CFA_Bifactor_NoCovariate

Contents

1 Load packages & set working directory & read in data

```
library(matrixcalc);library(MASS);library(Matrix)
```

```
## Warning:   'matrixcalc' R 4.3.1

## Warning:   'Matrix' R 4.3.1
```

```
library(coda);library(R2OpenBUGS);library(metaSEM)
```

```
## Warning:   'coda' R 4.3.1

## Warning:   'R2OpenBUGS' R 4.3.2

##      OpenMx

##
##    'OpenMx'

## The following objects are masked from 'package:Matrix':
##
##     %&%, expm

## The following object is masked from 'package:matrixcalc':
##
##     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".
```

```
# Working directory
wd = 'D:/Research/2023/CompareMASEM/CFA/Bifactor/NoCovariate/'
setwd(wd)
```

2 Functions

```r
# vector to matrix
v2m <- function(vec,p,corr= T){
    M = matrix(0,p,p)
    M[lower.tri(M)] = vec
    M = M + t(M)
    if(corr==TRUE){
        diag(M) = 1
    }else{
        diag(M) = diag(M)/2
    }
    return(M)
}

# impute missing values in covariance / correlation matrices of each study
# to obtain a rough estimate of the covariance matrix of covariance / correlation matrix
# weighted average correlation
Mimpute <- function(R,N,missing){
    if(is.null(missing)){
        return(R)
    }else{
        na.pos = which(is.na(R),arr.ind = TRUE)
        mu.N = mean(N)
        Rbar = apply(R,2,mean,na.rm = TRUE)# Becker's mean r

        for(coli in unique(na.pos[,2])){
            id = na.pos[(na.pos[,2] == coli),1]
            R[id,coli] = Rbar[coli]
        }
        return(R)
    }
}

# change the coordinating system of a vectorized matrix to the coordinating system of
# the original matrix
# e.g., from vS to S, the former uses one coordinate (vil), whereas the latter uses two (j,k).
Get.vi2jk <- function(p,diag.incl=FALSE,byrow=FALSE){
    A = matrix(1,p,p)
    if(diag.incl ==FALSE){
        pp = p*(p-1)/2
        vi2jk <- matrix(NA,pp,3)
        vi2jk[,3] <- 1:pp
        if(byrow == FALSE){
            vi2jk[,1:2] <- which(lower.tri(A)==1,arr.ind = TRUE)
        }else{
            vi2jk[,1:2] <- which(upper.tri(A)==1,arr.ind = TRUE)
        }
        colnames(vi2jk) = c('j','k','vi')
    }else{
        pp = p*(p+1)/2
        vi2jk <- matrix(NA,pp,3)
        vi2jk[,3] <- 1:pp
        if(byrow == FALSE){
```

```
            vi2jk[,1:2] <- which(lower.tri(A,diag = TRUE)==1,arr.ind = TRUE)
        }else{
            vi2jk[,1:2] <- which(upper.tri(A,diag = TRUE)==1,arr.ind = TRUE)
        }
        colnames(vi2jk) = c('j','k','vi')
    }
    return(vi2jk)
}

# change the coordinating system of a matrix to the coordinating system of
# the corresponding vectorized matrix
# e.g., from S to vS, the former uses two coordinates (j,k), whereas the latter uses only one (vil).
Get.jk2vi <- function(vi2jk,p,diag.incl=FALSE){
    jk2vi = matrix(0,p,p)
    jk2vi[vi2jk[,1:2]] = vi2jk[,3]
    if(diag.incl){
        jk2vi = jk2vi + t(jk2vi)
        diag(jk2vi) = diag(jk2vi)/2
    }else{
        pp = p*(p-1)/2
        jk2vi = jk2vi + t(jk2vi) + diag(rep(pp+1,p))
    }
    return(jk2vi)
}

jkvil <- function(p){
    vi2jk   = Get.vi2jk(p)
    j   = vi2jk[,1]
    k   = vi2jk[,2]
    vil = Get.jk2vi(vi2jk,p)
    return(list(j=j,k=k,vil=vil))
}

# compute the covariance matrix of correlation matrix
# based on Steiger (1980)
Corr.Cov <- function(vR,N,index.list){
    nvR = length(vR)
    vR  = c(vR,1)
    NvR.cov = matrix(NA,nvR,nvR)
    j = index.list$j
    k = index.list$k
    vil = index.list$vil

    for(vi in 1:nvR){
        NvR.cov[vi,vi] = (1-(vR[vi])^2)^2
    }
    for(vi in 1:(nvR-1)){
    for(vj in (vi+1):nvR){
        NvR.cov[vi,vj] = ((vR[vil[j[vi],j[vj]]]-vR[vi]*vR[vil[k[vi],j[vj]]])*(vR[vil[k[vi],k[vj]]]-vR[v
         +(vR[vil[j[vi],k[vj]]]-vR[vil[j[vi],j[vj]]]*vR[vj])*(vR[vil[k[vi],j[vj]]]-vR[vi]*vR[vil[j[vi],
         +(vR[vil[j[vi],j[vj]]]-vR[vil[j[vi],k[vj]]]*vR[vj])*(vR[vil[k[vi],k[vj]]]-vR[vi]*vR[vil[j[vi],
         +(vR[vil[j[vi],k[vj]]]-vR[vi]*vR[vil[k[vi],k[vj]]])*(vR[vil[j[vj],k[vi]]]-vR[vil[k[vi],k[vj]]]
        NvR.cov[vj,vi] <- NvR.cov[vi,vj]
```

```
    }
    }

    vR.cov = NvR.cov/(N)
    vR.cov = as.matrix(nearPD(vR.cov,posd.tol = 1e-5)$mat)
    return(vR.cov)
}

# Use average correlation vector to compute V_psi
Vj <- function(vR.bar,N,pp,Nstudy,index.list){

    mu.N = mean(N)
    S.vR.bar = Corr.Cov(vR.bar,mu.N,index.list)
    inv.S.vR.bar = solve(S.vR.bar)
    tau.vR = array(NA,dim = c(Nstudy,pp,pp))
    S.vR = array(NA,dim = c(Nstudy,pp,pp))
    for(i in 1:Nstudy){
        S.vR[i,,]<- S.vR.bar/N[i]*mu.N
        tau.vR[i,,] <- inv.S.vR.bar/mu.N*N[i]
    }
    return(list(S.vR = S.vR,tau.vR = tau.vR))
}

# Use individual correlation vectors to compute V_psi
Vj2 <- function(vR.impute,N,pp,Nstudy,index.list){

    tau.vR = array(NA,dim = c(Nstudy,pp,pp))
    S.vR = array(NA,dim = c(Nstudy,pp,pp))
    for(i in 1:Nstudy){
        S.vR[i,,] = Corr.Cov(vR.impute[i,],N[i],index.list)
        tau.vR[i,,] <- solve(S.vR[i,,])
    }
    return(list(S.vR = S.vR,tau.vR = tau.vR))
}

# generate data for meta-analytic CFA
# the two-level model of OSMASEM is used
Gen.CFA.data <- function(Nstudy,mu.N,Model.list,p,missing,N=NULL){

    beta = Model.list$beta
    tau = Model.list$tau
    ind = Model.list$ind
    Z = Model.list$Z
    pp = Model.list$pp
    j = Model.list$j
    j10 = Model.list$j10
    k = Model.list$k
    k10 = Model.list$k10
    vil = Model.list$vil

    # predicted SEM parameters
    coefM <- Z%*%t(beta)
```

```
    # predicted part of the true correlation vector for each study
    vPs = t(apply(coefM,1,function(x,pp,j,k,j10,k10,ind){
        r = rep(NA,pp)
        for(vi in 1:pp){
          r[vi] = x[j[vi]]*x[k[vi]]+x[j10[vi]]*x[k10[vi]]*ind[vi]
        }
        return(r)
    },pp=pp,j=j,k=k,j10=j10,k10=k10,ind=ind) )

    # true correlation vector for each study
    if(tau[1]>0){
       vP = t(apply(vPs,1,function(x,tau,pp){
        r = rep(NA,pp)
        for(vi in 1:pp){ r[vi] = rnorm(1,x[vi],sd=tau[vi]) }
        return(r)
       },tau=tau,pp=pp) )
    }else{ vP=vPs }

    # sample size for each study
    if(is.null(N)){
      N <- rzinb(n =Nstudy, k =0.8, lambda=round(mu.N*0.2), omega = 0)
      N <- N + round(mu.N*0.8)
    }

    # observed correlations
    vR = matrix(NA,Nstudy,pp)
    for(studyi in 1:Nstudy){
        Pm = v2m(vP[studyi,],p,T)
        Pm = nearPD(Pm,corr=T)$mat
        Ri = cor(mvrnorm(N[studyi],rep(0,p),Pm))
        vR[studyi,] = Ri[lower.tri(Ri)]
    }

    #source(paste(wd,'RealData.R',sep=''))
    #vR = Make.Missing2(vR,missing,miss.rate,N) # generate missing values
    return(list(j=j,k=k,vil=vil,pp=pp,N=N,vR=vR,Z=Z))
}

d4osmasem <- function(dsim){
    j = dsim$j
    vR = dsim$vR
    N = dsim$N
    Z = as.matrix(dsim$Z)

    p = max(j)
    R.l = as.list(as.data.frame(t(vR)))
    Mat = lapply(R.l,function(x,p) v2m(x,p,T),p=p)
    my.df = Cor2DataFrame(Mat,N,acov = 'weighted')
    my.df$data = data.frame(my.df$data,covariate=scale(Z[,1]),check.names = FALSE)
    return(my.df)
}

wbugs <-function(data,initsl,prm,mfn,
```

```
    nchains=1,niter=60000,nburnin=30000,nthin=1,wd,
    diagm){
# data: a named list of the data in the likelihood model for OpenBUGS
# initsl: a list with nchains elements; each element is a list of starting values
# prm: vector of names of the parameters to save
# mfn: the file name of the likelihood model for OpenBUGS
# diagm: name of the convergence diagnostic method; either 'Geweke' or 'Gelman'
# The function checks convergence every niter-nburnin iterations

    fit = bugs(data,initsl,prm,mfn,
        n.chains=nchains,n.iter=niter,n.burnin=nburnin,n.thin=1,
        debug=F,saveExec=T,working.directory = wd)

    for(tryi in 2:20){
        print(paste0('Iteration: ',tryi*(niter-nburnin)))
        fit.coda = read.openbugs(stem="",thin = nthin)
        del.id = na.omit(match(c('ppp'),varnames(fit.coda)))
        print(summary(fit.coda),3)
        if(diagm=='Geweke'){
            if(length(del.id)>0){
                tmp.conv = geweke.diag(fit.coda[,-del.id])[[1]]$z
            }else{ tmp.conv = geweke.diag(fit.coda)[[1]]$z }
            crit = (sum((abs(tmp.conv)>1.96),na.rm = T)==0)
        }else if(diagm=='Gelman'){
            if(length(del.id)>0){
                tmp.conv = gelman.diag(fit.coda)$psrf[-del.id,2]
            }else{ tmp.conv = gelman.diag(fit.coda)$psrf[,2] }
            crit = (sum((tmp.conv>1.1),na.rm = T)==0)
        }
        if(crit){
            print(tmp.conv)
            print(summary(fit.coda),3)
            break
        }else{
            fit = bugs(data,initsl,prm,mfn,
            n.chains=nchains,n.iter=niter-nburnin+1,n.burnin=1,n.thin=1,
            restart=T,saveExec=T,working.directory = wd)
        }
    }
    ppp.id = match('ppp',prm)
    sel = NA
    if(is.na(ppp.id)){
        nprm = length(prm)
        for(i in 1:nprm){
            sel = c(sel,grep(prm[i],rownames(summary(fit.coda)$quantiles)))
        }
    }else{
        prm = prm[-ppp.id]
        nprm = length(prm)
        for(i in 1:nprm){
            sel = c(sel,grep(prm[i],rownames(summary(fit.coda)$quantiles)))
        }
    }
```

```
    sel = sel[-1]
    sel = unique(sel)

    if(is.na(ppp.id)){ est = round(summary(fit.coda)$quantiles[sel,'50%'],3)
    }else{
        est = round(c(summary(fit.coda)$quantiles[sel,'50%'],
        summary(fit.coda)$statistics['ppp','Mean']),3)
    }
    psd = round(summary(fit.coda)$statistics[sel,'SD'],3)
    if(diagm=='Geweke'){
        CIl = round(HPDinterval(fit.coda,prob = .95)[[1]][sel,1],3)
        CIu = round(HPDinterval(fit.coda,prob = .95)[[1]][sel,2],3)
    }else if(diagm=='Gelman'){
        fit.coda.l = do.call(rbind,fit.coda)
        HPDCI = HPDinterval(mcmc(fit.coda.l),prob = .95)
        CIl = HPDCI[sel,1]
        CIu = HPDCI[sel,2]
    }
    sel.muL = grep('mu.L',names(est))
    sel.sdL = grep('sd.L',names(est))
    CVl = round(est[sel.muL] - 1.28*est[sel.sdL],3)
    CVu = round(est[sel.muL] + 1.28*est[sel.sdL],3)

    conv = round(c(tryi,tmp.conv),3)
    return(list(est=est,psd=psd,CIl=CIl,CIu=CIu,CVl=CVl,CVu=CVu,conv=conv,
        DIC=fit$DIC,fit.coda=fit.coda))
}
```

3 BMASEM

3.1 Data preparation

```
## Exclude studies that did not report bivariate correlations
index <- Gnambs18$CorMat==1
Gnambs18 <- lapply(Gnambs18, function(x) x[index])

# Convert correlation matrices to correlation vectors
mR = Gnambs18$data
vR = sapply(mR,function(x){ x = x[c(1,3,4,7,10,2,5,6,8,9),c(1,3,4,7,10,2,5,6,8,9)]
    return(x[lower.tri(x)]) })
vR = t(vR)

N      = Gnambs18$n # sample sizes within primary studies
mu.N   = mean(N) # mean sample size
Nstudy = length(Gnambs18$data) # the number of primary studies
Ninv   = 1/N # reciprocals of sample sizes

# Coordinates of correlation matrices and vectors
p  = 10 # number of variables
pp = p*(p-1)/2  # number of bivariate correlations
index.list = jkvil(p)
j = index.list$j
```

```r
k = index.list$k
vil = index.list$vil
j10 = j+10
k10 = k+10
# Do items load on the same factor? 1=No; 0 = Yes
ind = (j>(p+1)/2)*(k<(p+2)/2)

# Covariance matrices of sample correlation vectors
vR.bar = apply(vR,2,mean,na.rm = TRUE)
Stau.vR = Vj(vR.bar,N,pp,Nstudy,index.list)
tau.vR = Stau.vR$tau.vR

# information for the additional error term
mu.vR.psi = rep(0,pp)
df.prelim = 100*pp/mu.N+pp
alpha.prior.vE = (df.prelim-pp+1)/2
beta.prior.vE = alpha.prior.vE*(0.3/mu.N)

# Matrices for computing ppp
# Compute the between-study covariance matrix of true study-specific correlation vectors
# Z: First derivative of study-specific correlation vectors with respect to model
#    parameters (factor loadings)
# NA: for Openbugs to replace with parameter estimates
# The vi_th element in the vectorized correlation matrix corresponds to the
# correlation between the j_th and the k_th items.
# In the bifactor model, the correlation between the j_th and the_kth items
# equals the product of the j_th and the_kth
# factor loadings plus the product of the (j+10)_th and the (k+10)_th factor
# loadings (the factor loadings of the method factors) if the two items are
# loaded on the same method factor. Therefore, the first derivative of the vi_th
# correlation equals a nonzero value when the derivative is with respect to the
# j(+10)_th or the k(+10)_th factor loading and zero when it is with
# respect to other SEM parameters
Z <- matrix(0,pp,p*2)
for(vi in 1:pp){
    Z[vi,c(j[vi],k[vi])] = NA
    Z[vi,c(j[vi]+10,k[vi]+10)] = NA
}
# Diagonal covariance matrix of study-specific model parameters (factor loadings)
# Random factor loadings are assumed to be uncorrelated
V.theta = matrix(0,20,20)
diag(V.theta) = NA
```

3.2 Model fitting

```r
data<-list("Nstudy","N","Ninv","mu.N",'p',"pp","j","k","j10","k10",'ind',
    "vR","tau.vR",'Z','V.theta',"mu.vR.psi",'alpha.prior.vE','beta.prior.vE') # data

initsl <- list(list(mu.L=rep(.6,p*2),sd.L=rep(0.1,20),tau.R=mu.N*3,
    vR.psi=matrix(0,Nstudy,pp),vR.rep = vR))# Initial values


prm = c('mu.L','sd.L','ppp') # Parameters to save
```

```r
model.fn = paste(wd,'CFARandom.txt',sep='') #  model file name


# stop every 10000 iterations to check whether convergence is achieved
fit = wbugs(data,initsl,prm,model.fn,
        nchains=1,niter=60000,nburnin=30000,nthin=1,wd,diagm='Geweke')
```

```
## [1] "Iteration: 60000"
## Abstracting deviance ... 30000 valid values
## Abstracting mu.L[1] ... 30000 valid values
## Abstracting mu.L[2] ... 30000 valid values
## Abstracting mu.L[3] ... 30000 valid values
## Abstracting mu.L[4] ... 30000 valid values
## Abstracting mu.L[5] ... 30000 valid values
## Abstracting mu.L[6] ... 30000 valid values
## Abstracting mu.L[7] ... 30000 valid values
## Abstracting mu.L[8] ... 30000 valid values
## Abstracting mu.L[9] ... 30000 valid values
## Abstracting mu.L[10] ... 30000 valid values
## Abstracting mu.L[11] ... 30000 valid values
## Abstracting mu.L[12] ... 30000 valid values
## Abstracting mu.L[13] ... 30000 valid values
## Abstracting mu.L[14] ... 30000 valid values
## Abstracting mu.L[15] ... 30000 valid values
## Abstracting mu.L[16] ... 30000 valid values
## Abstracting mu.L[17] ... 30000 valid values
## Abstracting mu.L[18] ... 30000 valid values
## Abstracting mu.L[19] ... 30000 valid values
## Abstracting mu.L[20] ... 30000 valid values
## Abstracting ppp ... 30000 valid values
## Abstracting sd.L[1] ... 30000 valid values
## Abstracting sd.L[2] ... 30000 valid values
## Abstracting sd.L[3] ... 30000 valid values
## Abstracting sd.L[4] ... 30000 valid values
## Abstracting sd.L[5] ... 30000 valid values
## Abstracting sd.L[6] ... 30000 valid values
## Abstracting sd.L[7] ... 30000 valid values
## Abstracting sd.L[8] ... 30000 valid values
## Abstracting sd.L[9] ... 30000 valid values
## Abstracting sd.L[10] ... 30000 valid values
## Abstracting sd.L[11] ... 30000 valid values
## Abstracting sd.L[12] ... 30000 valid values
## Abstracting sd.L[13] ... 30000 valid values
## Abstracting sd.L[14] ... 30000 valid values
## Abstracting sd.L[15] ... 30000 valid values
## Abstracting sd.L[16] ... 30000 valid values
## Abstracting sd.L[17] ... 30000 valid values
## Abstracting sd.L[18] ... 30000 valid values
## Abstracting sd.L[19] ... 30000 valid values
## Abstracting sd.L[20] ... 30000 valid values
##
## Iterations = 30001:60000
## Thinning interval = 1
## Number of chains = 1
```

```
## Sample size per chain = 30000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean       SD Naive SE Time-series SE
## deviance -5.40e+03 58.66979 3.39e-01       0.661790
## mu.L[1]   7.50e-01  0.01922 1.11e-04       0.000149
## mu.L[2]   5.95e-01  0.01605 9.27e-05       0.000236
## mu.L[3]   5.22e-01  0.01594 9.20e-05       0.000180
## mu.L[4]   6.08e-01  0.02163 1.25e-04       0.000199
## mu.L[5]   8.01e-01  0.01208 6.98e-05       0.000145
## mu.L[6]   5.33e-01  0.01870 1.08e-04       0.000178
## mu.L[7]   5.17e-01  0.02356 1.36e-04       0.000187
## mu.L[8]   5.11e-01  0.01562 9.02e-05       0.000147
## mu.L[9]   3.77e-01  0.04289 2.48e-04       0.000283
## mu.L[10]  5.93e-01  0.01900 1.10e-04       0.000160
## mu.L[11] -5.20e-02  0.01895 1.09e-04       0.000527
## mu.L[12]  5.24e-01  0.02646 1.53e-04       0.000582
## mu.L[13]  3.07e-01  0.02131 1.23e-04       0.000323
## mu.L[14]  3.35e-01  0.03325 1.92e-04       0.000397
## mu.L[15] -2.99e-02  0.01602 9.25e-05       0.000539
## mu.L[16]  5.77e-01  0.01651 9.53e-05       0.000213
## mu.L[17]  3.32e-01  0.02122 1.23e-04       0.000231
## mu.L[18]  5.96e-01  0.01581 9.13e-05       0.000204
## mu.L[19]  3.66e-01  0.02243 1.29e-04       0.000230
## mu.L[20]  3.90e-01  0.02464 1.42e-04       0.000232
## ppp       7.26e-01  0.44612 2.58e-03       0.002748
## sd.L[1]   1.04e-01  0.01569 9.06e-05       0.000137
## sd.L[2]   7.88e-02  0.01229 7.09e-05       0.000117
## sd.L[3]   8.20e-02  0.01201 6.94e-05       0.000107
## sd.L[4]   1.18e-01  0.01673 9.66e-05       0.000137
## sd.L[5]   5.81e-02  0.00957 5.52e-05       0.000096
## sd.L[6]   9.99e-02  0.01460 8.43e-05       0.000127
## sd.L[7]   1.30e-01  0.01853 1.07e-04       0.000153
## sd.L[8]   7.97e-02  0.01280 7.39e-05       0.000133
## sd.L[9]   2.47e-01  0.03303 1.91e-04       0.000251
## sd.L[10]  1.02e-01  0.01487 8.59e-05       0.000131
## sd.L[11]  4.98e-02  0.01946 1.12e-04       0.000633
## sd.L[12]  6.78e-02  0.02622 1.51e-04       0.000755
## sd.L[13]  7.97e-02  0.01861 1.07e-04       0.000300
## sd.L[14]  1.57e-01  0.03121 1.80e-04       0.000425
## sd.L[15]  2.89e-02  0.01481 8.55e-05       0.000514
## sd.L[16]  6.38e-02  0.01490 8.61e-05       0.000232
## sd.L[17]  9.92e-02  0.01896 1.09e-04       0.000248
## sd.L[18]  5.89e-02  0.01677 9.68e-05       0.000332
## sd.L[19]  1.07e-01  0.01874 1.08e-04       0.000212
## sd.L[20]  1.22e-01  0.02210 1.28e-04       0.000259
##
## 2. Quantiles for each variable:
##
##                2.5%       25%       50%       75%     97.5%
## deviance -5.52e+03 -5.44e+03 -5.40e+03 -5.36e+03 -5.29e+03
## mu.L[1]   7.12e-01  7.37e-01  7.50e-01  7.62e-01  7.88e-01
```

```
## mu.L[2]   5.63e-01  5.84e-01  5.95e-01  6.05e-01  6.27e-01
## mu.L[3]   4.90e-01  5.11e-01  5.22e-01  5.32e-01  5.53e-01
## mu.L[4]   5.65e-01  5.93e-01  6.08e-01  6.22e-01  6.50e-01
## mu.L[5]   7.77e-01  7.93e-01  8.01e-01  8.09e-01  8.25e-01
## mu.L[6]   4.96e-01  5.20e-01  5.33e-01  5.45e-01  5.70e-01
## mu.L[7]   4.71e-01  5.02e-01  5.17e-01  5.33e-01  5.64e-01
## mu.L[8]   4.80e-01  5.01e-01  5.11e-01  5.22e-01  5.42e-01
## mu.L[9]   2.93e-01  3.48e-01  3.77e-01  4.05e-01  4.62e-01
## mu.L[10]  5.55e-01  5.81e-01  5.93e-01  6.06e-01  6.30e-01
## mu.L[11] -8.81e-02 -6.47e-02 -5.24e-02 -4.00e-02 -1.34e-02
## mu.L[12]  4.73e-01  5.06e-01  5.23e-01  5.41e-01  5.77e-01
## mu.L[13]  2.65e-01  2.93e-01  3.07e-01  3.21e-01  3.49e-01
## mu.L[14]  2.70e-01  3.13e-01  3.35e-01  3.57e-01  4.01e-01
## mu.L[15] -6.05e-02 -4.06e-02 -3.00e-02 -1.96e-02  2.33e-03
## mu.L[16]  5.46e-01  5.66e-01  5.77e-01  5.88e-01  6.10e-01
## mu.L[17]  2.91e-01  3.18e-01  3.32e-01  3.46e-01  3.75e-01
## mu.L[18]  5.65e-01  5.86e-01  5.96e-01  6.07e-01  6.28e-01
## mu.L[19]  3.22e-01  3.51e-01  3.66e-01  3.81e-01  4.11e-01
## mu.L[20]  3.41e-01  3.74e-01  3.90e-01  4.06e-01  4.39e-01
## ppp       0.00e+00  0.00e+00  1.00e+00  1.00e+00  1.00e+00
## sd.L[1]   7.71e-02  9.24e-02  1.02e-01  1.13e-01  1.38e-01
## sd.L[2]   5.84e-02  7.01e-02  7.77e-02  8.61e-02  1.06e-01
## sd.L[3]   6.20e-02  7.35e-02  8.08e-02  8.92e-02  1.09e-01
## sd.L[4]   9.06e-02  1.06e-01  1.17e-01  1.28e-01  1.56e-01
## sd.L[5]   4.21e-02  5.13e-02  5.72e-02  6.39e-02  7.95e-02
## sd.L[6]   7.53e-02  8.96e-02  9.85e-02  1.09e-01  1.32e-01
## sd.L[7]   9.92e-02  1.17e-01  1.28e-01  1.41e-01  1.71e-01
## sd.L[8]   5.85e-02  7.06e-02  7.85e-02  8.73e-02  1.08e-01
## sd.L[9]   1.91e-01  2.23e-01  2.43e-01  2.66e-01  3.21e-01
## sd.L[10]  7.66e-02  9.11e-02  1.00e-01  1.11e-01  1.35e-01
## sd.L[11]  1.22e-02  3.68e-02  4.89e-02  6.20e-02  9.04e-02
## sd.L[12]  2.21e-02  4.98e-02  6.55e-02  8.37e-02  1.26e-01
## sd.L[13]  4.86e-02  6.66e-02  7.78e-02  9.09e-02  1.22e-01
## sd.L[14]  1.05e-01  1.35e-01  1.54e-01  1.76e-01  2.28e-01
## sd.L[15]  5.49e-03  1.82e-02  2.69e-02  3.78e-02  6.26e-02
## sd.L[16]  3.84e-02  5.33e-02  6.25e-02  7.28e-02  9.64e-02
## sd.L[17]  6.76e-02  8.57e-02  9.73e-02  1.11e-01  1.42e-01
## sd.L[18]  3.02e-02  4.71e-02  5.74e-02  6.90e-02  9.60e-02
## sd.L[19]  7.59e-02  9.35e-02  1.05e-01  1.18e-01  1.49e-01
## sd.L[20]  8.46e-02  1.06e-01  1.20e-01  1.35e-01  1.71e-01
##
## [1] "Iteration: 90000"
## Abstracting deviance ... 30000 valid values
## Abstracting mu.L[1] ... 30000 valid values
## Abstracting mu.L[2] ... 30000 valid values
## Abstracting mu.L[3] ... 30000 valid values
## Abstracting mu.L[4] ... 30000 valid values
## Abstracting mu.L[5] ... 30000 valid values
## Abstracting mu.L[6] ... 30000 valid values
## Abstracting mu.L[7] ... 30000 valid values
## Abstracting mu.L[8] ... 30000 valid values
## Abstracting mu.L[9] ... 30000 valid values
## Abstracting mu.L[10] ... 30000 valid values
## Abstracting mu.L[11] ... 30000 valid values
```

```
## Abstracting mu.L[12] ... 30000 valid values
## Abstracting mu.L[13] ... 30000 valid values
## Abstracting mu.L[14] ... 30000 valid values
## Abstracting mu.L[15] ... 30000 valid values
## Abstracting mu.L[16] ... 30000 valid values
## Abstracting mu.L[17] ... 30000 valid values
## Abstracting mu.L[18] ... 30000 valid values
## Abstracting mu.L[19] ... 30000 valid values
## Abstracting mu.L[20] ... 30000 valid values
## Abstracting ppp ... 30000 valid values
## Abstracting sd.L[1] ... 30000 valid values
## Abstracting sd.L[2] ... 30000 valid values
## Abstracting sd.L[3] ... 30000 valid values
## Abstracting sd.L[4] ... 30000 valid values
## Abstracting sd.L[5] ... 30000 valid values
## Abstracting sd.L[6] ... 30000 valid values
## Abstracting sd.L[7] ... 30000 valid values
## Abstracting sd.L[8] ... 30000 valid values
## Abstracting sd.L[9] ... 30000 valid values
## Abstracting sd.L[10] ... 30000 valid values
## Abstracting sd.L[11] ... 30000 valid values
## Abstracting sd.L[12] ... 30000 valid values
## Abstracting sd.L[13] ... 30000 valid values
## Abstracting sd.L[14] ... 30000 valid values
## Abstracting sd.L[15] ... 30000 valid values
## Abstracting sd.L[16] ... 30000 valid values
## Abstracting sd.L[17] ... 30000 valid values
## Abstracting sd.L[18] ... 30000 valid values
## Abstracting sd.L[19] ... 30000 valid values
## Abstracting sd.L[20] ... 30000 valid values
##
## Iterations = 60002:90001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 30000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##               Mean       SD Naive SE Time-series SE
## deviance -5.41e+03 58.21400 3.36e-01       6.77e-01
## mu.L[1]   7.50e-01  0.01918 1.11e-04       1.49e-04
## mu.L[2]   5.95e-01  0.01589 9.18e-05       2.55e-04
## mu.L[3]   5.21e-01  0.01597 9.22e-05       1.54e-04
## mu.L[4]   6.08e-01  0.02149 1.24e-04       2.01e-04
## mu.L[5]   8.01e-01  0.01221 7.05e-05       1.40e-04
## mu.L[6]   5.33e-01  0.01871 1.08e-04       1.62e-04
## mu.L[7]   5.17e-01  0.02346 1.35e-04       1.75e-04
## mu.L[8]   5.11e-01  0.01565 9.04e-05       1.54e-04
## mu.L[9]   3.77e-01  0.04272 2.47e-04       2.74e-04
## mu.L[10]  5.93e-01  0.01905 1.10e-04       1.67e-04
## mu.L[11] -5.20e-02  0.01933 1.12e-04       5.39e-04
## mu.L[12]  5.24e-01  0.02631 1.52e-04       6.52e-04
## mu.L[13]  3.07e-01  0.02119 1.22e-04       3.02e-04
```

```
## mu.L[14]  3.35e-01  0.03376 1.95e-04       3.56e-04
## mu.L[15] -2.98e-02  0.01616 9.33e-05       5.92e-04
## mu.L[16]  5.77e-01  0.01639 9.46e-05       2.09e-04
## mu.L[17]  3.32e-01  0.02103 1.21e-04       2.12e-04
## mu.L[18]  5.97e-01  0.01593 9.20e-05       2.08e-04
## mu.L[19]  3.66e-01  0.02260 1.30e-04       2.23e-04
## mu.L[20]  3.90e-01  0.02453 1.42e-04       2.32e-04
## ppp       7.33e-01  0.44226 2.55e-03       2.78e-03
## sd.L[1]   1.04e-01  0.01570 9.07e-05       1.43e-04
## sd.L[2]   7.88e-02  0.01201 6.93e-05       1.14e-04
## sd.L[3]   8.20e-02  0.01216 7.02e-05       1.07e-04
## sd.L[4]   1.18e-01  0.01672 9.65e-05       1.38e-04
## sd.L[5]   5.81e-02  0.00956 5.52e-05       9.53e-05
## sd.L[6]   9.98e-02  0.01466 8.46e-05       1.34e-04
## sd.L[7]   1.30e-01  0.01849 1.07e-04       1.57e-04
## sd.L[8]   7.98e-02  0.01276 7.36e-05       1.33e-04
## sd.L[9]   2.47e-01  0.03346 1.93e-04       2.52e-04
## sd.L[10]  1.02e-01  0.01497 8.65e-05       1.34e-04
## sd.L[11]  5.11e-02  0.01864 1.08e-04       5.28e-04
## sd.L[12]  6.72e-02  0.02634 1.52e-04       7.43e-04
## sd.L[13]  7.98e-02  0.01872 1.08e-04       3.21e-04
## sd.L[14]  1.57e-01  0.03085 1.78e-04       4.20e-04
## sd.L[15]  2.82e-02  0.01510 8.72e-05       5.98e-04
## sd.L[16]  6.34e-02  0.01479 8.54e-05       2.35e-04
## sd.L[17]  9.92e-02  0.01918 1.11e-04       2.62e-04
## sd.L[18]  5.97e-02  0.01665 9.61e-05       3.32e-04
## sd.L[19]  1.07e-01  0.01892 1.09e-04       2.17e-04
## sd.L[20]  1.22e-01  0.02214 1.28e-04       2.59e-04
##
## 2. Quantiles for each variable:
##
##                2.5%       25%       50%       75%     97.5%
## deviance -5.52e+03 -5.44e+03 -5.40e+03 -5.37e+03 -5.29e+03
## mu.L[1]   7.12e-01  7.37e-01  7.50e-01  7.62e-01  7.87e-01
## mu.L[2]   5.64e-01  5.84e-01  5.94e-01  6.05e-01  6.26e-01
## mu.L[3]   4.90e-01  5.11e-01  5.21e-01  5.32e-01  5.53e-01
## mu.L[4]   5.65e-01  5.94e-01  6.08e-01  6.22e-01  6.51e-01
## mu.L[5]   7.77e-01  7.93e-01  8.01e-01  8.09e-01  8.26e-01
## mu.L[6]   4.95e-01  5.20e-01  5.33e-01  5.45e-01  5.69e-01
## mu.L[7]   4.71e-01  5.02e-01  5.17e-01  5.33e-01  5.64e-01
## mu.L[8]   4.80e-01  5.01e-01  5.11e-01  5.21e-01  5.42e-01
## mu.L[9]   2.93e-01  3.49e-01  3.77e-01  4.05e-01  4.61e-01
## mu.L[10]  5.55e-01  5.80e-01  5.93e-01  6.06e-01  6.30e-01
## mu.L[11] -8.81e-02 -6.52e-02 -5.25e-02 -3.93e-02 -1.25e-02
## mu.L[12]  4.73e-01  5.06e-01  5.23e-01  5.41e-01  5.78e-01
## mu.L[13]  2.66e-01  2.93e-01  3.07e-01  3.21e-01  3.50e-01
## mu.L[14]  2.69e-01  3.13e-01  3.35e-01  3.57e-01  4.03e-01
## mu.L[15] -6.03e-02 -4.09e-02 -3.03e-02 -1.92e-02  2.94e-03
## mu.L[16]  5.46e-01  5.66e-01  5.77e-01  5.88e-01  6.11e-01
## mu.L[17]  2.91e-01  3.18e-01  3.32e-01  3.45e-01  3.74e-01
## mu.L[18]  5.65e-01  5.86e-01  5.97e-01  6.07e-01  6.28e-01
## mu.L[19]  3.22e-01  3.51e-01  3.66e-01  3.81e-01  4.11e-01
## mu.L[20]  3.42e-01  3.74e-01  3.90e-01  4.06e-01  4.39e-01
## ppp       0.00e+00  0.00e+00  1.00e+00  1.00e+00  1.00e+00
```

```
## sd.L[1]   7.73e-02  9.29e-02  1.02e-01  1.13e-01  1.39e-01
## sd.L[2]   5.87e-02  7.03e-02  7.77e-02  8.61e-02  1.06e-01
## sd.L[3]   6.18e-02  7.34e-02  8.07e-02  8.93e-02  1.09e-01
## sd.L[4]   9.01e-02  1.06e-01  1.17e-01  1.28e-01  1.55e-01
## sd.L[5]   4.20e-02  5.14e-02  5.72e-02  6.38e-02  7.93e-02
## sd.L[6]   7.54e-02  8.94e-02  9.84e-02  1.09e-01  1.33e-01
## sd.L[7]   9.92e-02  1.17e-01  1.28e-01  1.41e-01  1.71e-01
## sd.L[8]   5.83e-02  7.08e-02  7.87e-02  8.74e-02  1.08e-01
## sd.L[9]   1.90e-01  2.23e-01  2.43e-01  2.67e-01  3.21e-01
## sd.L[10]  7.67e-02  9.09e-02  1.00e-01  1.10e-01  1.35e-01
## sd.L[11]  1.79e-02  3.81e-02  4.98e-02  6.25e-02  9.14e-02
## sd.L[12]  2.19e-02  4.89e-02  6.51e-02  8.35e-02  1.24e-01
## sd.L[13]  4.83e-02  6.66e-02  7.80e-02  9.10e-02  1.22e-01
## sd.L[14]  1.05e-01  1.35e-01  1.54e-01  1.76e-01  2.26e-01
## sd.L[15]  4.84e-03  1.69e-02  2.62e-02  3.70e-02  6.27e-02
## sd.L[16]  3.84e-02  5.31e-02  6.21e-02  7.23e-02  9.64e-02
## sd.L[17]  6.69e-02  8.55e-02  9.73e-02  1.11e-01  1.42e-01
## sd.L[18]  3.13e-02  4.81e-02  5.82e-02  6.97e-02  9.65e-02
## sd.L[19]  7.58e-02  9.34e-02  1.05e-01  1.18e-01  1.49e-01
## sd.L[20]  8.44e-02  1.06e-01  1.20e-01  1.36e-01  1.71e-01
##
##    deviance     mu.L[1]     mu.L[2]     mu.L[3]     mu.L[4]     mu.L[5]
## -0.62618879  0.60279679  0.89412333  1.18356765 -0.54143333 -0.09840504
##     mu.L[6]     mu.L[7]     mu.L[8]     mu.L[9]    mu.L[10]    mu.L[11]
## -0.45099278 -0.90208605 -0.53881585 -1.82380509  0.88765782 -0.54745007
##    mu.L[12]    mu.L[13]    mu.L[14]    mu.L[15]    mu.L[16]    mu.L[17]
## -1.82525431  0.90636922  1.58690437 -0.16585584  1.18652116 -0.55173415
##    mu.L[18]    mu.L[19]    mu.L[20]     sd.L[1]     sd.L[2]     sd.L[3]
##  1.77928466  0.26221234 -0.73566732  0.99505845  0.32013643 -0.16956391
##     sd.L[4]     sd.L[5]     sd.L[6]     sd.L[7]     sd.L[8]     sd.L[9]
## -0.15825453 -0.51835907  0.64445442  0.06718110 -0.46901451  1.08707068
##    sd.L[10]    sd.L[11]    sd.L[12]    sd.L[13]    sd.L[14]    sd.L[15]
## -0.54077133 -0.46345129 -1.50740792  0.58986141  1.17019768 -0.25528457
##    sd.L[16]    sd.L[17]    sd.L[18]    sd.L[19]    sd.L[20]
##  1.38582911  0.17779924 -0.89331257  1.18362302  1.24708359
##
## Iterations = 60002:90001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 30000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##               Mean       SD Naive SE Time-series SE
## deviance -5.41e+03 58.21400 3.36e-01       6.77e-01
## mu.L[1]   7.50e-01  0.01918 1.11e-04       1.49e-04
## mu.L[2]   5.95e-01  0.01589 9.18e-05       2.55e-04
## mu.L[3]   5.21e-01  0.01597 9.22e-05       1.54e-04
## mu.L[4]   6.08e-01  0.02149 1.24e-04       2.01e-04
## mu.L[5]   8.01e-01  0.01221 7.05e-05       1.40e-04
## mu.L[6]   5.33e-01  0.01871 1.08e-04       1.62e-04
## mu.L[7]   5.17e-01  0.02346 1.35e-04       1.75e-04
## mu.L[8]   5.11e-01  0.01565 9.04e-05       1.54e-04
```

```
## mu.L[9]    3.77e-01  0.04272 2.47e-04       2.74e-04
## mu.L[10]   5.93e-01  0.01905 1.10e-04       1.67e-04
## mu.L[11]  -5.20e-02  0.01933 1.12e-04       5.39e-04
## mu.L[12]   5.24e-01  0.02631 1.52e-04       6.52e-04
## mu.L[13]   3.07e-01  0.02119 1.22e-04       3.02e-04
## mu.L[14]   3.35e-01  0.03376 1.95e-04       3.56e-04
## mu.L[15]  -2.98e-02  0.01616 9.33e-05       5.92e-04
## mu.L[16]   5.77e-01  0.01639 9.46e-05       2.09e-04
## mu.L[17]   3.32e-01  0.02103 1.21e-04       2.12e-04
## mu.L[18]   5.97e-01  0.01593 9.20e-05       2.08e-04
## mu.L[19]   3.66e-01  0.02260 1.30e-04       2.23e-04
## mu.L[20]   3.90e-01  0.02453 1.42e-04       2.32e-04
## ppp        7.33e-01  0.44226 2.55e-03       2.78e-03
## sd.L[1]    1.04e-01  0.01570 9.07e-05       1.43e-04
## sd.L[2]    7.88e-02  0.01201 6.93e-05       1.14e-04
## sd.L[3]    8.20e-02  0.01216 7.02e-05       1.07e-04
## sd.L[4]    1.18e-01  0.01672 9.65e-05       1.38e-04
## sd.L[5]    5.81e-02  0.00956 5.52e-05       9.53e-05
## sd.L[6]    9.98e-02  0.01466 8.46e-05       1.34e-04
## sd.L[7]    1.30e-01  0.01849 1.07e-04       1.57e-04
## sd.L[8]    7.98e-02  0.01276 7.36e-05       1.33e-04
## sd.L[9]    2.47e-01  0.03346 1.93e-04       2.52e-04
## sd.L[10]   1.02e-01  0.01497 8.65e-05       1.34e-04
## sd.L[11]   5.11e-02  0.01864 1.08e-04       5.28e-04
## sd.L[12]   6.72e-02  0.02634 1.52e-04       7.43e-04
## sd.L[13]   7.98e-02  0.01872 1.08e-04       3.21e-04
## sd.L[14]   1.57e-01  0.03085 1.78e-04       4.20e-04
## sd.L[15]   2.82e-02  0.01510 8.72e-05       5.98e-04
## sd.L[16]   6.34e-02  0.01479 8.54e-05       2.35e-04
## sd.L[17]   9.92e-02  0.01918 1.11e-04       2.62e-04
## sd.L[18]   5.97e-02  0.01665 9.61e-05       3.32e-04
## sd.L[19]   1.07e-01  0.01892 1.09e-04       2.17e-04
## sd.L[20]   1.22e-01  0.02214 1.28e-04       2.59e-04
##
## 2. Quantiles for each variable:
##
##                2.5%       25%       50%       75%     97.5%
## deviance -5.52e+03 -5.44e+03 -5.40e+03 -5.37e+03 -5.29e+03
## mu.L[1]   7.12e-01  7.37e-01  7.50e-01  7.62e-01  7.87e-01
## mu.L[2]   5.64e-01  5.84e-01  5.94e-01  6.05e-01  6.26e-01
## mu.L[3]   4.90e-01  5.11e-01  5.21e-01  5.32e-01  5.53e-01
## mu.L[4]   5.65e-01  5.94e-01  6.08e-01  6.22e-01  6.51e-01
## mu.L[5]   7.77e-01  7.93e-01  8.01e-01  8.09e-01  8.26e-01
## mu.L[6]   4.95e-01  5.20e-01  5.33e-01  5.45e-01  5.69e-01
## mu.L[7]   4.71e-01  5.02e-01  5.17e-01  5.33e-01  5.64e-01
## mu.L[8]   4.80e-01  5.01e-01  5.11e-01  5.21e-01  5.42e-01
## mu.L[9]   2.93e-01  3.49e-01  3.77e-01  4.05e-01  4.61e-01
## mu.L[10]  5.55e-01  5.80e-01  5.93e-01  6.06e-01  6.30e-01
## mu.L[11] -8.81e-02 -6.52e-02 -5.25e-02 -3.93e-02 -1.25e-02
## mu.L[12]  4.73e-01  5.06e-01  5.23e-01  5.41e-01  5.78e-01
## mu.L[13]  2.66e-01  2.93e-01  3.07e-01  3.21e-01  3.50e-01
## mu.L[14]  2.69e-01  3.13e-01  3.35e-01  3.57e-01  4.03e-01
## mu.L[15] -6.03e-02 -4.09e-02 -3.03e-02 -1.92e-02  2.94e-03
## mu.L[16]  5.46e-01  5.66e-01  5.77e-01  5.88e-01  6.11e-01
```

```
## mu.L[17]  2.91e-01  3.18e-01  3.32e-01  3.45e-01  3.74e-01
## mu.L[18]  5.65e-01  5.86e-01  5.97e-01  6.07e-01  6.28e-01
## mu.L[19]  3.22e-01  3.51e-01  3.66e-01  3.81e-01  4.11e-01
## mu.L[20]  3.42e-01  3.74e-01  3.90e-01  4.06e-01  4.39e-01
## ppp       0.00e+00  0.00e+00  1.00e+00  1.00e+00  1.00e+00
## sd.L[1]   7.73e-02  9.29e-02  1.02e-01  1.13e-01  1.39e-01
## sd.L[2]   5.87e-02  7.03e-02  7.77e-02  8.61e-02  1.06e-01
## sd.L[3]   6.18e-02  7.34e-02  8.07e-02  8.93e-02  1.09e-01
## sd.L[4]   9.01e-02  1.06e-01  1.17e-01  1.28e-01  1.55e-01
## sd.L[5]   4.20e-02  5.14e-02  5.72e-02  6.38e-02  7.93e-02
## sd.L[6]   7.54e-02  8.94e-02  9.84e-02  1.09e-01  1.33e-01
## sd.L[7]   9.92e-02  1.17e-01  1.28e-01  1.41e-01  1.71e-01
## sd.L[8]   5.83e-02  7.08e-02  7.87e-02  8.74e-02  1.08e-01
## sd.L[9]   1.90e-01  2.23e-01  2.43e-01  2.67e-01  3.21e-01
## sd.L[10]  7.67e-02  9.09e-02  1.00e-01  1.10e-01  1.35e-01
## sd.L[11]  1.79e-02  3.81e-02  4.98e-02  6.25e-02  9.14e-02
## sd.L[12]  2.19e-02  4.89e-02  6.51e-02  8.35e-02  1.24e-01
## sd.L[13]  4.83e-02  6.66e-02  7.80e-02  9.10e-02  1.22e-01
## sd.L[14]  1.05e-01  1.35e-01  1.54e-01  1.76e-01  2.26e-01
## sd.L[15]  4.84e-03  1.69e-02  2.62e-02  3.70e-02  6.27e-02
## sd.L[16]  3.84e-02  5.31e-02  6.21e-02  7.23e-02  9.64e-02
## sd.L[17]  6.69e-02  8.55e-02  9.73e-02  1.11e-01  1.42e-01
## sd.L[18]  3.13e-02  4.81e-02  5.82e-02  6.97e-02  9.65e-02
## sd.L[19]  7.58e-02  9.34e-02  1.05e-01  1.18e-01  1.49e-01
## sd.L[20]  8.44e-02  1.06e-01  1.20e-01  1.36e-01  1.71e-01
```

```r
fit[-9]
```

```
## $est
##  mu.L[1]  mu.L[2]  mu.L[3]  mu.L[4]  mu.L[5]  mu.L[6]  mu.L[7]  mu.L[8]
##    0.750    0.594    0.521    0.608    0.801    0.533    0.517    0.511
##  mu.L[9] mu.L[10] mu.L[11] mu.L[12] mu.L[13] mu.L[14] mu.L[15] mu.L[16]
##    0.377    0.593   -0.053    0.523    0.307    0.335   -0.030    0.577
## mu.L[17] mu.L[18] mu.L[19] mu.L[20]  sd.L[1]  sd.L[2]  sd.L[3]  sd.L[4]
##    0.332    0.597    0.366    0.390    0.102    0.078    0.081    0.116
##  sd.L[5]  sd.L[6]  sd.L[7]  sd.L[8]  sd.L[9] sd.L[10] sd.L[11] sd.L[12]
##    0.057    0.098    0.128    0.079    0.243    0.100    0.050    0.065
## sd.L[13] sd.L[14] sd.L[15] sd.L[16] sd.L[17] sd.L[18] sd.L[19] sd.L[20]
##    0.078    0.154    0.026    0.062    0.097    0.058    0.105    0.120
##
##    0.733
##
## $psd
##  mu.L[1]  mu.L[2]  mu.L[3]  mu.L[4]  mu.L[5]  mu.L[6]  mu.L[7]  mu.L[8]
##    0.019    0.016    0.016    0.021    0.012    0.019    0.023    0.016
##  mu.L[9] mu.L[10] mu.L[11] mu.L[12] mu.L[13] mu.L[14] mu.L[15] mu.L[16]
##    0.043    0.019    0.019    0.026    0.021    0.034    0.016    0.016
## mu.L[17] mu.L[18] mu.L[19] mu.L[20]  sd.L[1]  sd.L[2]  sd.L[3]  sd.L[4]
##    0.021    0.016    0.023    0.025    0.016    0.012    0.012    0.017
##  sd.L[5]  sd.L[6]  sd.L[7]  sd.L[8]  sd.L[9] sd.L[10] sd.L[11] sd.L[12]
##    0.010    0.015    0.018    0.013    0.033    0.015    0.019    0.026
## sd.L[13] sd.L[14] sd.L[15] sd.L[16] sd.L[17] sd.L[18] sd.L[19] sd.L[20]
##    0.019    0.031    0.015    0.015    0.019    0.017    0.019    0.022
##
## $CIl
```

```
##  mu.L[1]  mu.L[2]  mu.L[3]  mu.L[4]  mu.L[5]  mu.L[6]  mu.L[7]  mu.L[8]
##    0.712    0.563    0.490    0.565    0.777    0.497    0.472    0.481
##  mu.L[9] mu.L[10] mu.L[11] mu.L[12] mu.L[13] mu.L[14] mu.L[15] mu.L[16]
##    0.292    0.556   -0.090    0.472    0.264    0.270   -0.060    0.545
## mu.L[17] mu.L[18] mu.L[19] mu.L[20]  sd.L[1]  sd.L[2]  sd.L[3]  sd.L[4]
##    0.291    0.565    0.322    0.341    0.075    0.057    0.060    0.087
##  sd.L[5]  sd.L[6]  sd.L[7]  sd.L[8]  sd.L[9] sd.L[10] sd.L[11] sd.L[12]
##    0.041    0.073    0.096    0.056    0.186    0.075    0.015    0.018
## sd.L[13] sd.L[14] sd.L[15] sd.L[16] sd.L[17] sd.L[18] sd.L[19] sd.L[20]
##    0.047    0.101    0.001    0.036    0.065    0.029    0.073    0.082
##
## $CIu
##  mu.L[1]  mu.L[2]  mu.L[3]  mu.L[4]  mu.L[5]  mu.L[6]  mu.L[7]  mu.L[8]
##    0.787    0.625    0.553    0.650    0.826    0.571    0.565    0.543
##  mu.L[9] mu.L[10] mu.L[11] mu.L[12] mu.L[13] mu.L[14] mu.L[15] mu.L[16]
##    0.460    0.631   -0.015    0.577    0.348    0.403    0.003    0.610
## mu.L[17] mu.L[18] mu.L[19] mu.L[20]  sd.L[1]  sd.L[2]  sd.L[3]  sd.L[4]
##    0.374    0.628    0.410    0.438    0.135    0.103    0.106    0.151
##  sd.L[5]  sd.L[6]  sd.L[7]  sd.L[8]  sd.L[9] sd.L[10] sd.L[11] sd.L[12]
##    0.077    0.129    0.166    0.104    0.315    0.132    0.088    0.119
## sd.L[13] sd.L[14] sd.L[15] sd.L[16] sd.L[17] sd.L[18] sd.L[19] sd.L[20]
##    0.118    0.220    0.057    0.092    0.138    0.093    0.145    0.167
##
## $CVl
##  mu.L[1]  mu.L[2]  mu.L[3]  mu.L[4]  mu.L[5]  mu.L[6]  mu.L[7]  mu.L[8]
##    0.619    0.494    0.417    0.460    0.728    0.408    0.353    0.410
##  mu.L[9] mu.L[10] mu.L[11] mu.L[12] mu.L[13] mu.L[14] mu.L[15] mu.L[16]
##    0.066    0.465   -0.117    0.440    0.207    0.138   -0.063    0.498
## mu.L[17] mu.L[18] mu.L[19] mu.L[20]
##    0.208    0.523    0.232    0.236
##
## $CVu
##  mu.L[1]  mu.L[2]  mu.L[3]  mu.L[4]  mu.L[5]  mu.L[6]  mu.L[7]  mu.L[8]
##    0.881    0.694    0.625    0.756    0.874    0.658    0.681    0.612
##  mu.L[9] mu.L[10] mu.L[11] mu.L[12] mu.L[13] mu.L[14] mu.L[15] mu.L[16]
##    0.688    0.721    0.011    0.606    0.407    0.532    0.003    0.656
## mu.L[17] mu.L[18] mu.L[19] mu.L[20]
##    0.456    0.671    0.500    0.544
##
## $conv
##          deviance  mu.L[1]  mu.L[2]  mu.L[3]  mu.L[4]  mu.L[5]  mu.L[6]
##    3.000   -0.626    0.603    0.894    1.184   -0.541   -0.098   -0.451
##  mu.L[7]  mu.L[8]  mu.L[9] mu.L[10] mu.L[11] mu.L[12] mu.L[13] mu.L[14]
##   -0.902   -0.539   -1.824    0.888   -0.547   -1.825    0.906    1.587
## mu.L[15] mu.L[16] mu.L[17] mu.L[18] mu.L[19] mu.L[20]  sd.L[1]  sd.L[2]
##   -0.166    1.187   -0.552    1.779    0.262   -0.736    0.995    0.320
##  sd.L[3]  sd.L[4]  sd.L[5]  sd.L[6]  sd.L[7]  sd.L[8]  sd.L[9] sd.L[10]
##   -0.170   -0.158   -0.518    0.644    0.067   -0.469    1.087   -0.541
## sd.L[11] sd.L[12] sd.L[13] sd.L[14] sd.L[15] sd.L[16] sd.L[17] sd.L[18]
##   -0.463   -1.507    0.590    1.170   -0.255    1.386    0.178   -0.893
## sd.L[19] sd.L[20]
##    1.184    1.247
##
## $DIC
```

```
## [1] -4792
```

4 OSMASEM

4.1 Data preparation

```
# Modified based on the code from Jak & Cheung (2019)
# Exclude studies that reported CFA results only
index <- Gnambs18$CorMat==1
Gnambs18 <- lapply(Gnambs18, function(x) x[index])

## Create a dataframe with the data and the asymptotic variances and covariances (acov)
my.df <- Cor2DataFrame(Gnambs18$data, Gnambs18$n, acov = "weighted")

## Add the standardized individualism as the moderator
## Standardization of the moderator improves the convergence.
my.df$data <- data.frame(my.df$data,
                         Individualism=scale(Gnambs18$Individualism),
                         check.names=FALSE)
summary(my.df)
```

```
##           Length Class      Mode
## data      1081   data.frame list
## n           36   -none-     numeric
## obslabels   10   -none-     character
## ylabels     45   -none-     character
## vlabels   1035   -none-     character
```

4.2 Model fitting

```
## Specify the bifactor model
model0 <- "G =~  g1*I1 + g2*I2 + g3*I3 + g4*I4 + g5*I5 +
                 g6*I6 + g7*I7 + g8*I8 + g9*I9 + g10*I10
          POS =~ p1*I1 + p3*I3 + p4*I4 + p7*I7 + p10*I10
          NEG =~ n2*I2 + n5*I5 + n6*I6 + n8*I8 + n9*I9"
RAM0 <- lavaan2RAM(model0, obs.variables = paste0("I", 1:10),std.lv = TRUE)


## Create matrices with implicit diagonal constraints
M0 <- create.vechsR(A0=RAM0$A, S0=RAM0$S, F0=RAM0$F)

## Create heterogeneity variances
T0 <- create.Tau2(RAM=RAM0, RE.type="Diag", Transform="expLog", RE.startvalues=0.05)

## Fit the bifactor model with One-Stage MASEM
fit0 <- osmasem(model.name="No moderator", Mmatrix=M0, Tmatrix=T0, data=my.df)
summary(fit0, fitIndices= T)
```

```
## Summary of No moderator
##
## free parameters:
##       name  matrix row col   Estimate  Std.Error A    z value     Pr(>|z|)
## 1       g1      A0  I1   G  0.75376651 0.01428142    52.7795357 0.000000e+00
```

```
## 2       g2      A0  I2   G  0.52690812 0.01316695    40.0174708 0.000000e+00
## 3       g3      A0  I3   G  0.59383220 0.01794379    33.0940255 0.000000e+00
## 4       g4      A0  I4   G  0.52030803 0.01279994    40.6492470 0.000000e+00
## 5       g5      A0  I5   G  0.51957427 0.01521478    34.1493153 0.000000e+00
## 6       g6      A0  I6   G  0.50463142 0.01166448    43.2622373 0.000000e+00
## 7       g7      A0  I7   G  0.61377018 0.01430301    42.9119689 0.000000e+00
## 8       g8      A0  I8   G  0.37704998 0.01821173    20.7036830 0.000000e+00
## 9       g9      A0  I9   G  0.58502358 0.01397718    41.8556141 0.000000e+00
## 10     g10      A0 I10   G  0.80216595 0.01479533    54.2175145 0.000000e+00
## 11      p1      A0  I1 POS -0.04435759 0.04343238    -1.0213024 3.071112e-01
## 12      p3      A0  I3 POS  0.52666674 0.06755835     7.7957311 6.439294e-15
## 13      p4      A0  I4 POS  0.31017237 0.03679538     8.4296559 0.000000e+00
## 14      p7      A0  I7 POS  0.32915003 0.04110032     8.0084540 1.110223e-15
## 15     p10      A0 I10 POS -0.02982338 0.04483588    -0.6651677 5.059432e-01
## 16      n2      A0  I2 NEG  0.59626978 0.02436976    24.4676084 0.000000e+00
## 17      n5      A0  I5 NEG  0.33215576 0.02253133    14.7419528 0.000000e+00
## 18      n6      A0  I6 NEG  0.60359794 0.02342084    25.7718329 0.000000e+00
## 19      n8      A0  I8 NEG  0.40420952 0.02947566    13.7133321 0.000000e+00
## 20      n9      A0  I9 NEG  0.39207555 0.02262261    17.3311388 0.000000e+00
## 21  Tau1_1 vecTau1   1   1 -4.73538818 0.25346939   -18.6822884 0.000000e+00
## 22  Tau1_2 vecTau1   2   1 -5.06608146 0.25834504   -19.6097493 0.000000e+00
## 23  Tau1_3 vecTau1   3   1 -5.09422214 0.25758293   -19.7770175 0.000000e+00
## 24  Tau1_4 vecTau1   4   1 -4.38166777 0.25670935   -17.0685943 0.000000e+00
## 25  Tau1_5 vecTau1   5   1 -5.40630749 0.26900224   -20.0976299 0.000000e+00
## 26  Tau1_6 vecTau1   6   1 -4.49283070 0.24830739   -18.0938256 0.000000e+00
## 27  Tau1_7 vecTau1   7   1 -3.72863021 0.24376561   -15.2959647 0.000000e+00
## 28  Tau1_8 vecTau1   8   1 -4.92081370 0.25414114   -19.3625231 0.000000e+00
## 29  Tau1_9 vecTau1   9   1 -4.40562783 0.24877852   -17.7090366 0.000000e+00
## 30 Tau1_10 vecTau1  10   1 -5.13630458 0.26019820   -19.7399694 0.000000e+00
## 31 Tau1_11 vecTau1  11   1 -5.36833759 0.26151700   -20.5276809 0.000000e+00
## 32 Tau1_12 vecTau1  12   1 -5.20391390 0.26417659   -19.6986187 0.000000e+00
## 33 Tau1_13 vecTau1  13   1 -4.73457357 0.26648698   -17.7666226 0.000000e+00
## 34 Tau1_14 vecTau1  14   1 -4.74288169 0.24758407   -19.1566513 0.000000e+00
## 35 Tau1_15 vecTau1  15   1 -4.17034894 0.24976031   -16.6974043 0.000000e+00
## 36 Tau1_16 vecTau1  16   1 -5.16882130 0.26522534   -19.4884139 0.000000e+00
## 37 Tau1_17 vecTau1  17   1 -5.02363230 0.25034902   -20.0665145 0.000000e+00
## 38 Tau1_18 vecTau1  18   1 -5.61090499 0.26770889   -20.9589793 0.000000e+00
## 39 Tau1_19 vecTau1  19   1 -4.67875580 0.26201227   -17.8570104 0.000000e+00
## 40 Tau1_20 vecTau1  20   1 -5.45518419 0.27586930   -19.7745245 0.000000e+00
## 41 Tau1_21 vecTau1  21   1 -4.05751465 0.24506049   -16.5571966 0.000000e+00
## 42 Tau1_22 vecTau1  22   1 -4.29850628 0.25583344   -16.8019722 0.000000e+00
## 43 Tau1_23 vecTau1  23   1 -5.25245576 0.26092947   -20.1297914 0.000000e+00
## 44 Tau1_24 vecTau1  24   1 -5.20113851 0.27046847   -19.2301105 0.000000e+00
## 45 Tau1_25 vecTau1  25   1 -5.11065616 0.26751221   -19.1043844 0.000000e+00
## 46 Tau1_26 vecTau1  26   1 -5.48910548 0.26487481   -20.7233958 0.000000e+00
## 47 Tau1_27 vecTau1  27   1 -4.43866571 0.25417664   -17.4629173 0.000000e+00
## 48 Tau1_28 vecTau1  28   1 -4.23257069 0.25220081   -16.7825420 0.000000e+00
## 49 Tau1_29 vecTau1  29   1 -5.08512828 0.26297390   -19.3370074 0.000000e+00
## 50 Tau1_30 vecTau1  30   1 -5.52340960 0.26906785   -20.5279432 0.000000e+00
## 51 Tau1_31 vecTau1  31   1 -5.42059315 0.27477584   -19.7273279 0.000000e+00
## 52 Tau1_32 vecTau1  32   1 -4.70962182 0.25275347   -18.6332625 0.000000e+00
## 53 Tau1_33 vecTau1  33   1 -4.66788776 0.26016139   -17.9422771 0.000000e+00
## 54 Tau1_34 vecTau1  34   1 -4.16909395 0.25496447   -16.3516664 0.000000e+00
## 55 Tau1_35 vecTau1  35   1 -4.65798886 0.26799505   -17.3808764 0.000000e+00
```

```
## 56 Tau1_36 vecTau1  36   1 -5.17375963 0.25621629   -20.1929380 0.000000e+00
## 57 Tau1_37 vecTau1  37   1 -4.10193197 0.25183095   -16.2884345 0.000000e+00
## 58 Tau1_38 vecTau1  38   1 -5.40177287 0.26300180   -20.5389201 0.000000e+00
## 59 Tau1_39 vecTau1  39   1 -5.49125703 0.26052110   -21.0779743 0.000000e+00
## 60 Tau1_40 vecTau1  40   1 -4.11753568 0.24725406   -16.6530562 0.000000e+00
## 61 Tau1_41 vecTau1  41   1 -4.82079579 0.24845538   -19.4030649 0.000000e+00
## 62 Tau1_42 vecTau1  42   1 -4.86610897 0.24766348   -19.6480685 0.000000e+00
## 63 Tau1_43 vecTau1  43   1 -4.09065004 0.24785859   -16.5039671 0.000000e+00
## 64 Tau1_44 vecTau1  44   1 -3.51119105 0.24379173   -14.4024209 0.000000e+00
## 65 Tau1_45 vecTau1  45   1 -4.88975638 0.25241546   -19.3718576 0.000000e+00
##
## To obtain confidence intervals re-run with intervals=TRUE
##
## Model Statistics:
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:             65                   1555             -2737.794
##    Saturated:             90                   1530             -2777.449
## Independence:             45                   1575              1549.587
## Number of observations/statistics: 109988/1620
##
## chi-square:   ² ( df=25 ) = 39.65516,  p = 0.03163193
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:      -5847.794             -2607.794                 -2607.715
## BIC:     -20788.430             -1983.265                 -2189.838
## CFI: 0.9965775
## TLI: 0.9938395   (also known as NNFI)
## RMSEA:  0.002308621  [95% CI (0, 0.003835472)]
## Prob(RMSEA <= 0.05): 1
## timestamp: 2023-12-12 01:32:22
## Wall clock time: 115.7895 secs
## optimizer:  SLSQP
## OpenMx version number: 2.21.8
## Need help?  See help(mxSummary)
```

```
## SRMR
osmasemSRMR(fit0)
```

```
## [1] 0.01646314
```