L5_DFA_MIIVsem

Zachary Fisher & Kathleen M. Gates

Function for generating data.

Below we introduce how to make a function. This function requires the user to input the structure and values of the relations in matrix form (i.e., psi, lambda, theta, and phi). The user must also specify an argument, "time", that indicates the length of the series.

By default, the time series is padded by 100 observations. This helps to initialize the data generation; the time points after this are retained.

```
library(mvtnorm)

# Function for generating data
generateData <- function(psi, lambda, phi, theta, time, npad = 100){
  ne        <- nrow(psi)   # Number of latent variables
  ny        <- nrow(theta) # Number of manifest variables
  timepad   <- time + npad
  # Generate LV time series
  #  Latent variable residuals.
  zeta <- mvtnorm::rmvnorm(timepad, rep(0, ne), sigma = psi)
  #  Set up matrix for contemporaneous variables.
  etaC <- matrix(0, nrow = ne, ncol = timepad)
  #  Set up matrix for lagged variables.
  etaL <- matrix(0, nrow = ne, ncol = timepad + 1)
  etaL[,1]  <- c(0,0)
  for (i in 1:(timepad)){
    etaC[ ,i]   <- phi %*% etaL[ ,i] + zeta[i, ]
    etaL[ ,i+1] <- etaC[,i]
  }
  etaC <- etaC[,(npad+1):(timepad)]
  etaL <- etaL[,(npad+1):(timepad)]
  eta  <- t(etaC)
  # Generate observed variable time series
  #  Measurement errors.
  epsilon   <- rmvnorm(time, mean = rep(0, ny), sigma = theta)
  #  Observed variables.
  y   <- matrix(0, nrow = time, ncol = ny)
  for (p in 1:nrow(y)){
    y[p, ] <- lambda %*% eta[p, ] + epsilon[p, ]
  }
  # Create block toeplitz data structure
  data <- embed(y,2)
  # create column names
  var.names <- c()
  for(j in 0:1){
    for (i in 1:ny){
        var.names <- c(var.names,paste0("y",i,"_L",j))
    }
  }
  colnames(data) <- var.names
```

```
  return(data)
}
```

Generate Data

The data are generated to have the same pattern used in the book as well as in last lecture. However, now we are generating a lag of 2.

```
# Residual variance-covariance matrix
psi <- matrix(
  c( 2.77, 2.47,
     2.47, 8.40 ),
  nrow = 2, ncol = 2, byrow = T
)

# Lambda matrix containing contemporaneous relations among
# observed variables and 2 latent variables
lambda <- matrix(
  c( 1, 0,
     2, 0,
     1, 0,
     0, 1,
     0, 2,
     0, 1 ),
  nrow = 6, ncol = 2, byrow = TRUE
)

 # Measurement error variances.
theta     <- diag(.5, ncol = 6, nrow = 6)

# Lagged directed relations among variables.
phi      <- matrix(
  c(0.5, 0.0,
    0.4, 0.5),
  nrow = 2, ncol = 2, byrow = TRUE
)

time <- 200

set.seed(123456)

data <- generateData(psi, lambda, phi, theta, time, npad = 100)
```

Process Factor Analysis Models

Below we define the model structures for the correctly and the incorrectly specified models. The incorrectly specified model removes the cross loadings from observed variables 5 and 2.

```
model.correct <- '
  eta1_L0 =~ y1_L0 + l1*y2_L0 + l2*y3_L0
  eta2_L0 =~ y4_L0 + l4*y5_L0 + l5*y6_L0
  eta1_L1 =~ y1_L1 + l1*y2_L1 + l2*y3_L1
  eta2_L1 =~ y4_L1 + l4*y5_L1 + l5*y6_L1
  eta1_L0 ~ eta1_L1
  eta2_L0 ~ eta2_L1
  eta2_L0 ~ eta1_L1
'

model.misspecified <- '
  eta1_L0 =~ y1_L0 + l1*y2_L0 + l2*y3_L0 + l3*y5_L0
  eta2_L0 =~ y4_L0 + l4*y5_L0 + l5*y6_L0 + l6*y2_L0
  eta1_L1 =~ y1_L1 + l1*y2_L1 + l2*y3_L1 + l3*y5_L1
  eta2_L1 =~ y4_L1 + l4*y5_L1 + l5*y6_L1 + l6*y2_L1
  eta2_L0 ~ eta2_L1
  eta2_L0 ~ eta1_L1
  eta1_L0 ~ eta2_L1
'
```

Fit Process Factor Analysis Models using MIIVsem

Correctly Specified Model

```
fit.miiv.cor <- MIIVsem::miive(model.correct, data = data)
fit.miiv.cor
```

```
## MIIVsem (0.5.4) results
##
## Number of observations                                                    199
## Number of equations                                                        10
## Estimator                                                           MIIV-2SLS
## Standard Errors                                                      standard
## Missing                                                              listwise
##
##
## Parameter Estimates:
##
##
## STRUCTURAL COEFFICIENTS:
##                    Estimate  Std.Err  z-value  P(>|z|)   Sargan   df   P(Chi)
##   eta1_L0 =~
##     y1_L0             1.000
##     y2_L0             1.972    0.039   50.325    0.000   14.864    9    0.095
##     y3_L0             0.956    0.024   39.533    0.000   12.543    9    0.184
##   eta1_L1 =~
##     y1_L1             1.000
##     y2_L1             1.972    0.039   50.325    0.000    3.137    9    0.959
```

```
##     y3_L1            0.956    0.024   39.533    0.000   13.524    9    0.140
##   eta2_L0 =~
##     y4_L0            1.000
##     y5_L0            1.894    0.021   89.361    0.000    3.552    9    0.938
##     y6_L0            0.936    0.014   68.416    0.000    5.222    9    0.815
##   eta2_L1 =~
##     y4_L1            1.000
##     y5_L1            1.894    0.021   89.361    0.000   10.049    9    0.347
##     y6_L1            0.936    0.014   68.416    0.000    5.318    9    0.806
##
##   eta1_L0 ~
##     eta1_L1          0.426    0.070    6.056    0.000    6.043    4    0.196
##   eta2_L0 ~
##     eta1_L1          0.347    0.123    2.813    0.005    1.753    2    0.416
##     eta2_L1          0.526    0.068    7.779    0.000
##
## INTERCEPTS:
##                   Estimate  Std.Err  z-value  P(>|z|)
##     eta1_L0          0.160    0.144    1.113    0.266
##     eta2_L0          0.215    0.214    1.006    0.314
##     y1_L0            0.000
##     y1_L1            0.000
##     y2_L0            0.151    0.108    1.404    0.160
##     y2_L1            0.156    0.108    1.446    0.148
##     y3_L0            0.051    0.070    0.733    0.463
##     y3_L1            0.057    0.070    0.813    0.416
##     y4_L0            0.000
##     y4_L1            0.000
##     y5_L0            0.047    0.109    0.428    0.668
##     y5_L1            0.063    0.110    0.569    0.569
##     y6_L0            0.086    0.072    1.189    0.235
##     y6_L1            0.086    0.072    1.192    0.233
```

Misspecified Model

Here we might expect to see problems with the Λ_{21} and Λ_{52} parameters at both lags.

```
fit.miiv.mis <- MIIVsem::miive(model.misspecified, data = data)
fit.miiv.mis
```

```
## MIIVsem (0.5.4) results
##
## Number of observations                                                   199
## Number of equations                                                       10
## Estimator                                                          MIIV-2SLS
## Standard Errors                                                     standard
## Missing                                                             listwise
##
##
## Parameter Estimates:
##
##
## STRUCTURAL COEFFICIENTS:
##                   Estimate  Std.Err  z-value  P(>|z|)   Sargan   df   P(Chi)
```

```
##   eta1_L0 =~
##     y1_L0              1.000
##     y2_L0              1.950    0.047   41.693    0.000   14.409    7    0.044
##     y3_L0              0.956    0.024   39.533    0.000   12.543    9    0.184
##     y5_L0              0.040    0.045    0.880    0.379    3.089    7    0.877
##   eta1_L1 =~
##     y1_L1              1.000
##     y2_L1              1.950    0.047   41.693    0.000    2.906    7    0.894
##     y3_L1              0.956    0.024   39.533    0.000   13.524    9    0.140
##     y5_L1              0.040    0.045    0.880    0.379    9.742    7    0.204
##   eta2_L0 =~
##     y4_L0              1.000
##     y2_L0              0.020    0.024    0.820    0.412   14.409    7    0.044
##     y5_L0              1.882    0.025   74.799    0.000    3.089    7    0.877
##     y6_L0              0.936    0.014   68.416    0.000    5.222    9    0.815
##   eta2_L1 =~
##     y4_L1              1.000
##     y2_L1              0.020    0.024    0.820    0.412    2.906    7    0.894
##     y5_L1              1.882    0.025   74.799    0.000    9.742    7    0.204
##     y6_L1              0.936    0.014   68.416    0.000    5.318    9    0.806
##
##   eta1_L0 ~
##     eta2_L1            0.112    0.041    2.750    0.006   30.756    4    0.000
##   eta2_L0 ~
##     eta2_L1            0.526    0.068    7.779    0.000    1.753    2    0.416
##     eta1_L1            0.347    0.123    2.813    0.005
##
## INTERCEPTS:
##                     Estimate  Std.Err  z-value  P(>|z|)
##     eta1_L0            0.192    0.153    1.252    0.210
##     eta2_L0            0.215    0.214    1.006    0.314
##     y1_L0              0.000
##     y1_L1              0.000
##     y2_L0              0.144    0.107    1.347    0.178
##     y2_L1              0.148    0.107    1.383    0.167
##     y3_L0              0.051    0.070    0.733    0.463
##     y3_L1              0.057    0.070    0.813    0.416
##     y4_L0              0.000
##     y4_L1              0.000
##     y5_L0              0.044    0.109    0.405    0.685
##     y5_L1              0.061    0.110    0.555    0.579
##     y6_L0              0.086    0.072    1.189    0.235
##     y6_L1              0.086    0.072    1.192    0.233
```

Fit Process Factor Analysis Models in lavaan

```
fit.lava.cor    <- lavaan::sem(model.correct, data = data)
fit.lava.mis    <- lavaan::sem(model.misspecified, data = data)
```

Compare lavaan and MIIVsem Estimates

```
# Pull the MIIV parameter estimates.
fit.miiv.cor.et <- MIIVsem::estimatesTable(fit.miiv.cor)
fit.miiv.mis.et <- MIIVsem::estimatesTable(fit.miiv.mis)
# Pull the lavaan paramter estimates.
fit.lava.cor.pt <- lavaan::parameterTable(fit.lava.cor)
fit.lava.mis.pt <- lavaan::parameterTable(fit.lava.mis)
# Create a pretty table.
keep.cols       <- c("lhs", "op", "rhs", "est")
res.miiv <- merge(fit.miiv.cor.et[,keep.cols],fit.miiv.mis.et[,keep.cols], by = c("lhs","op","rhs"))
res.lava <- merge(fit.lava.cor.pt[,keep.cols],fit.lava.mis.pt[,keep.cols], by = c("lhs","op","rhs"))
res.all  <- merge(res.miiv,res.lava, by = c("lhs","op","rhs"))
colnames(res.all) <- c("lhs","op","rhs","miiv.cor","miiv.mis","lava.cor","lava.mis")
col.order <- c("lhs","op","rhs","pop","miiv.cor","miiv.mis","lava.cor","lava.mis")
res.all  <- res.all[!grepl("L1",res.all$lhs) & res.all$op %in% c("~","=~"),]
res.all[,c(4:7)]  <- round(res.all[,c(4:7)],2)
```