # # # # # # # #
##  Output for subgroup MASEM analysis
## # # # # # # # # # # # # # # # # # # #
##
##  Total sample size: 12906
##
##  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   0.000
##          Chi-square   0.000
##                   p   0.000
##               RMSEA     Inf
##  RMSEA lower 95% CI     Inf
##  RMSEA upper 95% CI     Inf
```

```
##                 CFI   1.000
##                 TLI    -Inf
##                 AIC  60.000
##                 BIC 283.963
##                SRMR   0.000
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Fit indices of the model with equality constraints:
##
##           Statistic Constrained_m2
##                  df          2.000
##          Chi-square          0.596
##                   p          0.742
##               RMSEA          0.000
##  RMSEA lower 95% CI          0.000
##  RMSEA upper 95% CI          0.025
##                 CFI          1.000
##                 TLI          1.001
##                 AIC         56.596
##                 BIC        265.628
##                SRMR          0.005
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Chi-square difference between free and constrained model:
##
##   Statistic Diff_m1_m2
##          df      2.000
##  Chi-square      0.596
##           p      0.742
##
##  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

Models with two subgroup analysis

```r
# Data for studies with short Lag
data_g1 <- data[Lag<12]
n_g1 <- n[Lag<12]

# Data for studies with long Lag
data_g2 <- data[Lag>=12]
n_g2 <- n[Lag>=12]
```

Fitting a fixed-effects Stage 1 model in two subgroups as there is not enough data

```r
## Stage 1 analysis per subgroup (random-effects analysis)
stage1_g1.fit <- tssem1(Cov = data_g1, n = n_g1, method = "REM", RE.type = "Zero")
stage1_g2.fit <- tssem1(Cov = data_g2, n = n_g2, method = "REM", RE.type = "Zero")

summary(stage1_g1.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = matrix(0, ncol = no.es,
```

```
##     nrow = no.es), 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 0.495918  0.015262 0.466004 0.525831  32.493 < 2.2e-16 ***
## Intercept2 0.645661  0.027231 0.592289 0.699033  23.710 < 2.2e-16 ***
## Intercept3 0.405956  0.014437 0.377659 0.434253  28.118 < 2.2e-16 ***
## Intercept4 0.418468  0.027556 0.364459 0.472477  15.186 < 2.2e-16 ***
## Intercept5 0.701277  0.011791 0.678167 0.724386  59.477 < 2.2e-16 ***
## Intercept6 0.389202  0.023183 0.343764 0.434639  16.788 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 274.2606
## Degrees of freedom of the Q statistic: 33
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                              Estimate
## Intercept1: I2 (Q statistic)        0
## Intercept2: I2 (Q statistic)        0
## Intercept3: I2 (Q statistic)        0
## Intercept4: I2 (Q statistic)        0
## Intercept5: I2 (Q statistic)        0
## Intercept6: I2 (Q statistic)        0
##
## Number of studies (or clusters): 13
## Number of observed statistics: 39
## Number of estimated parameters: 6
## Degrees of freedom: 33
## -2 log likelihood: 115.598
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

```
summary(stage1_g2.fit)
```

```
##
## Call:
## meta(y = ES, v = acovR, RE.constraints = matrix(0, ncol = no.es,
##     nrow = no.es), 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 0.348831  0.013561 0.322252 0.375410  25.724 < 2.2e-16 ***
## Intercept2 0.572044  0.012149 0.548232 0.595855  47.086 < 2.2e-16 ***
## Intercept3 0.276310  0.017762 0.241496 0.311123  15.556 < 2.2e-16 ***
## Intercept4 0.260796  0.013339 0.234652 0.286940  19.552 < 2.2e-16 ***
## Intercept5 0.561047  0.017352 0.527038 0.595057  32.333 < 2.2e-16 ***
## Intercept6 0.352792  0.016809 0.319847 0.385737  20.988 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Q statistic on the homogeneity of effect sizes: 224.2743
## Degrees of freedom of the Q statistic: 51
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                              Estimate
## Intercept1: I2 (Q statistic)        0
## Intercept2: I2 (Q statistic)        0
## Intercept3: I2 (Q statistic)        0
## Intercept4: I2 (Q statistic)        0
## Intercept5: I2 (Q statistic)        0
## Intercept6: I2 (Q statistic)        0
##
## Number of studies (or clusters): 19
## Number of observed statistics: 57
## Number of estimated parameters: 6
## Degrees of freedom: 51
## -2 log likelihood: -2.751177
## 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_g1.fit <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)
stage2_g2.fit <- tssem2(stage1_g2.fit, Amatrix=RAM1$A, Smatrix=RAM1$S)

summary(stage2_g1.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s      0.663014  0.017066 0.629566 0.696462 38.8508 < 2.2e-16 ***
## w2s      0.077156  0.019047 0.039825 0.114486  4.0509 5.103e-05 ***
## s2w      0.130325  0.039301 0.053296 0.207353  3.3160  0.000913 ***
## w2w      0.581031  0.037971 0.506609 0.655453 15.3020 < 2.2e-16 ***
## w1WITHs1 0.495918  0.015262 0.466004 0.525831 32.4931 < 2.2e-16 ***
## w2WITHs2 0.061935  0.025071 0.012797 0.111074  2.4704  0.013496 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                           Value
## Sample size                                4863
```

```
## Chi-square of target model                     0
## DF of target model                             0
## p value of target model                        0
## Number of constraints imposed on "Smatrix"     0
## DF manually adjusted                           0
## Chi-square of independence model            4713
## DF of independence model                       6
## RMSEA                                          0
## RMSEA lower 95% CI                             0
## RMSEA upper 95% CI                             0
## SRMR                                           0
## TLI                                         -Inf
## CFI                                            1
## AIC                                            0
## BIC                                            0
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values indicate problems.)
```

```
summary(stage2_g2.fit)
```

```
##
## Call:
## wls(Cov = pooledS, aCov = aCov, 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|)
## s2s      0.529037  0.020362 0.489128 0.568946 25.9817 < 2.2e-16 ***
## w2s      0.091765  0.021627 0.049376 0.134154  4.2430 2.205e-05 ***
## s2w      0.069735  0.015724 0.038916 0.100554  4.4349 9.210e-06 ***
## w2w      0.547718  0.014317 0.519656 0.575780 38.2553 < 2.2e-16 ***
## w1WITHs1 0.348831  0.013561 0.322252 0.375410 25.7235 < 2.2e-16 ***
## w2WITHs2 0.162327  0.018770 0.125540 0.199115  8.6484 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Goodness-of-fit indices:
##                                            Value
## Sample size                               8043.0
## Chi-square of target model                   0.0
## DF of target model                           0.0
## p value of target model                      0.0
## Number of constraints imposed on "Smatrix"   0.0
## DF manually adjusted                         0.0
## Chi-square of independence model          3984.8
## DF of independence model                     6.0
## RMSEA                                        0.0
## RMSEA lower 95% CI                           0.0
## RMSEA upper 95% CI                           0.0
## SRMR                                         0.0
## TLI                                         -Inf
```

```
## CFI                                                 1.0
## AIC                                                 0.0
## BIC                                                 0.0
## 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 group 2.

```
## Proposed model g2
model2 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ g2w1WITHs1*S1
           W2 ~~ g2w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ g2Errw2*W2
           S2 ~~ g2Errs2*S2'

RAM2 <- lavaan2RAM(model2, obs.variables=c("W1", "S1", "W2", "S2"))

# Create the models for the two groups, make sure to set the argument run=FALSE
stage2_g1 <- tssem2(stage1_g1.fit, Amatrix=RAM1$A, Smatrix=RAM1$S, run=FALSE, model.name="g1")

stage2_g2 <- tssem2(stage1_g2.fit, Amatrix=RAM1$A, Smatrix=RAM2$S, run=FALSE, model.name="g2")

# Create the multigroup model
stage2_constrained <- mxModel(model="same_regression_coef", stage2_g1, stage2_g2,
                              mxFitFunctionMultigroup(c("g1", "g2")))

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

# first make a list of the fitted models in the separate groups
submodels.fit <- list(stage2_g1.fit,stage2_g2.fit)

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

```
## # # # # # # # # # # # # # # # # # # #
##  Output for subgroup MASEM analysis
## # # # # # # # # # # # # # # # # # # #
##
##  Total sample size: 12906
##
##  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   0.000
##          Chi-square   0.000
##                   p   0.000
##               RMSEA     Inf
##  RMSEA lower 95% CI     Inf
##  RMSEA upper 95% CI     Inf
##                 CFI   1.000
##                 TLI    -Inf
##                 AIC  40.000
##                 BIC 189.309
##               SRMR   0.000
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Fit indices of the model with equality constraints:
##
##           Statistic Constrained_m2
##                  df          4.000
##          Chi-square         45.456
##                   p          0.000
##               RMSEA          0.040
##  RMSEA lower 95% CI          0.028
##  RMSEA upper 95% CI          0.053
##                 CFI          0.995
##                 TLI          0.986
##                 AIC         77.456
##                 BIC        196.903
##               SRMR          0.037
## - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
##  Chi-square difference between free and constrained model:
##
##   Statistic Diff_m1_m2
##          df      4.000
##  Chi-square     45.456
##           p      0.000
##
##  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

OSMASEM (with 1/4 variables (3/6 correlations) per study randomly deleted)

Data preparation

```
## Get the data
n <- Nohe15A1$n
Lag <- Nohe15A1$Lag

## Calculate the sampling covariance matrix of the correlations
my.df <- Cor2DataFrame(data.missing, n, acov = "weighted")

## Add standardized Lag as a moderator.
## Standardization of the moderator improves the convergence.
my.df$data <- data.frame(my.df$data, Lag=scale(Nohe15A1$Lag),
```

```
                          check.names=FALSE)
head(my.df$data)
```

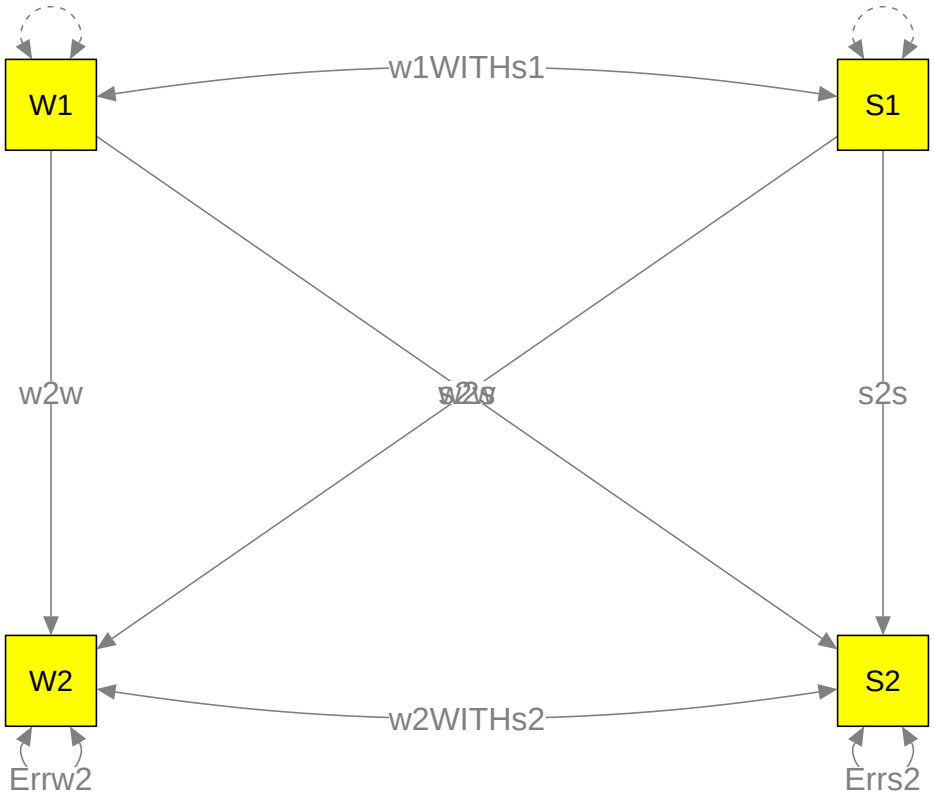
```
##                                       S1_W1 W2_W1 S2_W1 W2_S1 S2_S1 S2_W2 C(S1_W1 S1_W1)
## Britt...Dawson..2005.                    NA  0.58  0.22    NA    NA  0.27   0.0013998484
## Demerouti.et.al...2004.                  NA    NA    NA  0.41  0.68  0.54   0.0020433630
## Ford..2010.                            0.35    NA  0.32    NA  0.74    NA   0.0020869716
## Hammer.et.al...2005...female.subsample   NA    NA    NA  0.30  0.43  0.30   0.0029253245
## Hammer.et.al...2005...male.subsample    0.19  0.54    NA  0.21    NA    NA   0.0029253209
## Innstrand.et.al...2008.                 0.42  0.63    NA  0.30    NA    NA   0.0003062759
##                                       C(W2_W1 S1_W1) C(S2_W1 S1_W1) C(W2_S1 S1_W1) C(S2_S1 S1_W1)
## Britt...Dawson..2005.                   1.877752e-04   0.0008664212   0.0008400105   2.294741e-04
## Demerouti.et.al...2004.                 2.740967e-04   0.0012647194   0.0012261672   3.349653e-04
## Ford..2010.                             2.799474e-04   0.0012917106   0.0012523357   3.421137e-04
## Hammer.et.al...2005...female.subsample  3.924015e-04   0.0018105987   0.0017554066   4.795423e-04
## Hammer.et.al...2005...male.subsample    3.923979e-04   0.0018105942   0.0017554012   4.795393e-04
## Innstrand.et.al...2008.                 4.108388e-05   0.0001895664   0.0001837879   5.020725e-05
##                                       C(S2_W2 S1_W1) C(W2_W1 W2_W1) C(S2_W1 W2_W1) C(W2_S1 W2_W1)
## Britt...Dawson..2005.                   0.0004902557   0.0008846510   3.149208e-04   4.179668e-04
## Demerouti.et.al...2004.                 0.0007156300   0.0012913267   4.596920e-04   6.101080e-04
## Ford..2010.                             0.0007309028   0.0013188863   4.695035e-04   6.231299e-04
## Hammer.et.al...2005...female.subsample  0.0010245096   0.0018486933   6.581039e-04   8.734432e-04
## Hammer.et.al...2005...male.subsample    0.0010245043   0.0018486916   6.580997e-04   8.734388e-04
## Innstrand.et.al...2008.                 0.0001072644   0.0001935546   6.890238e-05   9.144806e-05
##                                       C(S2_S1 W2_W1) C(S2_W2 W2_W1) C(S2_W1 S2_W1) C(W2_S1 S2_W1)
## Britt...Dawson..2005.                   8.918876e-05   2.956967e-04   0.0015530429   0.0005529344
## Demerouti.et.al...2004.                 1.301896e-04   4.316303e-04   0.0022669823   0.0008071232
## Ford..2010.                             1.329684e-04   4.408430e-04   0.0023153635   0.0008243486
## Hammer.et.al...2005...female.subsample  1.863816e-04   6.179306e-04   0.0032454624   0.0011554919
## Hammer.et.al...2005...male.subsample    1.863795e-04   6.179265e-04   0.0032454578   0.0011554851
## Innstrand.et.al...2008.                 1.951394e-05   6.469632e-05   0.0003397937   0.0001209781
##                                       C(S2_S1 S2_W1) C(S2_W2 S2_W1) C(W2_S1 W2_S1) C(S2_S1 W2_S1)
## Britt...Dawson..2005.                   3.268524e-04   0.0008266340   0.0017159346   3.139485e-04
## Demerouti.et.al...2004.                 4.771091e-04   0.0012066426   0.0025047550   4.582729e-04
## Ford..2010.                             4.872910e-04   0.0012323944   0.0025582107   4.680528e-04
## Hammer.et.al...2005...female.subsample  6.830386e-04   0.0017274545   0.0035858632   6.560722e-04
## Hammer.et.al...2005...male.subsample    6.830351e-04   0.0017274490   0.0035858570   6.560685e-04
## Innstrand.et.al...2008.                 7.151294e-05   0.0001808613   0.0003754331   6.868964e-05
##                                       C(S2_W2 W2_S1) C(S2_S1 S2_S1) C(S2_W2 S2_S1) C(S2_W2 S2_W2)
## Britt...Dawson..2005.                   0.0009758571   0.0007207015   1.813287e-04   0.0015262265
## Demerouti.et.al...2004.                 0.0014244635   0.0010520098   2.646877e-04   0.0022278383
## Ford..2010.                             0.0014548638   0.0010744610   2.703362e-04   0.0022753837
## Hammer.et.al...2005...female.subsample  0.0020392919   0.0015060815   3.789315e-04   0.0031894230
## Hammer.et.al...2005...male.subsample    0.0020392854   0.0015060803   3.789277e-04   0.0031894179
## Innstrand.et.al...2008.                 0.0002135101   0.0001576838   3.967352e-05   0.0003339264
##                                              Lag
## Britt...Dawson..2005.                  -0.6794521
## Demerouti.et.al...2004.                -0.7711151
## Ford..2010.                            -0.8016694
## Hammer.et.al...2005...female.subsample -0.1294740
## Hammer.et.al...2005...male.subsample   -0.1294740
## Innstrand.et.al...2008.                 0.6038301
```

```
## Check the number of studies
pattern.na(my.df, show.na=FALSE, type="osmasem")
```

```
##       S1_W1 W2_W1 S2_W1 W2_S1 S2_S1 S2_W2
## S1_W1    17    26    26    23    23    32
## W2_W1    26    16    26    22    32    22
## S2_W1    26    26    19    32    25    25
## W2_S1    23    22    32    13    23    22
## S2_S1    23    32    25    23    16    25
## S2_W2    32    22    25    22    25    15
```

```
## Proposed model
model1 <- 'W2 ~ w2w*W1 + s2w*S1
           S2 ~ w2s*W1 + s2s*S1
           W1 ~~ w1WITHs1*S1
           W2 ~~ w2WITHs2*S2
           W1 ~~ 1*W1
           S1 ~~ 1*S1
           W2 ~~ Errw2*W2
           S2 ~~ Errs2*S2'

plot(model1, col="yellow")
```




```
## Convert the lavaan syntax into the RAM specification
RAM1 <- lavaan2RAM(model1, obs.variables=c("W1", "S1", "W2", "S2"))
RAM1
```

```
## $A
##    W1      S1      W2  S2
## W1 "0"     "0"     "0" "0"
```

```
## S1 "0"     "0"     "0" "0"
## W2 "0*w2w" "0*s2w" "0" "0"
## S2 "0*w2s" "0*s2s" "0" "0"
##
## $S
##    W1           S1           W2           S2
## W1 "1"          "0*w1WITHs1" "0"          "0"
## S1 "0*w1WITHs1" "1"          "0"          "0"
## W2 "0"          "0"          "0*Errw2"    "0*w2WITHs2"
## S2 "0"          "0"          "0*w2WITHs2" "0*Errs2"
##
## $F
##    W1 S1 W2 S2
## W1  1  0  0  0
## S1  0  1  0  0
## W2  0  0  1  0
## S2  0  0  0  1
##
## $M
##   W1 S1 W2 S2
## 1  0  0  0  0
```

Model without any moderator

```
## Create the model implied correlation structure with implicit diagonal constraints
M0 <- create.vechsR(A0=RAM1$A, S0=RAM1$S)

## Create the heterogeneity variance-covariance matrix
## RE.type= either "Diag" or "Symm"
## Transform= either "expLog" or "sqSD" for better estimation on variances
T0 <- create.Tau2(RAM=RAM1, RE.type="Diag", Transform="expLog", RE.startvalues=0.05)

mx.fit0 <- osmasem(model.name="No moderator", Mmatrix=M0, Tmatrix=T0, data=my.df)
summary(mx.fit0)
```

```
## Summary of No moderator
##
## free parameters:
##        name  matrix row col   Estimate  Std.Error A    z value     Pr(>|z|)
## 1       w2w      A0  W2  W1  0.5378864 0.03312371    16.238712 0.000000e+00
## 2       w2s      A0  S2  W1  0.1133992 0.03958516     2.864689 4.174188e-03
## 3       s2w      A0  W2  S1  0.1022856 0.03617687     2.827376 4.693112e-03
## 4       s2s      A0  S2  S1  0.5853285 0.03383426    17.299878 0.000000e+00
## 5  w1WITHs1      S0  S1  W1  0.3561702 0.03478062    10.240482 0.000000e+00
## 6  w2WITHs2      S0  S2  W2  0.1495221 0.03735490     4.002743 6.261224e-05
## 7    Tau1_1 vecTau1   1   1 -2.0164428 0.18926325   -10.654170 0.000000e+00
## 8    Tau1_2 vecTau1   2   1 -2.2949140 0.20279730   -11.316295 0.000000e+00
## 9    Tau1_3 vecTau1   3   1 -2.2137542 0.18656403   -11.865921 0.000000e+00
## 10   Tau1_4 vecTau1   4   1 -2.5756073 0.26399695    -9.756201 0.000000e+00
## 11   Tau1_5 vecTau1   5   1 -2.2524657 0.19004813   -11.852080 0.000000e+00
## 12   Tau1_6 vecTau1   6   1 -2.2537646 0.21901756   -10.290337 0.000000e+00
##
## Model Statistics:
```

```
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:             12                      84             -140.8313
##    Saturated:             27                      69                    NA
## Independence:             12                      84                    NA
## Number of observations/statistics: 12906/96
##
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:      -308.8313             -116.83130               -116.80710
## BIC:      -935.9289              -27.24593                -65.38072
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2019-05-08 12:55:54
## Wall clock time: 0.240984 secs
## optimizer:  SLSQP
## OpenMx version number: 2.12.2
## Need help?  See help(mxSummary)
```

```
## The variance-covariance matrix in mx.fit0 is based on the untransformed matrix
## Extract the heterogeneity variance-covariance matrix
VarCorr(mx.fit0)
```

```
##            Tau2_1    Tau2_2     Tau2_3      Tau2_4     Tau2_5     Tau2_6
## Tau2_1 0.01772312 0.0000000 0.00000000 0.000000000 0.00000000 0.00000000
## Tau2_2 0.00000000 0.0101546 0.00000000 0.000000000 0.00000000 0.00000000
## Tau2_3 0.00000000 0.0000000 0.01194421 0.000000000 0.00000000 0.00000000
## Tau2_4 0.00000000 0.0000000 0.00000000 0.005792365 0.00000000 0.00000000
## Tau2_5 0.00000000 0.0000000 0.00000000 0.000000000 0.01105435 0.00000000
## Tau2_6 0.00000000 0.0000000 0.00000000 0.000000000 0.00000000 0.01102567
```

Model with Lag as a moderator on the A matrix

```
Ax <- matrix(c(0,0,0,0,
               0,0,0,0,
               "0*data.Lag","0*data.Lag",0,0,
               "0*data.Lag","0*data.Lag",0,0),
             nrow=4, ncol=4, byrow=TRUE)
Ax
```

```
##      [,1]          [,2]          [,3] [,4]
## [1,] "0"           "0"           "0"  "0"
## [2,] "0"           "0"           "0"  "0"
## [3,] "0*data.Lag" "0*data.Lag" "0"  "0"
## [4,] "0*data.Lag" "0*data.Lag" "0"  "0"
```

```
## When there are more than one moderators
## Ax <- list(A1, A2, A3)

## Create the model implied correlation structure with the standardized Lag as the moderator
M1 <- create.vechsR(A0=RAM1$A, S0=RAM1$S, Ax=Ax)

mx.fit1 <- osmasem(model.name="Ax as moderator", Mmatrix=M1, Tmatrix=T0, data=my.df)
summary(mx.fit1)
```

```
## Summary of Ax as moderator
##
```

```
## free parameters:
##        name  matrix row col    Estimate  Std.Error A     z value     Pr(>|z|)
## 1       w2w      A0  W2  W1  0.55456158 0.02666651    20.7961790 0.0000000000
## 2       w2s      A0  S2  W1  0.14362286 0.04454436     3.2242658 0.0012629612
## 3       s2w      A0  W2  S1  0.10532419 0.03497889     3.0110788 0.0026032130
## 4       s2s      A0  S2  S1  0.53648579 0.04024728    13.3297401 0.0000000000
## 5  w1WITHs1      S0  S1  W1  0.35649751 0.03531886    10.0936860 0.0000000000
## 6  w2WITHs2      S0  S2  W2  0.13883471 0.03786649     3.6664265 0.0002459636
## 7     w2w_1      A1  W2  W1 -0.05205183 0.01963389    -2.6511217 0.0080224934
## 8     w2s_1      A1  S2  W1  0.11069099 0.08645869     1.2802761 0.2004480408
## 9     s2w_1      A1  W2  S1 -0.01046078 0.02087855    -0.5010302 0.6163498971
## 10    s2s_1      A1  S2  S1 -0.17181266 0.09094700    -1.8891514 0.0588715441
## 11   Tau1_1 vecTau1   1   1 -1.99825359 0.18951558   -10.5440071 0.0000000000
## 12   Tau1_2 vecTau1   2   1 -2.63083342 0.23783982   -11.0613664 0.0000000000
## 13   Tau1_3 vecTau1   3   1 -2.21216305 0.18761517   -11.7909603 0.0000000000
## 14   Tau1_4 vecTau1   4   1 -2.70374877 0.28096077    -9.6232252 0.0000000000
## 15   Tau1_5 vecTau1   5   1 -2.36706455 0.19746997   -11.9869598 0.0000000000
## 16   Tau1_6 vecTau1   6   1 -2.24336285 0.22025329   -10.1853774 0.0000000000
##
## Model Statistics:
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:             16                    80             -154.1179
##    Saturated:             27                    69                    NA
## Independence:             12                    84                    NA
## Number of observations/statistics: 12906/96
##
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:      -314.1179           -122.117867               -122.07566
## BIC:      -911.3537             -2.670706                -53.51709
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2019-05-08 12:55:55
## Wall clock time: 0.6668763 secs
## optimizer:  SLSQP
## OpenMx version number: 2.12.2
## Need help?  See help(mxSummary)
```

```
## Extract the residual heterogeneity variance-covariance matrix
VarCorr(mx.fit1)
```

```
##            Tau2_1      Tau2_2     Tau2_3      Tau2_4      Tau2_5     Tau2_6
## Tau2_1 0.01837972 0.000000000 0.00000000 0.000000000 0.000000000 0.00000000
## Tau2_2 0.00000000 0.005186652 0.00000000 0.000000000 0.000000000 0.00000000
## Tau2_3 0.00000000 0.000000000 0.01198228 0.000000000 0.000000000 0.00000000
## Tau2_4 0.00000000 0.000000000 0.00000000 0.004482844 0.000000000 0.00000000
## Tau2_5 0.00000000 0.000000000 0.00000000 0.000000000 0.008790101 0.00000000
## Tau2_6 0.00000000 0.000000000 0.00000000 0.000000000 0.000000000 0.01125744
```

```
## Calculate the R2
## Tau2.0: Heterogeneity variances without the predictors
## Tau2.1: Heterogeneity variances with the predictors
## R2: (Tau2.0-Tau2.1)/Tau2.0
osmasemR2(mx.fit1, mx.fit0)
```

```
## $Tau2.0
##    Tau2_1_1    Tau2_2_2    Tau2_3_3    Tau2_4_4    Tau2_5_5    Tau2_6_6
```

```
## 0.017723115 0.010154604 0.011944214 0.005792365 0.011054349 0.011025670
##
## $Tau2.1
##    Tau2_1_1    Tau2_2_2    Tau2_3_3    Tau2_4_4    Tau2_5_5    Tau2_6_6
## 0.018379724 0.005186652 0.011982283 0.004482844 0.008790101 0.011257444
##
## $R2
##  Tau2_1_1  Tau2_2_2  Tau2_3_3  Tau2_4_4  Tau2_5_5  Tau2_6_6
## 0.0000000 0.4892315 0.0000000 0.2260770 0.2048288 0.0000000
```

```
## Compare the models with and without the moderator
anova(mx.fit1, mx.fit0)
```

```
##              base   comparison ep  minus2LL df       AIC   diffLL diffdf           p
## 1 Ax as moderator         <NA> 16 -154.1179 80 -314.1179       NA     NA          NA
## 2 Ax as moderator No moderator 12 -140.8313 84 -308.8313 13.28656      4 0.009957243
```

```
## Get the estimated A0 and A1
A0 <- mxEval(A0, mx.fit1$mx.fit)
A0
```

```
##           W1        S1 W2 S2
## W1 0.0000000 0.0000000  0  0
## S1 0.0000000 0.0000000  0  0
## W2 0.5545616 0.1053242  0  0
## S2 0.1436229 0.5364858  0  0
```

```
A1 <- mxEval(A1, mx.fit1$mx.fit)
A1
```

```
##             W1          S1 W2 S2
## W1  0.00000000  0.00000000  0  0
## S1  0.00000000  0.00000000  0  0
## W2 -0.05205183 -0.01046078  0  0
## S2  0.11069099 -0.17181266  0  0
```

```
## Compute the estimated A matrix at -1SD (-1) of the standardized Lag
A0 - A1
```

```
##            W1        S1 W2 S2
## W1 0.00000000 0.0000000  0  0
## S1 0.00000000 0.0000000  0  0
## W2 0.60661341 0.1157850  0  0
## S2 0.03293187 0.7082984  0  0
```

```
## Compute the estimated A matrix at 0 (mean) of the standardized Lag
A0
```

```
##           W1        S1 W2 S2
## W1 0.0000000 0.0000000  0  0
## S1 0.0000000 0.0000000  0  0
## W2 0.5545616 0.1053242  0  0
## S2 0.1436229 0.5364858  0  0
```

```
## Compute the estimated A matrix at +1SD (+1) of the standardized Lag
A0 + A1
```

```
##           W1         S1 W2 S2
## W1 0.0000000 0.00000000  0  0
```

```
## S1 0.0000000 0.00000000  0  0
## W2 0.5025098 0.09486341  0  0
## S2 0.2543138 0.36467313  0  0
```

```
sessionInfo()
```

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.2 LTS
##
## Matrix products: default
## BLAS: /opt/microsoft/ropen/3.5.1/lib64/R/lib/libRblas.so
## LAPACK: /opt/microsoft/ropen/3.5.1/lib64/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
##  [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] metaSEM_1.2.1.1      OpenMx_2.12.2        RevoUtils_11.0.1     RevoUtilsMath_11.0.0
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-137        RColorBrewer_1.1-2  mi_1.0              tools_3.5.1
##  [5] backports_1.1.3     R6_2.4.0            d3Network_0.5.2.1   rpart_4.1-13
##  [9] Hmisc_4.2-0         lazyeval_0.2.1      colorspace_1.4-0    nnet_7.3-12
## [13] tidyselect_0.2.5    gridExtra_2.3       mnormt_1.5-5        compiler_3.5.1
## [17] qgraph_1.6.1        fdrtool_1.2.15      htmlTable_1.13.1    scales_1.0.0
## [21] checkmate_1.9.1     mvtnorm_1.0-9       psych_1.8.12        pbapply_1.4-0
## [25] sem_3.1-9           stringr_1.4.0       digest_0.6.18       pbivnorm_0.6.0
## [29] foreign_0.8-71      minqa_1.2.4         rmarkdown_1.11      base64enc_0.1-3
## [33] jpeg_0.1-8          pkgconfig_2.0.2     htmltools_0.3.6     lme4_1.1-20
## [37] lisrelToR_0.1.4     htmlwidgets_1.3     rlang_0.3.1         rstudioapi_0.9.0
## [41] huge_1.3.0          gtools_3.8.1        acepack_1.4.1       dplyr_0.8.0.1
## [45] zip_2.0.0           magrittr_1.5        Formula_1.2-3       Matrix_1.2-15
## [49] Rcpp_1.0.0          munsell_0.5.0       abind_1.4-5         rockchalk_1.8.140
## [53] whisker_0.3-2       stringi_1.3.1       yaml_2.2.0          carData_3.0-2
## [57] MASS_7.3-51.1       plyr_1.8.4          matrixcalc_1.0-3    lavaan_0.6-3
## [61] grid_3.5.1          parallel_3.5.1      crayon_1.3.4        lattice_0.20-38
## [65] semPlot_1.1         kutils_1.64         splines_3.5.1       knitr_1.21
## [69] pillar_1.3.1        igraph_1.2.4        rjson_0.2.20        boot_1.3-20
## [73] corpcor_1.6.9       BDgraph_2.55        reshape2_1.4.3      stats4_3.5.1
## [77] XML_3.98-1.17       glue_1.3.0          evaluate_0.13       latticeExtra_0.6-28
## [81] data.table_1.12.0   png_0.1-7           nloptr_1.2.1        gtable_0.2.0
## [85] purrr_0.3.0         assertthat_0.2.0    ggplot2_3.1.0       xfun_0.5
## [89] openxlsx_4.1.0      xtable_1.8-3        semTools_0.5-1      coda_0.19-2
## [93] survival_2.43-3     glasso_1.10         tibble_2.0.1        arm_1.10-1
## [97] ggm_2.3             ellipse_0.4.1       cluster_2.0.7-1
```