Overall analyses

Melnyk et al

september 12, 2018

Contents

```
library(metaSEM)

fulldata <- read.csv('dataset revision.txt', sep = "\t")

fulldata[1:10,1:30]
```

```
##    Rec Dependent Sample Year Paper Nfinal IN_A IN_I IN_B PBC_IN DN_A DN_I
## 1    2         1      2 2006     2   1250   NA 0.11   NA     NA   NA   NA
## 2    3         1      2 2006     2   1250   NA 0.21   NA     NA   NA   NA
## 3    4         1      2 2006     2   1250   NA 0.27   NA     NA   NA   NA
## 4    5         2      3 2003     3    707   NA 0.42   NA     NA   NA   NA
## 5    6         3      4 2004     4    205   NA 0.37   NA     NA   NA   NA
## 6    7         4      5 2005     5    429   NA 0.39 0.35     NA   NA   NA
## 7    9         5      7 1997     7    143 0.65 0.68   NA   0.17   NA   NA
## 8   10         6      8 1998     8    178   NA 0.38 0.27     NA   NA   NA
## 9   11         6      8 1998     8    178   NA 0.57 0.31     NA   NA   NA
## 10  12         6      8 1998     8    178   NA 0.41 0.32     NA   NA   NA
##    DN_B PBC_DN IN_DN  A_I  A_B  I_B PBC_A PBC_I PBC_B S_Beh_s S_San S_Rew
## 1    NA     NA    NA 0.36   NA   NA    NA    NA    NA       0     0     0
## 2    NA     NA    NA 0.36   NA   NA    NA    NA    NA       0     0     0
## 3    NA     NA    NA 0.36   NA   NA    NA    NA    NA       0     0     0
## 4    NA     NA    NA 0.76   NA   NA    NA  0.57    NA       1     0     0
## 5    NA     NA    NA 0.57   NA 0.59    NA  0.33    NA       0     0     0
## 6    NA     NA    NA 0.49 0.40 0.64    NA  0.50  0.48       0     0     1
## 7    NA     NA    NA 0.80   NA   NA  0.25  0.25    NA       0     0     0
## 8    NA     NA    NA 0.70 0.51 0.59    NA  0.08  0.17       0     0     0
## 9    NA     NA    NA 0.70 0.51 0.59    NA  0.08  0.17       0     0     0
## 10   NA     NA    NA 0.70 0.51 0.59    NA  0.08  0.17       0     0     0
##    S_Personal studentsample culture appli_area source gender_group
## 1           1             0       2          4      3            2
## 2           1             0       2          4      3            2
## 3           1             0       2          4      3            2
## 4           1             1       1          3      3            1
## 5           1             0       1          3      3            1
## 6           1             0       1          1      2            2
## 7           1             1      NA          3      3            1
## 8           1             0       1          3      3            2
## 9           1             0       1          3      3            2
## 10          1             0       1          3      3            2
```

```
ztrans = function(r){
        z = .5*log((1+r)/(1-r))
        return(z)}

rtrans = function(z){
```

```
        r = (exp(2*z)-1)/(exp(2*z)+1)
        return(r)}


indep = unique(fulldata$Dependent)

z = sapply(fulldata[,c(7:21)],ztrans)

meanz = aggregate(z, by=list(fulldata$Dependent), FUN=mean, na.rm=TRUE)

meanr = as.data.frame(sapply(meanz[,2:16],rtrans))

# change NaN to NA
is.nan.data.frame <- function(x)
do.call(cbind, lapply(x, is.nan))
meanr[is.nan(meanr)]=NA

meanN = aggregate(fulldata$Nfinal, by = list(fulldata$Dependent), FUN = mean)
meanN = meanN[,-1]


########################################
## AVERAGED across products #####
########################################

nvar = 6
nsample = nrow(meanr)
varnames = c("IN","DN","A","PBC","I","B")

# make list of cormatrices

    cormatrices = list()
    for (i in 1:nrow(meanr)){

        X = matrix(0,nvar,nvar)
        val =
        c(1,
        meanr$IN_DN[i], 1,
        meanr$IN_A[i], meanr$DN_A[i], 1,
        meanr$PBC_IN[i], meanr$PBC_DN[i], meanr$PBC_A[i],1,
        meanr$IN_I[i],meanr$DN_I[i], meanr$A_I[i], meanr$PBC_I[i],1,
        meanr$IN_B[i], meanr$DN_B[i], meanr$A_B[i],meanr$PBC_B[i],meanr$I_B[i],1
        )

        X[upper.tri(X, diag=TRUE)] = val
            cormatrices[[i]] = X + t(X) - diag(diag(X))

        dimnames(cormatrices[[i]]) = list(varnames,varnames)

    }


    # put NA on diagonal if variable is missing
```

```
    for (i in 1:length(cormatrices)){
        for (j in 1:nrow(cormatrices[[i]])){
            if (sum(is.na(cormatrices[[i]][j,]))==(nvar-1)) {
            cormatrices[[i]][j,j] = NA }
            }
        }


    ## delete variables?
    varpresent = matrix(0,nsample,nvar)
    colnames(varpresent)=varnames



### put NA on diagonal for variable missing correlation with least present correlations


for (i in 1:length(cormatrices)){
    for (j in c(2,1,6,5,3,4)){
            for (k in c(2,1,6,5,3,4)){
            if ( is.na(cormatrices[[i]][j,k]==TRUE)&is.na(cormatrices[[i]][j,j])!=TRUE
                 &is.na(cormatrices[[i]][k,k])!=TRUE){
                if (sum(is.na(cormatrices[[i]])[j,])>sum(is.na(cormatrices[[i]])[k,]))
                    { cormatrices[[i]][k,k] = NA
                      cormatrices[[i]][k,] = NA
                      cormatrices[[i]][,k] = NA  }
                if (sum(is.na(cormatrices[[i]])[j,])<=sum(is.na(cormatrices[[i]])[k,]))
                    { cormatrices[[i]][j,j] = NA
                      cormatrices[[i]][j,] = NA
                      cormatrices[[i]][,j] = NA  }
                 }
            }
    }
varpresent[i,]=diag(cormatrices[[i]])
}

varpresent[is.na(varpresent)]=FALSE


## make list of smaller correlationmatrices (cormat)

cormat = list()
    for (i in 1:length(cormatrices)){
    cormat[[i]]=cormatrices[[i]][varpresent[i,]==1,varpresent[i,]==1]
    }



#### stage 1 RANDOM with MASEM

#stage1random_overall = tssem1(my.df = cormatrices, n = meanN, method = "REM", RE.type = "Diag")
#save(stage1random_overall, file = 'stage1random_overall.R')
```

```r
load('stage1random_overall.R')
rsum = summary(stage1random_overall)

round(coef(stage1random_overall),2)
```

```
##  Intercept1  Intercept2  Intercept3  Intercept4  Intercept5  Intercept6
##        0.38        0.37        0.25        0.39        0.22        0.31
##  Intercept7  Intercept8  Intercept9 Intercept10 Intercept11 Intercept12
##        0.23        0.42        0.35        0.35        0.54        0.34
## Intercept13 Intercept14 Intercept15    Tau2_1_1    Tau2_2_2    Tau2_3_3
##        0.43        0.31        0.51        0.03        0.02        0.03
##    Tau2_4_4    Tau2_5_5    Tau2_6_6    Tau2_7_7    Tau2_8_8    Tau2_9_9
##        0.02        0.02        0.01        0.04        0.02        0.02
##  Tau2_10_10  Tau2_11_11  Tau2_12_12  Tau2_13_13  Tau2_14_14  Tau2_15_15
##        0.04        0.02        0.02        0.05        0.04        0.03
```

```r
est_fixed <- coef(stage1random_overall, select="fixed")

pooledRandom <- vec2symMat(est_fixed, diag=FALSE)
dimnames(pooledRandom) = list(varnames,varnames)
round(pooledRandom,2)
```

```
##       IN   DN    A  PBC    I    B
## IN  1.00 0.38 0.37 0.25 0.39 0.22
## DN  0.38 1.00 0.31 0.23 0.42 0.35
## A   0.37 0.31 1.00 0.35 0.54 0.34
## PBC 0.25 0.23 0.35 1.00 0.43 0.31
## I   0.39 0.42 0.54 0.43 1.00 0.51
## B   0.22 0.35 0.34 0.31 0.51 1.00
```

```r
### Stage 2 with WLS

Rpop = pooledRandom
noffd = nvar*(nvar-1)/2
acovR = 2 * solve(stage1random_overall$mx.fit@output$calculatedHessian)[1:noffd,1:noffd]
N = stage1random_overall$total.n

data = mxData(observed = Rpop, type = 'cor', numObs = N)


matrixobsR =  mxMatrix(
            type = "Symm",
            nrow = nvar,
            ncol = nvar,
            free = FALSE,
        values = Rpop,
            byrow = TRUE,
            name = "obsR")

matrixinvAcov = mxMatrix(
        type = "Full",
        ncol = noffd,
        nrow = noffd,
        values = c(solve(acovR)),
            free = FALSE,
```

```
        name = "invAcov")


B =   mxMatrix(
        type = "Full",
        nrow = nvar,
        ncol = nvar,
        free = c(FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,
               FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,
               FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,
               FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,
               TRUE,TRUE,TRUE,TRUE,FALSE,FALSE,
               TRUE,TRUE,TRUE,TRUE,TRUE,FALSE),

        values = c(0,0,0,0,0,0,
                 0,0,0,0,0,0,
                 0,0,0,0,0,0,
                 0,0,0,0,0,0,
                 0.1,0.1,0.1,0.1,0,0,
                 0.1,0.1,0.1,0.1,0.1,0),

        labels = matrix(NA,6,6),

        byrow = TRUE,
        name = "B",
        dimnames = list(varnames,varnames))


P =   mxMatrix(
            type = "Symm",
            nrow = nvar,
            ncol = nvar,
            free = c(TRUE,
               TRUE,TRUE,
               TRUE,TRUE,TRUE,
               TRUE,TRUE,TRUE,TRUE,
               FALSE,FALSE,FALSE,FALSE,TRUE,
               FALSE,FALSE,FALSE,FALSE,FALSE,TRUE),

        values = c(1,
                 .1,1,
                 .1,.1,1,
                 .1,.1,.1,1,
                 0,0,0,0,.6,
                 0,0,0,0,0,.7),

        labels = c(NA,
                 NA,NA,
                 NA,NA,NA,
                 NA,NA,NA,NA,
                 NA,NA,NA,NA,NA,
                 NA,NA,NA,NA,NA,NA),
```

```
            byrow = TRUE,
            name = "P",
        dimnames = list(varnames,varnames))


Id =    mxMatrix(
            type = "Iden",
            nrow = nvar,
            ncol = nvar,
            name = "Id")

indirect_IN = mxAlgebra(expression = B[5,1] * B[6,5], name = "IN_indirect")
total_IN = mxAlgebra(expression = B[5,1] * B[6,5] + B[6,1], name = "IN_total")
total_DN = mxAlgebra(expression = B[5,2] * B[6,5] + B[6,2], name = "DN_total")
total_A = mxAlgebra(expression = B[5,3] * B[6,5] + B[6,3], name = "A_total")
total_PBC = mxAlgebra(expression = B[5,4] * B[6,5] + B[6,4], name = "PBC_total")
indirect_DN = mxAlgebra(expression = B[5,2] * B[6,5], name = "DN_indirect")
indirect_A = mxAlgebra(expression = B[5,3] * B[6,5], name = "A_indirect")
indirect_PBC = mxAlgebra(expression = B[5,4] * B[6,5], name = "PBC_indirect")

constraint = mxConstraint(diag2vec(impliedR) == diag2vec(Id), name = "km")

conf = mxCI(c("B","P","IN_indirect","DN_indirect","A_indirect","PBC_indirect",
              "IN_total","DN_total","A_total","PBC_total"), interval = .95)

impliedR = mxAlgebra(expression = solve(Id-B) %*% P %*% t(solve(Id-B)),
                     name = "impliedR", dimnames = list(varnames,varnames))

vecR = mxAlgebra(vechs(obsR - impliedR), name = "vecR")
obj = mxAlgebra(t(vecR) %&% invAcov, name = "obj")
objective = mxFitFunctionAlgebra("obj", numStats = 15 )

modelwls =  mxModel('step1wls',objective,obj,vecR,impliedR,Id,P,B,total_DN,total_A,
                    total_PBC,matrixinvAcov,matrixobsR,data,constraint,conf,total_IN,
                    indirect_IN,indirect_DN,indirect_A,indirect_PBC)

modelwlsout = mxTryHard(modelwls, intervals = TRUE)

wlssum=summary(modelwlsout,verbose=T)


round(modelwlsout$B@values,2)
```

```
##        IN   DN    A  PBC    I B
## IN   0.00 0.00 0.00 0.00 0.00 0
## DN   0.00 0.00 0.00 0.00 0.00 0
## A    0.00 0.00 0.00 0.00 0.00 0
## PBC  0.00 0.00 0.00 0.00 0.00 0
## I    0.13 0.21 0.35 0.22 0.00 0
## B   -0.04 0.17 0.06 0.10 0.38 0
```

```
round(modelwlsout$P@values,2)
```

```
##       IN   DN    A  PBC    I   B
## IN  1.00 0.38 0.37 0.25 0.00 0.0
```

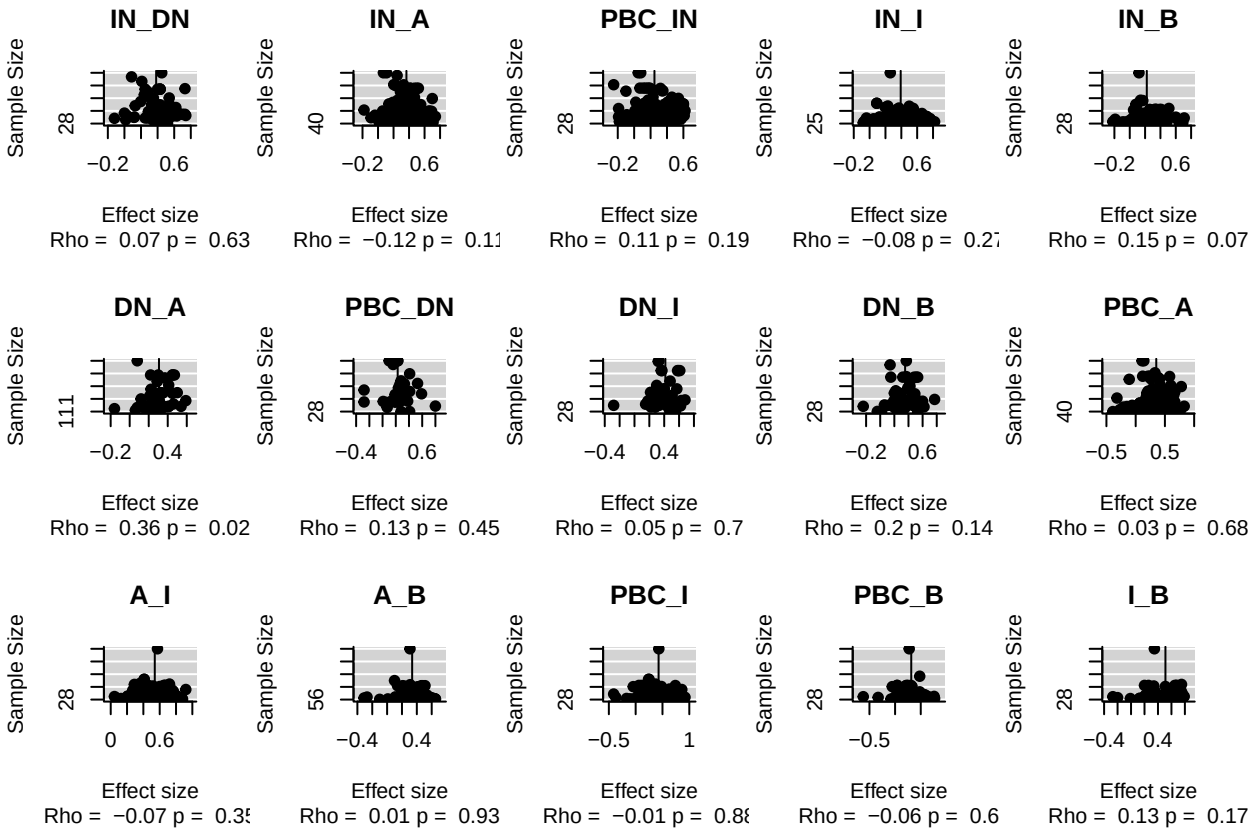
```
## DN  0.38 1.00 0.31 0.23 0.00 0.0
## A   0.37 0.31 1.00 0.35 0.00 0.0
## PBC 0.25 0.23 0.35 1.00 0.00 0.0
## I   0.00 0.00 0.00 0.00 0.58 0.0
## B   0.00 0.00 0.00 0.00 0.00 0.7
```

```
round(wlssum$CI[,1:3],2)
```

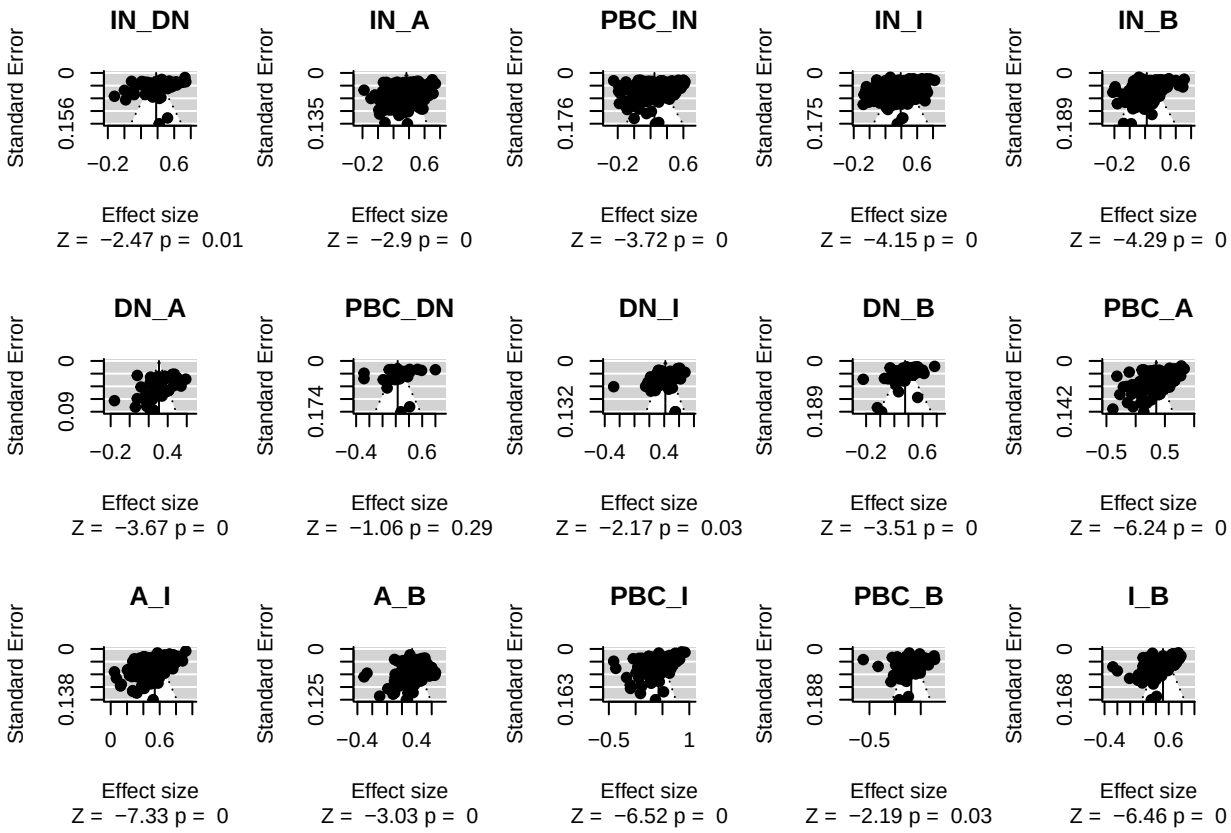
```
##                              lbound estimate ubound
## step1wls.B[5,1]                0.09     0.13   0.17
## step1wls.B[5,2]                0.15     0.21   0.26
## step1wls.B[5,3]                0.31     0.35   0.39
## step1wls.B[5,4]                0.18     0.22   0.27
## step1wls.B[6,1]               -0.08    -0.04   0.00
## step1wls.B[6,2]                0.10     0.17   0.23
## step1wls.B[6,3]                0.00     0.06   0.11
## step1wls.B[6,4]                0.04     0.10   0.16
## step1wls.B[6,5]                0.31     0.38   0.45
## step1wls.P[1,1]                  NA     1.00     NA
## step1wls.P[1,2]                0.33     0.38   0.43
## step1wls.P[1,3]                0.34     0.37   0.39
## step1wls.P[1,4]                0.22     0.25   0.28
## step1wls.P[2,2]                  NA     1.00     NA
## step1wls.P[2,3]                0.27     0.31   0.35
## step1wls.P[2,4]                0.16     0.23   0.30
## step1wls.P[3,3]                  NA     1.00     NA
## step1wls.P[3,4]                0.32     0.35   0.39
## step1wls.P[4,4]                  NA     1.00     NA
## step1wls.P[5,5]                0.55     0.58   0.61
## step1wls.P[6,6]                0.67     0.70   0.74
## step1wls.IN_indirect[1,1]      0.03     0.05   0.07
## step1wls.DN_indirect[1,1]      0.06     0.08   0.11
## step1wls.A_indirect[1,1]       0.11     0.13   0.16
## step1wls.PBC_indirect[1,1]     0.06     0.09   0.11
## step1wls.IN_total[1,1]        -0.03     0.01   0.05
## step1wls.DN_total[1,1]         0.19     0.25   0.30
## step1wls.A_total[1,1]          0.14     0.19   0.24
## step1wls.PBC_total[1,1]        0.13     0.19   0.25
```

```
## funnel plots

library(metafor)

cornames <- c("IN_DN","IN_A","PBC_IN","IN_I","IN_B",
              "DN_A","PBC_DN","DN_I","DN_B",
              "PBC_A","A_I","A_B","PBC_I","PBC_B","I_B")

ES <- meanr[,cornames]
n <- meanN
V <- (1-ES^2)^2 / n

pop <- stage1random_overall$mx.fit$Inter$values
```

```
### set up 5x3 array for plotting
par(mfrow=c(3,5))


for (i in 1:15){
  ### fit random-effects model
  res <- rma(ES[,i], V[,i], method="ML", ni=n)
  ### draw funnel plots
  title <- cornames[i]
  sub <- paste("Rho = ",round(cor.test(ES[,i],n, method = "spearman")[[4]],2),
               "p = ",round(cor.test(ES[,i],n, method = "spearman")[[3]],2))
  funnel(res, main=title, sub = sub, yaxis="ni", xlab ="Effect size", refline = pop[i])
}
```




```
for (i in 1:15){
  ### fit random-effects model
  res <- rma(ES[,i], V[,i], method="ML")
  ### draw funnel plots
  title <- cornames[i]
  sub <- paste("Z = ",round(regtest(res)[[3]],2),"p = ",round(regtest(res)[[4]],2))
  funnel(res, main=title, sub = sub, yaxis="sei", xlab ="Effect size", refline = pop[i])
}
```



Settings and versions of the R system and packages

```
sessionInfo()
```

```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Dutch_Netherlands.1252  LC_CTYPE=Dutch_Netherlands.1252
## [3] LC_MONETARY=Dutch_Netherlands.1252 LC_NUMERIC=C
## [5] LC_TIME=Dutch_Netherlands.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] metafor_2.0-0 Matrix_1.2-12 metaSEM_1.1.1 OpenMx_2.9.9
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.15    knitr_1.20      magrittr_1.5    MASS_7.3-49
##  [5] mnormt_1.5-5    pbivnorm_0.6.0  ellipse_0.4.1   lattice_0.20-35
##  [9] stringr_1.3.0   tools_3.4.4     parallel_3.4.4  grid_3.4.4
## [13] nlme_3.1-131.1  htmltools_0.3.6 yaml_2.1.18     rprojroot_1.3-2
## [17] digest_0.6.15   lavaan_0.6-2    evaluate_0.11   rmarkdown_1.9
```

```
## [21] stringi_1.1.6   compiler_3.4.4  backports_1.1.2 stats4_3.4.4
## [25] mvtnorm_1.0-8
```