

multi-group WLS

Pasco Fearon (modified by Mike Cheung)

15 Aug 2020

```
#MetaSEM twin modelling

#Main lavaan model and basic set up borrowed from...

#-----Twin modelling script-----#
#-----Chapter 4.6, Beaujean 2014 Latent Variable Modeling Using R-----#

# Environment set up
rm(list = ls()) # clear R
options(max.print=1000000) # set max print

# Load lavaan
library(lavaan)

## This is lavaan 0.6-6
## lavaan is BETA software! Please report any bugs.
library(metaSEM)

## Loading required package: OpenMx
## To take full advantage of multiple cores, use:
##   mxOption(key='Number of Threads', value=parallel::detectCores()) #now
##   Sys.setenv(OMP_NUM_THREADS=parallel::detectCores()) #before library(OpenMx)
##
## Attaching package: 'OpenMx'
## The following object is masked from 'package:lavaan':
##
##     vech
## "SLSQP" is set as the default optimizer in OpenMx.
## mxOption(NULL, "Gradient algorithm") is set at "central".
## mxOption(NULL, "Optimality tolerance") is set at "6.3e-14".
## mxOption(NULL, "Gradient iterations") is set at "2".

# MZ twins
MZ <- lav_matrix_lower2full(c(1, 0.7735, 1))
rownames(MZ) <- colnames(MZ) <- c("P1", "P2")
# DZ twins
DZ <- lav_matrix_lower2full(c(1, 0.3871, 1))
rownames(DZ) <- colnames(DZ) <- c("P1", "P2")
```

```

# MZ twins asymp cov
MZacov <- 0.00625681

# DZ twins asymp cov
DZacov <- 0.00311364

# combine the covariances, sample sizes, asypt covs
twin.cov <- list(MZ = MZ, DZ = DZ)
twin.acov <- list(MZ = MZacov, DZ = DZacov)
twin.n <- list(MZ = 8308, DZ = 11983)

ace.model<-"
A1=~ NA*P1 + c(a,a)*P1
A2=~ NA*P2 + c(a,a)*P2
C1 =~ NA*P1 + c(c,c)*P1
C2 =~ NA*P2 + c(c,c)*P2
# variances
A1 ~~ 1*A1
A2 ~~ 1*A2
C1 ~~ 1*C1
C2 ~~ 1*C2
P1~~c(e2,e2)*P1
P2~~c(e2,e2)*P2
# covariances
A1 ~~ c(1,.5)*A2
A1 ~~ 0*C1 + 0*C2
A2 ~~ 0*C1 + 0*C2
C1 ~~ c(1,1)*C2"

#standard ace model ignoring heterogeneity
ace.fit<-cfa(ace.model, sample.cov=twin.cov, sample.nobs=twin.n)
summary(ace.fit, standardized = TRUE, ci = TRUE)

```

```

## lavaan 0.6-6 ended normally after 26 iterations
##
##      Estimator                      ML
##      Optimization method          NLMINB
##      Number of free parameters      12
##      Number of equality constraints    9
##
##      Number of observations per group:
##      MZ                             8308
##      DZ                             11983
##
## Model Test User Model:
##
##      Test statistic                  0.000
##      Degrees of freedom              3
##      P-value (Chi-square)           1.000
##      Test statistic for each group:
##      MZ                             0.000
##      DZ                             0.000
##

```

```

## Parameter Estimates:
##
## Standard errors          Standard
## Information              Expected
## Information saturated (h1) model Structured
##
##
## Group 1 [MZ]:
##
## Latent Variables:
##      Estimate  Std.Err  z-value  P(>|z|)  ci.lower  ci.upper
##  A1 =~
##    P1      (a)    0.879    0.010   92.261    0.000    0.860    0.898
##  A2 =~
##    P2      (a)    0.879    0.010   92.261    0.000    0.860    0.898
##  C1 =~
##    P1      (c)    0.026    0.290    0.091    0.928   -0.542    0.594
##  C2 =~
##    P2      (c)    0.026    0.290    0.091    0.928   -0.542    0.594
##  Std.lv  Std.all
##
##    0.879    0.879
##
##    0.879    0.879
##
##    0.026    0.026
##
##    0.026    0.026
##
## Covariances:
##      Estimate  Std.Err  z-value  P(>|z|)  ci.lower  ci.upper
##  A1 ~~
##    A2          1.000          1.000    1.000
##    C1          0.000          0.000    0.000
##    C2          0.000          0.000    0.000
##  A2 ~~
##    C1          0.000          0.000    0.000
##    C2          0.000          0.000    0.000
##  C1 ~~
##    C2          1.000          1.000    1.000
##  Std.lv  Std.all
##
##    1.000    1.000
##    0.000    0.000
##    0.000    0.000
##
##    0.000    0.000
##    0.000    0.000
##
##    1.000    1.000
##
## Variances:
##      Estimate  Std.Err  z-value  P(>|z|)  ci.lower  ci.upper
##    A1          1.000          1.000    1.000

```

```

##      A2              1.000              1.000      1.000
##      C1              1.000              1.000      1.000
##      C2              1.000              1.000      1.000
##      .P1      (e2)    0.226      0.003    64.799      0.000    0.220    0.233
##      .P2      (e2)    0.226      0.003    64.799      0.000    0.220    0.233
##      Std.lv  Std.all
##      1.000      1.000
##      1.000      1.000
##      1.000      1.000
##      1.000      1.000
##      0.226      0.226
##      0.226      0.226
##
##
## Group 2 [DZ]:
##
## Latent Variables:
##      Estimate  Std.Err  z-value  P(>|z|)  ci.lower  ci.upper
##      A1 =~
##      P1      (a)      0.879      0.010    92.261      0.000      0.860      0.898
##      A2 =~
##      P2      (a)      0.879      0.010    92.261      0.000      0.860      0.898
##      C1 =~
##      P1      (c)      0.026      0.290      0.091      0.928     -0.542      0.594
##      C2 =~
##      P2      (c)      0.026      0.290      0.091      0.928     -0.542      0.594
##      Std.lv  Std.all
##
##      0.879      0.879
##
##      0.879      0.879
##
##      0.026      0.026
##
##      0.026      0.026
##
## Covariances:
##      Estimate  Std.Err  z-value  P(>|z|)  ci.lower  ci.upper
##      A1 ~~
##      A2              0.500              0.500      0.500
##      C1              0.000              0.000      0.000
##      C2              0.000              0.000      0.000
##      A2 ~~
##      C1              0.000              0.000      0.000
##      C2              0.000              0.000      0.000
##      C1 ~~
##      C2              1.000              1.000      1.000
##      Std.lv  Std.all
##
##      0.500      0.500
##      0.000      0.000
##      0.000      0.000
##
##      0.000      0.000

```

```
##      0.000      0.000
##
##      1.000      1.000
##
## Variances:
##              Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
##      A1              1.000
##      A2              1.000
##      C1              1.000
##      C2              1.000
##      .P1      (e2)    0.226    0.003   64.799    0.000    0.220    0.233
##      .P2      (e2)    0.226    0.003   64.799    0.000    0.220    0.233
##      Std.lv Std.all
##      1.000    1.000
##      1.000    1.000
##      1.000    1.000
##      1.000    1.000
##      0.226    0.226
##      0.226    0.226
```

```
standardizedsolution(ace.fit, ci = TRUE)
```

```
##      lhs op rhs group est.std      se      z pvalue ci.lower ci.upper
##  1  A1 =~ P1      1  0.879 0.009 95.748 0.000    0.861    0.897
##  2  A2 =~ P2      1  0.879 0.009 95.748 0.000    0.861    0.897
##  3  C1 =~ P1      1  0.026 0.290  0.091 0.928   -0.542    0.594
##  4  C2 =~ P2      1  0.026 0.290  0.091 0.928   -0.542    0.594
##  5  A1 ~~ A1      1  1.000 0.000      NA      NA    1.000    1.000
##  6  A2 ~~ A2      1  1.000 0.000      NA      NA    1.000    1.000
##  7  C1 ~~ C1      1  1.000 0.000      NA      NA    1.000    1.000
##  8  C2 ~~ C2      1  1.000 0.000      NA      NA    1.000    1.000
##  9  P1 ~~ P1      1  0.226 0.004 59.336 0.000    0.219    0.234
## 10  P2 ~~ P2      1  0.226 0.004 59.336 0.000    0.219    0.234
## 11  A1 ~~ A2      1  1.000 0.000      NA      NA    1.000    1.000
## 12  A1 ~~ C1      1  0.000 0.000      NA      NA    0.000    0.000
## 13  A1 ~~ C2      1  0.000 0.000      NA      NA    0.000    0.000
## 14  A2 ~~ C1      1  0.000 0.000      NA      NA    0.000    0.000
## 15  A2 ~~ C2      1  0.000 0.000      NA      NA    0.000    0.000
## 16  C1 ~~ C2      1  1.000 0.000      NA      NA    1.000    1.000
## 17  A1 =~ P1      2  0.879 0.009 95.748 0.000    0.861    0.897
## 18  A2 =~ P2      2  0.879 0.009 95.748 0.000    0.861    0.897
## 19  C1 =~ P1      2  0.026 0.290  0.091 0.928   -0.542    0.594
## 20  C2 =~ P2      2  0.026 0.290  0.091 0.928   -0.542    0.594
## 21  A1 ~~ A1      2  1.000 0.000      NA      NA    1.000    1.000
## 22  A2 ~~ A2      2  1.000 0.000      NA      NA    1.000    1.000
## 23  C1 ~~ C1      2  1.000 0.000      NA      NA    1.000    1.000
## 24  C2 ~~ C2      2  1.000 0.000      NA      NA    1.000    1.000
## 25  P1 ~~ P1      2  0.226 0.004 59.336 0.000    0.219    0.234
## 26  P2 ~~ P2      2  0.226 0.004 59.336 0.000    0.219    0.234
## 27  A1 ~~ A2      2  0.500 0.000      NA      NA    0.500    0.500
## 28  A1 ~~ C1      2  0.000 0.000      NA      NA    0.000    0.000
## 29  A1 ~~ C2      2  0.000 0.000      NA      NA    0.000    0.000
## 30  A2 ~~ C1      2  0.000 0.000      NA      NA    0.000    0.000
## 31  A2 ~~ C2      2  0.000 0.000      NA      NA    0.000    0.000
## 32  C1 ~~ C2      2  1.000 0.000      NA      NA    1.000    1.000
```

```
# metaSEM model
```

```
RAM1 <- lavaan2RAM(ace.model, obs.variables=c('P1', 'P2'), ngroups=2)
```

```
RAM1
```

```
## $`1`
## $`1`$A
##      P1 P2 A1  A2  C1  C2
## P1 "0" "0" "0*a" "0"  "0*c" "0"
## P2 "0" "0" "0"  "0*a" "0"  "0*c"
## A1 "0" "0" "0"  "0"  "0"  "0"
## A2 "0" "0" "0"  "0"  "0"  "0"
## C1 "0" "0" "0"  "0"  "0"  "0"
## C2 "0" "0" "0"  "0"  "0"  "0"
##
## $`1`$S
##      P1      P2      A1 A2 C1 C2
## P1 "0*e2" "0"  "0" "0" "0" "0"
## P2 "0"    "0*e2" "0" "0" "0" "0"
## A1 "0"    "0"    "1" "1" "0" "0"
## A2 "0"    "0"    "1" "1" "0" "0"
## C1 "0"    "0"    "0" "0" "1" "1"
## C2 "0"    "0"    "0" "0" "1" "1"
##
## $`1`$F
##      P1 P2 A1 A2 C1 C2
## P1  1  0  0  0  0  0
## P2  0  1  0  0  0  0
##
## $`1`$M
##      P1 P2 A1 A2 C1 C2
## 1  0  0  0  0  0  0
##
##
## $`2`
## $`2`$A
##      P1 P2 A1  A2  C1  C2
## P1 "0" "0" "0*a" "0"  "0*c" "0"
## P2 "0" "0" "0"  "0*a" "0"  "0*c"
## A1 "0" "0" "0"  "0"  "0"  "0"
## A2 "0" "0" "0"  "0"  "0"  "0"
## C1 "0" "0" "0"  "0"  "0"  "0"
## C2 "0" "0" "0"  "0"  "0"  "0"
##
## $`2`$S
##      P1      P2      A1  A2  C1 C2
## P1 "0*e2" "0"  "0"  "0"  "0" "0"
## P2 "0"    "0*e2" "0"  "0"  "0" "0"
## A1 "0"    "0"    "1"  "0.5" "0" "0"
## A2 "0"    "0"    "0.5" "1"  "0" "0"
## C1 "0"    "0"    "0"  "0"  "1" "1"
## C2 "0"    "0"    "0"  "0"  "1" "1"
##
## $`2`$F
```

```

##      P1 P2 A1 A2 C1 C2
## P1   1  0  0  0  0  0
## P2   0  1  0  0  0  0
##
## $`2`$M
##      P1 P2 A1 A2 C1 C2
## 1     0  0  0  0  0  0

## ACOV is a list of matrices
twin.acov <- lapply(twin.acov, matrix)

## Create models without running them
gp1 <- wls(model.name="gp1", Cov = twin.cov[[1]],
            aCov = twin.acov[[1]], n = twin.n[1], RAM = RAM1[[1]],
            cor.analysis = TRUE, run=FALSE)

gp2 <- wls(model.name="gp2", Cov = twin.cov[[2]],
            aCov = twin.acov[[2]], n = twin.n[2], RAM = RAM1[[2]],
            cor.analysis = TRUE, run=FALSE)

## Combine the models
ace <- mxModel("All", gp1, gp2,
               mxFitFunctionMultigroup(c("gp1.fitfunction", "gp2.fitfunction")))
fit1 <- mxRun(ace)

## Running All with 2 parameters
summary(fit1)

## Summary of All
##
## free parameters:
##   name      matrix row col  Estimate Std.Error A
## 1    a gp1.Amatrix  P1  A1 0.87909044  0.110115
## 2    c gp1.Amatrix  P1  C1 0.02645751  2.585079
##
## Model Statistics:
##           | Parameters | Degrees of Freedom | Fit (-2lnL units)
##      Model:              2              -2      4.127618e-20
## Saturated:              NA              NA              NA
## Independence:          NA              NA              NA
## Number of observations/statistics: 0/0
##
## Information Criteria:
##           | df Penalty | Parameters Penalty | Sample-Size Adjusted
## AIC:              NA              NA              NA
## BIC:              NA              NA              NA
## CFI: NA
## TLI: 1 (also known as NNFI)
## RMSEA: 0 [95% CI (NA, NA)]
## Prob(RMSEA <= 0.05): NA
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2020-08-15 17:19:02
## Wall clock time: 0.1115544 secs
## optimizer: SLSQP
## OpenMx version number: 2.17.4

```

```
## Need help? See help(mxSummary)
```

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.1 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_SG.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_SG.UTF-8      LC_COLLATE=en_SG.UTF-8
##  [5] LC_MONETARY=en_SG.UTF-8  LC_MESSAGES=en_SG.UTF-8
##  [7] LC_PAPER=en_SG.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_SG.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] metaSEM_1.2.4.1 OpenMx_2.17.4  lavaan_0.6-6
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.5      mvtnorm_1.1-1   lattice_0.20-41 digest_0.6.25
##  [5] MASS_7.3-51.6  grid_4.0.2      stats4_4.0.2   magrittr_1.5
##  [9] ellipse_0.4.2  evaluate_0.14   rlang_0.4.7    stringi_1.4.6
## [13] Matrix_1.2-18  pbivnorm_0.6.0  rmarkdown_2.3  tools_4.0.2
## [17] stringr_1.4.0  parallel_4.0.2  xfun_0.13      yaml_2.2.1
## [21] compiler_4.0.2 mnormt_2.0.1    tmvnsim_1.0-2  htmltools_0.4.0
## [25] knitr_1.28
```