Package ‘EFAutilities’

September 7, 2020

Type Package

Title Utility Functions for Exploratory Factor Analysis

Version 2.1.0

Date 2020-08-23

Author Guangjian Zhang, Ge Jiang, Minami Hattori, Lauren Trichtinger

Maintainer Guangjian Zhang <gzhang3@nd.edu>

Depends R (>= 2.10)

Description A number of utility function for exploratory factor analysis are included in this package. In particular, it computes standard errors for parameter estimates and factor correlations under a variety of conditions.

License GPL-2

LazyLoad yes

Imports GPArotation, mvtnorm, plyr, graphics, stats, utils, MASS, Rcpp (>= 1.0.3)

RoxygenNote 7.1.0

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-09-07 07:20:07 UTC

Suggests knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LinkingTo Rcpp

R topics documented:

`Align.Matrix` *Factor Alignment*

Description

The function is to align a factor solution according to an order matrix. The output matrix is a (p+m+1) by m matrix, where the first p rows are factor loadings of the best match, the next m rows are factor correlations of the best match, and the last row contains the sums of squared deviations for the best match and the second best match. The difference between the best match and the second best match could be considered as a confidence on the success of the aligning procedure (a computationally more efficient method exists for some conditions; whenever this occurs we only report that of the best match).

Usage

```
Align.Matrix(Order.Matrix, Input.Matrix, Weight.Matrix=NULL)
```

Arguments

`Order.Matrix`	A p by m matrix: p is the number of manifest variables and m is the number of latent factors
`Input.Matrix`	A (p+m) by m matrix, the first p rows are factor loadings, the last m rows are factor correlations
`Weight.Matrix`	A p by m matrix that assigns weight to the order matrix: NULL (default)

Details

Align.Matrix is an R function to reflect and interchange columns of Input.Matrix to match those of Order.Matrix. Because it considers all possible permutations of columns of Input.Matrix,the best match in terms of the smallest sum of squared deviations between these two matrices can always be found. It may be slow if there are too many factors.

Author(s)

Guangjian Zhang

Examples

```
#Order Matrix
A <- matrix(c(0.8,0.7,0,0,0,0,0.8,0.7),nrow=4,ncol=2)

#Input.Matrix
B <-matrix(c(0,0,-0.8,-0.7,1,-0.2,0.8,0.7,0,0,-0.2,1),nrow=6,ncol=2)

Align.Matrix(Order.Matrix=A, Input.Matrix=B)
```

`BFI228` *Ordinal Data of the Big Five Inventory (BFI)*

Description

The BFI228 is part of the study on personality and relationship satisfaction (Luo, 2005). The participants were 228 undergraduate students at a large public university in the US. The data were participants' self ratings on the 44 items of the Big Five Inventory (John, Donahue, & Kentle, 1991). These items are Likert variables: disagree strongly (1), disagree a little (2), neither agree nor disagree (3), agree a little (4), and agree strongly (5).

Usage

```
data(BFI228)
```

Format

The format is a n by p matrix of ordinal variables, where n is the number of participants (228) and p is the number of manifest variables (44).

Details

The variables were ordered such that indicators of the same factor are grouped together. Note that reverse-coded items are denoted by '_R'.

V01 to V08 are variables for the factor extraversion: talkative, reserved_R, fullenergy, enthusiastic, quiet_R, assertive, shy_R, and outgoing.

V09 to V17 are variables for the factor agreeableness: findfault_R, helpful, quarrels_R, forgiving, trusting, cold_R, considerate, rude_R, and cooperative.

V18 to V26 are variables for the factor conscientiousness are: thorough, careless_R, reliable, disorganized_R, lazy_R, persevere, efficient, plans, and distracted_R.

V27 to V34 are variables for the factor neuroticism: blue, relaxed_R, tense, worries, emostable_R, moody, calm_R, and nervous.

V35 to V44 are variables for the factor openness: ideas, curious, ingenious, imagination, inventive, artistic, routine_R, reflect, nonartistic, and sophisticated.

References

John, O. P., Donahue, E. M., & Kentle, R. L. (1991). The Big Five Inventory versions 4a and 54. Berkeley, CA: University of California,Berkeley, Institute of Personality and Social Research.

Luo, S. (2005): unpublished study on personality traits and relationship satisfaction.

CPAI537 *Composite Scores of the Chinese Personality Assessment Inventory (CPAI)*

Description

CPAI537 is part of a big survey study on martial satisfaction (Luo et al., 2008). The participants were 537 urban Chinese couples in the first year of their marriage. Included here are 28 composite scores of the CPAI (Cheung et al., 1996) for the 537 wives.

Usage

```
data(CPAI537)
```

Format

The format is a n by p matrix, where n is the number of participants (537) and p is the number of manifest variables (28).

Details

The column names stand for the following variable names:
Nov - Novelty
Div - Diversity
Dit - Diverse thinking
LEA - Leadership
L_A - Logical orientation vs affective orientation
AES - Aesthetics
E_I - Extroversion-Introversion
ENT - Enterprise
RES - Responsibility
EMO - Emotionality
I_S - Inferiority vs. self-acceptance
PRA - Practical mindedness
O_P - Optimistic vs. pessimistic
MET - Meticulousness
FAC - Face
I_E - Internal control vs. external control
FAM - Family orientation
DEF - Defensiveness
G_M - Graciousness vs. meanness
INT - Interpersonal tolerance
S_S - Self orientation vs. social orientation
V_S - Veraciousness vs. slickness
T_M - Traditionalism vs. modernity
REN - Relationship orientation
SOC - Social sensitivity

DIS - Discipline
HAR - Harmony
T_E - Thrift vs. extravagance

References

Cheung, F. M., Leung, K., Fan, R., Song, W., Zhang, J., & Zhang, J. (1996). Development of the Chinese Personality Assessment Inventory (CPAI). Journal of Cross-Cultural Psychology, 27 ,181-199.

Luo, S., Chen, H., Yue, G., Zhang, G., Zhaoyang, R., & Xu, D. (2008). Predicting marital satisfaction from self, partner, and couple characteristics: Is it me, you, or us? Journal of Personality, 76 ,1231-1266.

`efa` *Exploratory Factor Analysis*

Description

Performs exploratory factor analysis under a variety of conditions. In particular, it provides standard errors for rotated factor loadings and factor correlations for normal variables, nonnormal continuous variables, and Likert scale variables with and without model error.

Usage

```
efa(x=NULL, factors=NULL, covmat=NULL, acm=NULL, n.obs=NULL, dist='normal',
fm='ols', mtest = TRUE, rtype='oblique', rotation='CF-varimax', normalize=FALSE,
maxit=1000, geomin.delta=NULL, MTarget=NULL, MWeight=NULL,PhiWeight = NULL,
PhiTarget = NULL, useorder=FALSE, se='sandwich', LConfid=c(0.95,0.90),
CItype='pse', Ib=2000, mnames=NULL, fnames=NULL, merror='YES', wxt2 = 1e0)
```

Arguments

`x`	The raw data: an n-by-p matrix where n is number of participants and p is the number of manifest variables.
`factors`	The number of factors m: specified by the researcher; the default one is the Kaiser rule which is the number of eigenvalues of covmat larger than one.
`covmat`	A p-by-p manifest variable correlation matrix.
`acm`	A p(p-1)/2 by p(p-1)/2 asymptotic covariance matrix of correlations: specified by the researcher.
`n.obs`	The number of participants used in calculating the correlation matrix. This is not required when the raw data (x) is provided.
`dist`	Manifest variable distributions: 'normal'(default), 'continuous', 'ordinal' and 'ts'. 'normal' stands for normal distribution. 'continuous' stands for nonnormal continuous distributions. 'ordinal' stands for Likert scale variable. 'ts' stands for distributions for time-series data.

fm	Factor extraction methods: 'ols' (default) and 'ml'
mtest	Whether the test statistic is computed: TRUE (default) and FALSE
rtype	Factor rotation types: 'oblique' (default) and 'orthogonal'. Factors are correlated in 'oblique' rotation, and they are uncorrelated in 'orthogonal' rotation.
rotation	Factor rotation criteria: 'CF-varimax' (default), 'CF-quartimax', 'CF-equamax', 'CF-facparsim', 'CF-parsimax','target', and 'geomin'. These rotation criteria can be used in both orthogonal and oblique rotation. In addition, a fifth rotation criterion 'xtarget'(extended target) rotation is available for oblique rotation. The extended target rotation allows targets to be specified on both factor loadings and factor correlations.
normalize	Row standardization in factor rotation: FALSE (default) and TRUE (Kaiser standardization).
maxit	Maximum number of iterations in factor rotation: 1000 (default)
geomin.delta	The controlling parameter in Geomin rotation, 0.01 as the default value.
MTarget	The p-by-m target matrix for the factor loading matrix in target rotation and xtarget rotation.
MWeight	The p-by-m weight matrix for the factor loading matrix in target rotation and xtarget rotation. Optional
PhiWeight	The m-by-m target matrix for the factor correlation matrix in xtarget rotation. Optional
PhiTarget	The m-by-m weight matrix for the factor correlation matrix in xtarget rotation
useorder	Whether an order matrix is used for factor alignment: FALSE (default) and TRUE
se	Methods for estimating standard errors for rotated factor loadings and factor correlations, 'information', 'sandwich', 'bootstrap', and 'jackknife'. For normal variables and ml estimation, the default method is 'information'. For all other situations, the default method is 'sandwich'. In addition, the 'bootstrap' and 'jackknife' methods require raw data.
LConfid	Confidence levels for model parameters (factor loadings and factor correlations) and RMSEA, respectively: c(.95, .90) as default.
CItype	Type of confidence intervals: 'pse' (default) or 'percentile'. CIs with 'pse' are based on point and standard error estimates; CIs with 'percentile' are based on bootstrap percentiles.
Ib	The number of bootstrap samples when se='bootstrap': 2000 (default)
mnames	Names of p manifest variables: Null (default)
fnames	Names of m factors: Null (default)
merror	Model error: 'YES' (default) or 'NO'. In general, we expect our model is a parsimonious representation to the complex real world. Thus, some amount of model error is unavailable. When merror = 'NO', the efa model is assumed to fit perfectly in the population.
wxt2	The relative weight for factor correlations in 'xtarget' (extended target) rotation: 1 (default)

Details

The function efa conducts exploratory factor analysis (EFA) (Gorsuch, 1983) in a variety of conditions. Data can be normal variables, non-normal continuous variables, and Likert variables. Our implementation of EFA includes three major steps: factor extraction, factor rotation, and estimating standard errors for rotated factor loadings and factor correlations.

Factors can be extracted using two methods: maximum likelihood estimation (ml) and ordinary least squares (ols). These factor loading matrices are referred to as unrotated factor loading matrices. The ml unrotated factor loading matrix is obtained using factanal. The ols unrotated factor loading matrix is obtained using optim where the residual sum of squares is minimized. The starting values for communalities are squared multiple correlations (SMCs). The test statistic and model fit measures are provided.

Seven rotation criteria (CF-varimax, CF-quartimax, 'CF-equamax', 'CF-facparsim', 'CF-parsimax',geomin, and target) are available for both orthogonal rotation and oblique rotation (Browne, 2001). Additionally, a new rotation criteria, xtarget, can be specified for oblique rotation. The factor rotation methods are achieved by calling functions in the package GPArotation. CF-varimax, CF-quartimax, CF-equamax, CF-facparsim, and CF-parsimax are members of the Crawford-Fugersion family (Crawford, & Ferguson, 1970) whose kappa is 1/p, 0, m/2p, 1, and (m-1)/(p+m-2) respectively where p is the number of manifest variables and m is the number of factors. CF-varimax and CF-quartimax are equivalent to varimax and quartimax rotation in orthogonal rotation. The equivalence does not carry over to oblique rotation, however. Although varimax and quartimax often fail to give satisfactory results in oblique rotation, CF-varimax and CF-quartimax do give satisfactory results in many oblique rotation applications. CF-quartimax rotation is equivalent to direct oblimin rotation for oblique rotation. The target matrix in target rotation can either be a fully specified matrix or a partially specified matrix. Target rotation can be considered as a procedure which is located between EFA and CFA. In CFA, if a factor loading is specified to be zero, its value is fixed to be zero; if target rotation, if a factor loading is specified to be zero, it is made to zero as close as possible. In xtarget rotation, target values can be specified on both factor loadings and factor correlations.

Confidence intervals for rotated factor loadings and correlation matrices are constructed using point estimates and their standard error estimates. Standard errors for rotated factor loadings and factor correlations are computed using a sandwich method (Ogasawara, 1998; Yuan, Marshall, & Bentler, 2002), which generalizes the augmented information method (Jennrich, 1974). The sandwich standard error are consistent estimates even when the data distribution is non-normal and model error exists in the population. Sandwich standard error estimates require a consistent estimate of the asymptotic covariance matrix of manifest variable correlations. Such estimates are described in Browne & Shapiro (1986) for non-normal continuous variables and in Yuan & Schuster (2013) for Likert variables. Estimation of the asymptotic covariance matrix of polychoric correlations is slow if the EFA model involves a large number of Likert variables.

When manifest variables are normally distributed (dist = 'normal') and model error does not exist (merror = 'NO'), the sandwich standard errors are equivalent to the usual standard error estimates, which come from the inverse of the information matrix. The information standard error estimates in EFA is available CEFA (Browne, Cudeck, Tateneni, & Mels, 2010) and SAS Proc Factor. Mplus (Muthen & Muthen, 2015) also implemented a version of sandwich standard errors for EFA, which are robust against non-normal distribution but not model error. Sandwich standard errors computed in efa tend to be larger than those computed in Mplus. Sandwich standard errors for non-normal distributions and with model error are equivalent to the infinitesimal jackknife standard errors described in Zhang, Preacher, & Jennrich (2012). Two computationally intensive standard error methods (se='bootstrap' and se='jackknife') are also implemented. More details on standard error estimation methods in EFA are documented in Zhang (2014).

Value

An object of class efa, which includes:

`details`	summary information about the analysis such as number of manifest variables, number of factors, sample size, factor extraction method, factor rotation method, target values for target rotation and xtarget rotation, and levels for confidence intervals.
`unrotated`	the unrotated factor loading matrix
`fdiscrepancy`	discrepancy function value used in factor extraction
`convergence`	whether the factor extraction stage converged successfully, successful convergence indicated by 0
`heywood`	the number of heywood cases
`R0`	the sample correlation matrix
`Phat`	the model implied correlation matrix
`Residual`	the residual correlation matrix
`rotated`	the rotated factor loadings
`Phi`	the rotated factor correlations
`rotatedse`	the standard errors for rotated factor loadings
`Phise`	the standard errors for rotated factor correlations
`ModelF`	the test statistic and measures of model fit
`rotatedlow`	the lower bound of confidence levels for factor loadings
`rotatedupper`	the upper bound of confidence levels for factor loadings
`Philow`	the lower bound of confidence levels for factor correlations
`Phiupper`	the lower bound of confidence levels for factor correlations

Author(s)

Guangjian Zhang, Ge Jiang, Minami Hattori, and Lauren Trichtinger

References

Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. Multivariate Behavioral Research, 36, 111-150.

Browne, M. W., Cudeck, R., Tateneni, K., & Mels, G. (2010). CEFA 3.04: Comprehensive Exploratory Factor Analysis. Retrieved from http://faculty.psy.ohio-state.edu/browne/.

Browne, M. W., & Shapiro, A. (1986). The asymptotic covariance matrix of sample correlation coefficients under general conditions. Linear Algebra and its applications, 82, 169-176.

Crawford, C. B., & Ferguson, G. A. (1970). A general rotation criterion and its use in orthogonal rotation. Psychometrika, 35 , 321-332.

Engle, R. W., Tuholsjki, S.W., Laughlin, J.E., & Conway, A. R. A. (1999). Working memory, short-term memory, and general fluid intelligence: a latent-variable approach. Journal of Experimental Psychology: General, 309-331.

Gorsuch, R. L. (1983). Factor analysis (2nd ed.). Mahwah, NJ: Lawrence Erlbaum Associates.

Jennrich, R. I. (1974). Simplified formula for standard errors in maximum-likelihood factor analysis. British Journal of Mathematical and Statistical Psychology, 27, 122-131.

Jennrich, R. I. (2002). A simple general method for oblique rotation. Psychometrika, 67, 7-19.

Muthen, L. K., & Muthen, B. O. (1998-2015). Mplus user's guide (7th ed.). Los Angeles, CA: Muthen & Muthen.

Ogasawara, H. (1998). Standard errors of several indices for unrotated and rotated factors. Economic Review, Otaru University of Commerce, 49(1), 21-69.

Yuan, K., Marshall, L. L., & Bentler, P. M. (2002). A unified approach to exploratory factor analysis with missing data, nonnormal data, and in the presence of outliers. Psychometrika , 67 , 95-122.

Yuan, K.-H., & Schuster, C. (2013). Overview of statistical estimation methods. In T. D. Little (Ed.), The Oxford handbook of quantitative methods (pp. 361-387). New York, NY: Oxford University Press.

Zhang, G. (2014). Estimating standard errors in exploratory factor analysis. Multivariate Behavioral Research, 49, 339-353.

Zhang, G., Preacher, K. J., & Jennrich, R. I. (2012). The infinitesimal jackknife with exploratory factor analysis. Psychometrika, 77 , 634-648.

Examples

```
#Examples using the data sets included in the packages:

data("CPAI537")    # Chinese personality assessment inventory (N = 537)

#1a) normal, ml, oblique, CF-varimax, information, merror='NO'
res1 <- efa(x=CPAI537,factors=4, fm='ml')
res1

#1b) confidence intervals: normal, ml, oblique, CF-varimax, information, merror='NO'
#res1$rotatedlow     # lower bound for 95 percent confidence intervals for factor loadings
#res1$rotatedupper   # upper bound for 95 percent confidence intervals for factor loadings
#res1$Philow      # lower bound for 95 percent confidence intervals for factor correlations
#res1$Phiupper     # upper bound for 95 percent confidence intervals for factor correlations

#2) continuous, ml, oblique, CF-quartimax, sandwich, merror='YES'
#efa(x=CPAI537, factors=4, dist='continuous',fm='ml',rotation='CF-quartimax', merror='YES')

#3) continuous, ml, oblique, CF-equamax, sandwich, merror='YES'
#efa(x = CPAI537, factors = 4, dist = 'continuous',
#fm = 'ml', rotation = 'CF-equamax', merror ='YES')

#4) continuous, ml, oblique, CF-facparism, sandwich, merror='YES'
#efa(x = CPAI537, factors = 4, fm = 'ml',
#dist = 'continuous', rotation = 'CF-facparsim', merror='YES')
```

```
#5)continuous, ml, orthogonal, CF-parsimax, sandwich, merror='YES'
#efa(x = CPAI537, factors = 4, fm = 'ml', rtype = 'orthogonal',
#dist = 'continuous', rotation = 'CF-parsimax', merror = 'YES')

#6) continuous, ols, orthogonal, geomin, sandwich, merror='Yes'
#efa(x=CPAI537, factors=4, dist='continuous',
#rtype= 'orthogonal',rotation='geomin', merror='YES')

#7) ordinal, ols, oblique, CF-varimax, sandwich, merror='Yes'
#data("BFI228")      # Big-five inventory (N = 228)
# For ordinal data, estimating SE with the sandwich method
#   can take time with a dataset with 44 variables
#reduced2 <- BFI228[,1:17] # extracting 17 variables corresponding to the first 2 factors
#efa(x=reduced2, factors=2, dist='ordinal', merror='YES')

#8) continuous, ml, oblique, Cf-varimax, jackknife
#efa(x=CPAI537,factors=4, dist='continuous',fm='ml', merror='YES', se= 'jackknife')

#9) extracting the test statistic
#res2 <-efa(x=CPAI537,factors=4)
#res2
#res2$ModelF$f.stat

#10) extended target rotation, ml
# # The data come from Engle et al. (1999) on memory and intelligence.
# datcor <- matrix(c(1.00, 0.51, 0.47, 0.35, 0.37, 0.38, 0.28, 0.34,
#                    0.51, 1.00, 0.32, 0.35, 0.35, 0.31, 0.24, 0.28,
#                    0.47, 0.32, 1.00, 0.43, 0.31, 0.31, 0.29, 0.32,
#                    0.35, 0.35, 0.43, 1.00, 0.54, 0.44, 0.19, 0.27,
#                    0.37, 0.35, 0.31, 0.54, 1.00, 0.59, 0.05, 0.19,
#                    0.38, 0.31, 0.31, 0.44, 0.59, 1.00, 0.20, 0.21,
#                    0.28, 0.24, 0.29, 0.19, 0.05, 0.20, 1.00, 0.68,
#                    0.34, 0.28, 0.32, 0.27, 0.19, 0.21, 0.68, 1.00),
#                  ncol = 8)
#
# # Prepare target and weight matrices for lambda -------
# MTarget1 <- matrix(c(9, 0, 0,
#                      9, 0, 0,
#                      9, 0, 0, # 0 corresponds to targets
#                      0, 9, 0,
#                      0, 9, 0,
#                      0, 9, 0,
#                      0, 0, 9,
#                      0, 0, 9), ncol = 3, byrow = TRUE)
# MWeight1 <- matrix(0, ncol = 3, nrow = 8)
# MWeight1[MTarget1 == 0] <- 1 # 1 corresponds to targets
#
# # Prepare target and weight matrices for phi ---------
# PhiTarget1 <- matrix(c(1, 9, 9,
#                        9, 1, 0,
#                        9, 0, 1), ncol = 3)
# PhiWeight1 <- matrix(0, ncol = 3, nrow = 3)
# PhiWeight1[PhiTarget1 == 0] <- 1
```

```
#
# # Conduct extended target rotation -------------------
# mod.xtarget <- efa(covmat = datcor, factors = 3, n.obs = 133,
#                   rotation ='xtarget', fm = 'ml', useorder = T,
#                   MTarget = MTarget1, MWeight = MWeight1,
#                   PhiTarget = PhiTarget1, PhiWeight = PhiWeight1)
# mod.xtarget
#
```

efaMR *Exploratory Factor Analysis with Multiple Rotations*

Description

The function compares EFA solutions from multiple random starts or from multiple rotation criteria.

Usage

```
efaMR(x=NULL, factors=NULL, covmat=NULL, n.obs=NULL,
      dist='normal', fm='ols', rtype='oblique', rotation = 'CF-varimax',
      input.A=NULL, additionalRC = NULL,
      nstart = 100, compare = 'First', plot = T, cex = .5,
      normalize = FALSE, geomin.delta = .01,
      MTarget = NULL, MWeight = NULL, PhiTarget = NULL, PhiWeight = NULL,
      useorder = FALSE, mnames = NULL, fnames = NULL, wxt2 = 1)
```

Arguments

`x`	The raw data: an n-by-p matrix where n is number of participants and p is the number of manifest variables.
`factors`	The number of factors m: specified by a researcher; the default one is the Kaiser rule which is the number of eigenvalues of covmat larger than one.
`covmat`	A p-by-p manifest variable correlation matrix.
`n.obs`	The number of participants used in calculating the correlation matrix. This is not required when the raw data (x) is provided.
`dist`	Manifest variable distributions: 'normal'(default), 'continuous', 'ordinal' and 'ts'. 'normal' stands for normal distribution. 'continuous' stands for nonnormal continuous distributions. 'ordinal' stands for Likert scale variable. "ts" stands for distributions for time-series data.
`fm`	Factor extraction methods: 'ols' (default) and 'ml'
`rtype`	Factor rotation types: 'oblique' (default) and 'orthogonal'. Factors are correlated in 'oblique' rotation, and they are uncorrelated in 'orthogonal' rotation.

`rotation`	Factor rotation criteria: 'CF-varimax' (default), 'CF-quartimax', 'CF-equamax', 'CF-facparsim', 'CF-parsimax','target', and 'geomin'. These rotation criteria can be used in both orthogonal and oblique rotation. In addition, a fifth rotation criterion 'xtarget'(extended target) rotation is available for oblique rotation. The extended target rotation allows targets to be specified on both factor loadings and factor correlations.
`input.A`	A p-by-m unrotated factor loading matrix. It can replace x or covmat as input arguments. Only factor rotation will be conducted; factor extraction will not be conducted.
`additionalRC`	A string of factor extraction methods against which the main rotation is compared. Required only when nstart = 1. See details.
`nstart`	The number random orthogonal starts used, with 100 as the default value. With nstart = 1, only one random start is used. See details.
`compare`	'First' (default) or 'All': The global solution is compared against all local solutions with 'First'; All solutions are compared with each other with 'All'.
`plot`	Whether a bar graph that shows the number and frequencies of local solutions or not: TRUE (default) and FALSE.
`cex`	A tuning parameter if the plot is produced: .5 (default)
`normalize`	Row standardization in factor rotation: FALSE (default) and TRUE (Kaiser standardization).
`geomin.delta`	The controlling parameter in Geomin rotation, 0.01 as the default value.
`MTarget`	The p-by-m target matrix for the factor loading matrix in target rotation or xtarget rotation.
`MWeight`	The p-by-m weight matrix for the factor loading matrix in target rotation or xtarget rotation.
`PhiTarget`	The m-by-m target matrix for the factor correlation matrix in xtarget rotation.
`PhiWeight`	The m-by-m weight matrix for the factor correlation matrix in xtarget rotation.
`useorder`	Whether an order matrix is used for factor alignment: FALSE (default) and TRUE
`mnames`	Names of p manifest variables: Null (default)
`fnames`	Names of m factors: Null (default)
`wxt2`	The relative weight for factor correlations in 'xtarget' (extended target) rotation: 1 (default)

Details

efaMR performs EFA with multiple rotation using random starts.

Geomin rotation, in particular, is known to produce multiple local solutions; the use of random starts is advised (Hattori, Zhang, & Preacher, 2018).

The p-by-m unrotated factor loading matrix is post-multiplied by an m-by-m random orthogonal matrices before rotation.

The number of random starts can be specified with the default value of nstart = 100. Bar plot that represents frequencies of each solution is provided. If multiple solutions are found, they are compared with each other using congruence coefficient.

If nstart = 1, no random start is used. The solution is compared against solutions using additional rotation criterion provided by additionalRC.

For example, with rotation = geomin, additionalRC = c('CF-varimax', 'CF-quartimax), the geomin solution is compared against those with CF-varimax and CF-quartimax.

Estimation of standard errors and construction of confidence intervals are disabled with the function efaMR(). They are available with a function efa().

Author(s)

Minami Hattori, Guangjian Zhang

References

Hattori, M., Zhang, G., & Preacher, K. J. (2017). Multiple local solutions and geomin rotation. Multivariate Behavioral Research, 720–731. doi: 10.1080/00273171.2017.1361312

Examples

```
#data("CPAI537")    # Chinese personality assessment inventory (N = 537)

# # Example 1: Oblique geomin rotation with 10 random starts
# res1 <- efaMR(CPAI537, factors = 5, fm = 'ml',
#               rtype = 'oblique', rotation = 'geomin',
#               geomin.delta = .01, nstart = 10)
# res1
# summary(res1)
# res1$MultipleSolutions
# res1$Comparisons

# In practice, we recommend nstart = 100 or more (Hattori, Zhang, & Preacher, 2018).


# Example 2: Oblique geomin rotation (no random starts)
#            compared against CF-varimax and CF-quartimax rotation solutions
# res2 <- efaMR(CPAI537, factors = 5, fm = 'ml',
#               rtype = 'oblique', rotation = 'geomin',
#               additionalRC = c('CF-varimax', 'CF-quartimax'),
#               geomin.delta = .01, nstart = 1)
# res2$MultipleSolutions
# res2$Comparisons


# Example 3: Obtaining multiple solutions from the unrotated factor loading matrix as input
# res3 <- efa(CPAI537, factors = 5, fm = 'ml',
#             rtype = 'oblique', rotation = 'geomin')
# set.seed(2017)
# res3MR <- efaMR(input.A = res3$unrotated, rtype = 'oblique',
#                 rotation = 'geomin', geomin.delta = .01)
# res3MR$MultipleSolutions
# res3MR$Comparisons
```

PolychoricRM — *Estimate polychoric correlations and their asymptotic covariance matrices*

Description

The function is to estimate polychoric correlations and their asymptotic covariance matrix (ACM). A continuous response variable is assumed to underlie each ordinal variable (e.g., Likert variables). Polychoric correlations measure the associations between continuous response variables although only ordinal variables are directly measured. Note that polychoric correlations are tetrachoric correlations, when the ordinal data are binary. The ACM of polychoric correlations facilitates estimating standard errors and assessing test statistics for factor analysis and SEM with ordinal variables. Note that estimating the ACM requires a large sample size. The main implementation is done in Fortran 95 and the Fortran code is linked to R through wrapper functions.

Usage

```
PolychoricRM(iRaw=NULL, IAdjust=0, NCore=2, estimate.acm=FALSE)
```

Arguments

`iRaw`	The raw data: a n by p matrix where n is the number of participants and p is the number of manifest variables. Since the data are ordinal variables, all the element of the matrix are integers. In addition, the function deals with ordinal variables with 10 or fewer categories.
`IAdjust`	Methods to adjust for empty cells: a scalar where 0 is no adjustment is done (default), 1 adds 1/(nc*nr) to all cells where nc and nr is the number of columns and rows of the contingency table respectively, 2 adds 0.1 to all cells, 3 adds 0.5 to all cells, 11 adds 1/(nc*nr) to only zero cells, 12 adds 0.1 to only zero cells, and 13 adds 0.5 to only zero cells
`NCore`	Number of threads to be utilized: a scalar specified by the researcher; default (2)
`estimate.acm`	Estimate the ACM: FALSE (default) and TRUE

Details

The polychoric correlations are computed using a two-stage procedure described by Olsson(1979). The first stage is to estimate thresholds from univariate Normal distributions. The second stage is to estimate polychoric correlations from contingency tables formed with pairs of ordinal variables while threshold estimates are fixed as those obtained at the first stage. Estimating the thresholds at the first stage is closed-form one. Estimating the polychoric correlations at the second stage has to be done iteratively. We used a scoring method to obtain the estimate. Note that the second stage is a one-dimensional optimization problem and the problem is well conditioned in most cases. Our experience suggests that the solution converges in several iterations. In contrast, Olsson(1979) also described a one-stage procedure in which the polychoric correlations and thresholds are estimated simultaneously from the contingency table (it is a multiple dimensional optimization problem which is much more difficult to solve than a one-dimension problem). The two-stage method is often preferred due to 1) it is computationally more efficient than the one-stage method 2) the polychoric correlation estimates produced by these two methods are very close 3) the one-stage method involves the undesirable property that threshold estimates of a variable can vary when it pairs with different other variables.

Estimating polychoric correlations involves evaluating CDF functions of univariate and bivariate Normal distributions. We utilize Alan Miller's Fortran code (phi) for evaluating the CDF function of univariate Normal distributions. All Alan Millers' code has been released to the public domain. More details can be found at https://jblevins.org/mirror/amiller/.

We utilize Alan Genz and colleagues' Fortran code (bvn and bvnu) for evaluating the CDF function of bivariate Normal distributions. The code was extracted from the website http://www.math.wsu.edu/faculty/genz/software/s He allowed redistribution of the code either in the source form or compiled form, but the following information needs to be included in the distribution. Below please find the license information about Alan Genz's software.

In addition, both the aforementioned Alan Miller's and Alan Genz's Fortran code are included in the R package mnormt. Its license is GPL-2 | GPL-3, which is given at https://www.gnu.org/licenses/gpl-3.0.en.html. All other code (Fortran, C, and R) we prepared for the package is also distributed under the GPL-2 and GPL-3.

One issue in estimating polychoric correlations is how to deal with empty cells in contingency tables. Contingency table often contain empty cells particularly when the sample size is small and the number of categories is beyond two or three. Researchers often add a positive constant to these cells (or all cells) of the contingency table. Three popular choices of the positive constant are .1, .5, and 1/(nc*nr) where nc and nr is the number of columns and rows of the contingency table respectively. Some researchers (Savalei, 2011) prefer to make no adjustment to the empty cells at all when the number of categories is 3 or more. Note that adding a constant will introduce some additional bias to the polychoric correlation estimates but it makes the estimation process more stable. Our experience suggests polychoric correlations estimated with and without the added constant are close particularly when the constant is small and the sample size is large.

Estimating polychoric correlations (and particularly their ACM) is computationally intensive. Because multiple core (threads) CPUs are widely available, we can leverage the capability to reduce the computation time. We utilized openmp to improve the computational efficiency of the Fortran code. The benefits of parallel computing depends on different hardware and software setups. Our experience suggests that the Fortran code runs much more efficiently on Mac OS than Windows with comparable CPUs (I5-8257U for Mac OS vs I7-8650U for Windows) when only one core is used (.055 seconds vs .206 seconds for an dataset with 44 five-point Likert variables and 228 participants). Parallel computing improved the computational efficiency even further. The peak performance is achieved at 5 threads under Mac OS (.015 seconds) and the peak performance is achieved at 8 threads under Windows (.04 seconds). Note that both CPUS are equipped with 4 cores and 8 threads.

We compute the ACM of polychoric correlations according to the method described in Joreskog (1994). A key feature of the method is to estimate four-way contingency tables directly from raw data. Because a four-way contingency tables includes many cells (e.g., 625 cells for a four-way table of five-point Likert variables), its accurate estimation requires a larger sample. These ACM estimates are asymptotically equivalent to the estimates described by Muthen (1984). More recently, Monroe (2018) described a simulation based method to estimate the ACM.

Value

`threshold`	A 11 by p matrix. Its jth column contains estimates of the threshold for the jth variable. The lowest (the first) threshold estimate is -1^10 and the highest (the (c+1)th) threshold estimate is 1^10 where c is the number of categories for the variable.
`correlation`	The p by p polychoric correlation matrix. Unlike a Pearson correlation matrix, the polychoric correlation matrix may or may not be positive definite.
`flag`	A 2 by p(p-1)/2 matrix of integers. This matrix provides additional information on estimating the polychoric correlations from the contingency table. The first row indicates whether the contingency table contains an empty cell (0 indicates no empty cells and 1 indicates at least one empty cell) and the second row indicates the number of iterations it took to converge
`ACM`	A p(p-1)/2 by p(p-1)/2 symmetric matrix. Computing the matrix is more expensive than estimating the polychoric correlation matrix. Therefore, the output is an optional one and it is only computed upon request. Note the p(p-1)/2 non-duplicated polychoric correlations are arranged in the following way r_12, r_13, r_23, r_14, r_24, ..., r_(p-2)p, r_(p-1)p

Author(s)

Guangjian Zhang

References

Joreskog, K. G. (1994). On the estimation of polychoric correlations and their asymptotic variance matrix. Psychometrika, 59, 381-389. doi: 10.1007/bf02296131

Monroe, S. (2018). Contributions to estimation of polychoric correlations. Multivariate Behavioral Research, 53, 247–266. doi: 10.1080/00273171.2017.1419851

Muthen, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. Psychometrika, 49, 115-132. doi: 10.1007/bf02294210

Olsson, U. (1979). Maximum Likelihood estimation of the polychoric correlation coefficient. Psychometrika, 44, 443-460. doi: 10.1007/bf02296207

Savalei, V. (2011). What to do about zero frequency cells when estimating polychoric correlations. Structural Equation Modeling: A Multidisciplinary Journal, 18, 253–273. doi: 10.1080/10705511.2011.557339

Examples

```
#Examples using the data sets included in the packages:

data("BFI228")    # Big-five inventory (N = 228)

#For ordinal data, estimating the polychoric correlation and its ACM
#with 5 cores and 1/(nc*nr) added to all cells

polyACM = PolychoricRM(BFI228,NCore=5, IAdjust=1, estimate.acm=TRUE)
```

---

ssem *Saturated Structural Equation Models*

Description

This function fits saturated Structural equation models (SSEM) under a variety of conditions. SSEM re-parameterizes the obliquely rotated factor correlation matrix such that factors can be either endogenous or exogenous. In comparison, all factors are exogenous in exploratory factor analysis. Manifest variables can be normal variables, nonnormal variables, nonnormal continuous variable, Likert scale variables and time series. It also provides standard errors and confidence intervals for rotated factor loadings and structural parameters.

Usage

```
ssem(x=NULL, factors=NULL, exfactors=1, covmat=NULL,
acm=NULL, n.obs=NULL, dist='normal', fm='ml', mtest = TRUE,
rotation='semtarget', normalize=FALSE, maxit=1000, geomin.delta=NULL,
MTarget=NULL, MWeight=NULL, BGWeight = NULL, BGTarget = NULL,
PhiWeight = NULL, PhiTarget = NULL, useorder=TRUE, se='sandwich',
LConfid=c(0.95,0.90), CItype='pse', Ib=2000, mnames=NULL, fnames=NULL,
merror='YES', wxt2 = 1e0)
```

Arguments

x	The raw data: an n-by-p matrix where n is number of participants and p is the number of manifest variables.
factors	The number of factors m: specified by a researcher; the default one is the Kaiser rule which is the number of eigenvalues of covmat larger than one.
exfactors	The number of exogenous factors: 1 (default)

`covmat`	A p-by-p manifest variable correlation matrix.
`acm`	A p(p-1)/2 by p(p-1)/2 asymptotic covariance matrix of correlations: specified by the researcher.
`n.obs`	The number of participants used in calculating the correlation matrix. This is not required when the raw data (x) is provided.
`dist`	Manifest variable distributions: 'normal'(default), 'continuous', 'ordinal' and 'ts'. 'normal' stands for normal distribution. 'continuous' stands for nonnormal continuous distributions. 'ordinal' stands for Likert scale variable. 'ts' stands for distributions for time-series data.
`fm`	Factor extraction methods: 'ml' (default) and 'ols'
`mtest`	Whether the test statistic is computed: TRUE (default) and FALSE
`rotation`	Factor rotation criteria: 'semtarget' (default),'CF-varimax', 'CF-quartimax', 'CF-equamax', 'CF-parsimax', 'CF-facparsim','target', and 'geomin'. These rotation criteria can be used in both orthogonal and oblique rotation. In addition, a fifth rotation criterion 'xtarget'(extended target) rotation is available for oblique rotation. The ssem target rotation allows targets to be specified on both factor loadings and factor structural parameters.
`normalize`	Row standardization in factor rotation: FALSE (default) and TRUE (Kaiser standardization).
`maxit`	Maximum number of iterations in factor rotation: 1000 (default)
`geomin.delta`	The controlling parameter in Geomin rotation, 0.01 as the default value.
`MTarget`	The p-by-m target matrix for the factor loading matrix in target rotation and semtarget rotation.
`MWeight`	The p-by-m weight matrix for the factor loading matrix in target rotation and semtarget rotation. Optional
`BGWeight`	The m1-by-m weight matrix for the [Beta | Gamma] matrix in semtarget rotation (see details) Optional
`BGTarget`	The m1-by-m target matrix for the [Beta | Gamma] matrix in semtarget rotation where m1 is the number of endogenous factors (see details)
`PhiWeight`	The m2-by-m2 target matrix for the exogenous factor correlation matrix in semtarget rotation.Optional
`PhiTarget`	The m2-by-m2 weight matrix for the exogenous factor correlation matrix in semtarget rotation
`useorder`	Whether an order matrix is used for factor alignment: TRUE (default) and FALSE
`se`	Methods for estimating standard errors for rotated factor loadings and factor correlations, 'sandwich' (default),'information', 'bootstrap', and 'jackknife'. The 'bootstrap' and 'jackknife' methods require raw data.
`LConfid`	Confidence levels for model parameters (rotated factor loadings and structural parameters) and RMSEA, respectively: c(.95, .90) as default.
`CItype`	Type of confidence intervals: 'pse' (default) or 'percentile'. CIs with 'pse' are based on point and standard error estimates; CIs with 'percentile' are based on bootstrap percentiles.

`Ib`	The Number of bootstrap samples when se='bootstrap': 2000 (default)
`mnames`	Names of p manifest variables: Null (default)
`fnames`	Names of m factors: Null (default)
`merror`	Model error: 'YES' (default) or 'NO'. In general, we expect our model is a parsimonious representation to the complex real world. Thus, some amount of model error is unavailable. When merror = 'NO', the ssem model is assumed to fit perfectly in the population.
`wxt2`	The relative weight for structural parameters in 'semtarget' rotation: 1 (default)

Details

The function `ssem` conducts saturated structural equation modeling (ssem) in a variety of conditions. Data can be normal variables, non-normal continuous variables, and Likert variables. Our implementation of SSEM includes three major steps: factor extraction, factor rotation, and estimating standard errors for rotated factor loadings and factor correlations.

Factors can be extracted using two methods: maximum likelihood estimation (ml) and ordinary least squares (ols). These factor loading matrices are referred to as unrotated factor loading matrices. The ml unrotated factor loading matrix is obtained using `factanal`. The ols unrotated factor loading matrix is obtained using `optim` where the residual sum of squares is minimized. The starting values for communalities are squared multiple correlations (SMCs). The test statistic and model fit measures are provided.

Eight rotation criteria (semtarget, CF-varimax, CF-quartimax, CF-equamax, CF-parsimax, CF-facparsim, target, and geomin) are available for oblique rotation (Browne, 2001). Additionally, a new rotation criteria, ssemtarget, can be specified for oblique rotation. The factor rotation methods are achieved by calling functions in the package GPArotation. CF-varimax, CF-quartimax, CF-equamax, CF-parsimax, and CF-facparsim are members of the Crawford-Fugersion family (Crawford, & Ferguson, 1970) whose kappa = 1/p and kappa = 0, respectively. The target matrix in target rotation can either be a fully specified matrix or a partially specified matrix. Target rotation can be considered as a procedure which is located between EFA and CFA. In CFA, if a factor loading is specified to be zero, its value is fixed to be zero; if target rotation, if a factor loading is specified to be zero, it is made to zero as close as possible. In xtarget rotation, target values can be specified on both factor loadings and factor correlations. In ssemtarget, target values can be specified for the [Beta | Gamma] matrix where Beta is the regression weights of the endogenous factors on itself and the Gamma is the regression weights of the endogenous factors on the exogenous factors.

Confidence intervals for rotated factor loadings and correlation matrices are constructed using point estimates and their standard error estimates. Standard errors for rotated factor loadings and factor correlations are computed using a sandwich method (Ogasawara, 1998; Yuan, Marshall, & Bentler, 2002), which generalizes the augmented information method (Jennrich, 1974). The sandwich standard error are consistent estimates even when the data distribution is non-normal and model error exists in the population. Sandwich standard error estimates require a consistent estimate of the asymptotic covariance matrix of manifest variable correlations. Such estimates are described in Browne & Shapiro (1986) for non-normal continuous variables and in Yuan & Schuster (2013) for Likert variables. Estimation of the asymptotic covariance matrix of polychoric correlations is slow if the EFA model involves a large number of Likert variables.

When manifest variables are normally distributed (dist = 'normal') and model error does not exist (merror = 'NO'), the sandwich standard errors are equivalent to the usual standard error estimates, which come from the inverse of the information matrix. The information standard error estimates in EFA is available CEFA (Browne, Cudeck, Tateneni, & Mels, 2010) and SAS Proc Factor. Mplus (Muthen & Muthen, 2015) also implemented a version of sandwich standard errors for EFA, which are robust against non-normal distribution but not model error. Sandwich standard errors computed in efa tend to be larger than those computed in Mplus. Sandwich standard errors for non-normal distributions and with model error are equivalent to the infinitesimal jackknife standard errors described in Zhang, Preacher, & Jennrich (2012). Two computationally intensive standard error methods (se='bootstrap' and se='jackknife') are also implemented. More details on standard error estimation methods in EFA are documented in Zhang (2014).

Value

An object of class ssem, which includes:

`details` summary information about the analysis such as number of manifest variables, number of factors, number of endogenous factors, number of exogenous factors, sample size, distribution, factor extraction method, factor rotation method, target values for target rotation, xtarget rotation and ssemtarget rotation, and levels for confidence intervals.

`unrotated` the unrotated factor loading matrix

`fdiscrepancy` discrepancy function value used in factor extraction

`convergence` whether the factor extraction stage converged successfully, successful convergence indicated by 0

`heywood` the number of heywood cases

`nq` the number of effective parameters

`compsi` contains eigenvalues, SMCs, communalities, and unique variances

`R0` the sample correlation matrix

`Phat` the model implied correlation matrix

`Residual` the residual correlation matrix

`rotated` the rotated factor loadings

`Phi` the rotated factor correlations

`BG` the [Beta | Gamma] latent regression coefficients

`psi` the endogenous residuals

`Phi.xi` the exogenous correlation

`rotatedse` the standard errors for rotated factor loadings

`Phise` the standard errors for rotated factor correlations

`BGse` the standard errors for the [Beta | Gamma] latent regression coefficients

`psise` the standard errors for the endogenous residuals

`Phi.xise` the standard errors for the exogenous correlation

`ModelF` the test statistic and measures of model fit

`rotatedlow` the lower bound of confidence levels for factor loadings

`rotatedupper` the upper bound of confidence levels for factor loadings

`Philow` the lower bound of confidence levels for factor correlations

Phiupper	the lower bound of confidence levels for factor correlations
BGlower	the lower bound of the [Beta | Gamma] latent regression coefficients
BGupper	the upper bound of the [Beta | Gamma] latent regression coefficients
psilower	the lower bound of the endogenous residuals
psiupper	the upper bound of the endogenous residuals
Phixilower	the lower bound of the exogenous correlation
Phixiupper	the upper bound of the exogenous correlation

Author(s)

Guangjian Zhang, Minami Hattori, and Lauren Trichtinger

References

Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. Multivariate Behavioral Research, 36, 111-150.

Browne, M. W., Cudeck, R., Tateneni, K., & Mels, G. (2010). CEFA 3.04: Comprehensive Exploratory Factor Analysis. Retrieved from http://faculty.psy.ohio-state.edu/browne/.

Browne, M. W., & Shapiro, A. (1986). The asymptotic covariance matrix of sample correlation coefficients under general conditions. Linear Algebra and its applications, 82, 169-176.

Crawford, C. B., & Ferguson, G. A. (1970). A general rotation criterion and its use in orthogonal rotation. Psychometrika, 35 , 321-332.

Engle, R. W., Tuholsjki, S.W., Laughlin, J.E., & Conway, A. R. A. (1999). Working memory, short-term memory, and general fluid intelligence: a latent-variable approach. Journal of Experimental Psychology: General, 309-331.

Gorsuch, R. L. (1983). Factor analysis (2nd ed.). Mahwah, NJ: Lawrence Erlbaum Associates.

Jennrich, R. I. (1974). Simplified formula for standard errors in maximum-likelihood factor analysis. British Journal of Mathematical and Statistical Psychology, 27, 122-131.

Jennrich, R. I. (2002). A simple general method for oblique rotation. Psychometrika, 67, 7-19.

Muthen, L. K., & Muthen, B. O. (1998-2015). Mplus user's guide (7th ed.). Los Angeles, CA: Muthen & Muthen.

Ogasawara, H. (1998). Standard errors of several indices for unrotated and rotated factors. Economic Review, Otaru University of Commerce, 49(1), 21-69.

Yuan, K., Marshall, L. L., & Bentler, P. M. (2002). A unified approach to exploratory factor analysis with missing data, nonnormal data, and in the presence of outliers. Psychometrika , 67 , 95-122.

Yuan, K.-H., & Schuster, C. (2013). Overview of statistical estimation methods. In T. D. Little (Ed.), The Oxford handbook of quantitative methods (pp. 361-387). New York, NY: Oxford University Press.

Zhang, G. (2014). Estimating standard errors in exploratory factor analysis. Multivariate Behavioral Research, 49, 339-353.

Zhang, G., Preacher, K. J., & Jennrich, R. I. (2012). The infinitesimal jackknife with exploratory factor analysis. Psychometrika, 77 , 634-648.

Examples

```
#cormat <- matrix(c(1, .865, .733, .511, .412, .647, -.462, -.533, -.544,
#                  .865, 1, .741, .485, .366, .595, -.406, -.474, -.505,
#                  .733, .741, 1, .316, .268, .497, -.303, -.372, -.44,
#                  .511, .485, .316, 1, .721, .731, -.521, -.531, -.621,
#                  .412, .366, .268, .721, 1, .599, -.455, -.425, -.455,
#                  .647, .595, .497, .731, .599, 1, -.417, -.47, -.521,
#                 -.462, -.406, -.303, -.521, -.455, -.417, 1, .747, .727,
#                  -.533, -.474, -.372, -.531, -.425, -.47, .747, 1, .772,
#                  -.544, -.505, -.44, -.621, -.455, -.521, .727, .772, 1),
#                ncol = 9)


#p <- 9      # a number of manifest variables

#m <- 3      # a total number of factors

#m1 <- 2     # a number of endogenous variables
#N <- 138    # a sample size

#mvnames <- c("H1_likelihood", "H2_certainty", "H3_amount", "S1_sympathy",
#             "S2_pity", "S3_concern", "C1_controllable", "C2_responsible", "C3_fault")

#fnames <- c('H', 'S', 'C')
# Step 2: Preparing target and weight matrices ========================
# a 9 x 3 matrix for lambda; p = 9, m = 3

#MT <- matrix(0, p, m, dimnames = list(mvnames, fnames))

#MT[c(1:3,6),1] <- 9

#MT[4:6,2] <- 9

#MT[7:9,3] <- 9

#MW <- matrix(0, p, m, dimnames = list(mvnames, fnames))

#MW[MT == 0] <- 1

# a 2 x 3 matrix for [B|G]; m1 = 2, m = 3

# m1 = 2
#BGT <- matrix(0, m1, m, dimnames = list(fnames[1:m1], fnames))

#BGT[1,2] <- 9

#BGT[2,3] <- 9

#BGT[1,3] <- 9

#BGW <- matrix(0, m1, m, dimnames = list(fnames[1:m1], fnames))
```

```
#BGW[BGT == 0] <- 1

#BGW[,1] <- 0

#BGW[2,2] <- 0
# a 1 x 1 matrix for Phi.xi; m - m1 = 1 (only one exogenous factor)

#PhiT <- matrix(9, m - m1, m - m1)

#PhiW <- matrix(0, m - m1, m - m1)
#SSEMres <- ssem(covmat = cormat, factors = m, exfactors = m - m1,
#                dist = 'normal', n.obs = N, fm = 'ml', rotation = 'semtarget',
#                maxit = 10000,
#                MTarget = MT, MWeight = MW, BGTarget = BGT, BGWeight = BGW,
#               PhiTarget = PhiT, PhiWeight = PhiW,  useorder = TRUE, se = 'information',
#                mnames = mvnames, fnames = fnames)
#
```

Index