# Package 'wrapFA'

June 6, 2022

**Type** Package

**Title** A Wrapper for Factor Analysis using lavaan and MplusAutomation

**Version** 0.0.1

**Date** 2022-31-05

**Description** The package provides a simple and coherent syntax to facilitate the implementation of several factor analytic techniques using either the lavaan package (Rosseel, 2012, <doi:10.18637/jss.v048.i02>) or the MplusAutomation package (Hallquist & Wiley, 2018, <doi:10.1080/10705511.2017.1402334>). The package includes confirmatory factor analysis (CFA) with different sequential modification procedures (e.g., modification indices and the Saris procedure; Saris et al., 2009, <doi:10.1080/10705510903203433>), exploratory factor analysis (EFA) with different rotation criteria (e.g., target, RETAM; Lorenzo-Seva & Ferrando, 2020, <doi:10.3758/s13428-019-01209-1>), Bayesian structural equation modeling (BSEM; Muthén & Asparouhov, 2012, <doi:10.1037/a0026802>), and EFA-based CFA (ECFA; Nájera et al., 2022). It also contains a bootstrapping procedure to evaluate the stability and replicability of factor solutions (Nájera et al., 2022).

**License** GPL-3

**LazyData** FALSE

**Depends** R (>= 3.6.0)

**Imports** lavaan (>= 0.6-8), MplusAutomation (>= 0.8), psych, foreach

**URL** https://github.com/pablo-najera/wrapFA

**BugReports** https://github.com/pablo-najera/wrapFA/issues

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**Author** Pablo Nájera [aut, cre, cph],
Francisco J. Abad [aut, cph],
Miguel A. Sorrel [aut, cph]

**Maintainer** Pablo Nájera <pablo.najera@uam.es>

**Collate** 'BSEM.R'
'CFA.R'
'ECFA.R'
'EFA.R'
'bootFA.R'
'detFA.R'
'helper.R'
'modFA.R'
'print.wrapFA.R'

# R topics documented:



---

`bootFA` *Nonparametric bootstrap to evaluate the stability of the internal structure*

---

## Description

Evaluates the stability of a factor analysis solution (e.g., CFA, EFA, ECFA, BSEM) by conducting nonparametric bootstrapping, resampling with a replacement from the original dataset. This is used in Nájera et al. (2022), based on the bootstrapping EFA (Christensen & Golino, 2021).

## Usage

```
bootFA(
  fit,
  R = 100,
  n.cores = 2,
  plots = TRUE,
  rm.files = TRUE,
  seed = NULL,
  verbose = TRUE,
  digits = 3
)
```

## Arguments

| | |
|---|---|
| `fit` | A `wrapFA` object or a `list` of `wrapFA` objects. |
| `R` | A `double` indicating the number of bootstrapping replications. Default is 100. |
| `n.cores` | A `double` indicating the number of CPU processors to be used for the bootstrapping. Default is 2. |
| `plots` | A `logical` indicating whether result plots should be generated. Default is `TRUE`. |
| `rm.files` | A `logical` indicating whether the stored files should be deleted. The default is `TRUE`. |
| `seed` | A `double` containing a random seed. Default is `NULL`. |
| `verbose` | A `logical` indicating whether a progress bar should be printed. Default is `TRUE`. |

## Value

bootFA returns an object of class bootFA.

lambdaCC Average congruent coefficient of factor loadings between pairs of bootraspping solutions (matrix).

lambdaMAD Average mean absolute deviation of factor loadings between pairs of bootraspping solutions (matrix).

phiCC Average congruent coefficient of factor correlations between pairs of bootraspping solutions (matrix).

phiMAD Average mean absolute deviation of factor correlations between pairs of bootraspping solutions (matrix).

lambdaCC.r Congruent coefficient of factor loadings between pairs of bootraspping solutions (list).

lambdaMAD.r Mean absolute deviation of factor loadings between pairs of bootraspping solutions (list).

phiCC.r Congruent coefficient of factor correlations between pairs of bootraspping solutions (list).

phiMAD.r Mean absolute deviation of factor correlations between pairs of bootraspping solutions (list).

plots Plots summarizing the results. Only printed if plots = TRUE (list).

specifications Specifications used to run the function (list).

## References

Christensen, A. P., & Golino, H. (2021). Estimating the stability of the number of factors via bootstrap exploratory graph analysis: A tutorial. *Psych*, *3*(3), 479-500. https://doi.org/10.3390/psych3030032
Nájera, P., Abad, F. J., & Sorrel, M. A. (2022). Is EFA always to be preferred? A systematic comparison of factor analytic techniques throughout the confirmatory-exploratory continuum. *Manuscript submitted for publication*.

## Examples

```
library(MBESS)
data(HS)
HS <- HS[, -c(1:8)]
colnames(HS) <- paste0("t", 1:26)
modHS <- data.frame(spatial = c(rep(1, 4), rep(0, 20), rep(1, 2)),
                    verbal = c(rep(0, 4), rep(1, 5), rep(0, 17)),
                    speed = c(rep(0, 9), rep(1, 4), rep(0, 13)),
                    memory = c(rep(0, 13), rep(1, 6), rep(0, 7)),
                   maths = c(rep(0, 19), rep(1, 5), rep(0, 2)), row.names = colnames(HS))
modHS.lav <- modFA(lambda = modHS, software = "lavaan", keep.names = TRUE)
cfa.lav <- CFA(model = modHS.lav$CFA, data = HS)
boot.cfa <- bootFA(fit = cfa.lav, n.cores = 4)
```

BSEM *Bayesian structural equation modeling (BSEM)*

## Description

Fit a BSEM (BSEM; Muthén & Asparouhov, 2012) via *MplusAutomation*.

## Usage

```
BSEM(
  model,
  data,
  categorical = NULL,
  mplus.path = NULL,
  rm.files = FALSE,
  max.iter = 60000,
  n.cores = 1,
  n.chains = 2,
  suppressMessages = TRUE
)
```

## Arguments

| | |
|---|---|
| model | A list containing two character elements: the user-specified BSEM model and the priors specification using the *Mplus* syntax . |
| data | A *N* individuals x *J* observed variables matrix or data.frame. Missing values need to be coded as NA. |
| categorical | A vector containing the number or name of the observed variables that need to be treated as categorical. Default is NULL, which treates all observed variables as continuous. |
| mplus.path | A character indicating the path where the the files generated by *Mplus* will be stored, including the name of the files. Default is NULL, which stores the files in the current working directory with the name *wrapBSEM*. |
| rm.files | A logical indicating whether the stored files should be deleted. The default is FALSE. |
| max.iter | A double indicating the maximum number of iterations. Default is 60000. |
| n.cores | A double indicating the number of CPU processors to be used for the estimation of the model. Default is 1. |
| n.chains | A double indicating the number of MCMC chains. Default is 2. |
| suppressMessages | A logical indicating whether the messages from MplusAutomation should be silent. Default is TRUE. |

## Value

BSEM returns an object of class wrapFA.

model.fit Model fit indices (vector).

lambda The standardized estimated factor loading matrix (matrix).

`phi` The standardized estimated factor correlation matrix (`matrix`).

`psi` The standardized estimated error correlation matrix (`matrix`).

`lambda.p` The p-value associated to each free parameter in the estimated factor loading matrix (`matrix`).

`phi.p` The p-value associated to each free parameter in the estimated factor correlation matrix (`matrix`).

`psi.p` The p-value associated to each free parameter in the estimated error correlation matrix (`matrix`).

`out` The complete output provided by *Mplus* (`list`).

`specifications` Specifications used to run the function (`list`).

## References

Muthén, B. & Asparouhov, T. (2012). Bayesian structural equation modeling: A more flexible representation of substantive theory. *Psychological Methods*, *17*, 313-335. https://doi.org/10.1037/a0026802

## Examples

```
BFI <- psych::bfi
BFI <- BFI[,1:25]
modBFI <- data.frame(agree = c(rep(1, 5), rep(0, 20)),
                     consc = c(rep(0, 5), rep(1, 5), rep(0, 15)),
                     extra = c(rep(0, 10), rep(1, 5), rep(0, 10)),
                     neuro = c(rep(0, 15), rep(1, 5), rep(0, 5)),
                     openn = c(rep(0, 20), rep(1, 5)), row.names = colnames(BFI))
modBFI.mpl <- modFA(lambda = modBFI, software = "mplus", keep.names = TRUE)
bsem <- BSEM(model = modBFI.mpl$BSEM, data = BFI, categorical = 1:25, n.cores = 2)
```

---

# CFA *Confirmatory factor analysis (CFA)*

---

## Description

Fit a CFA model via *lavaan* or *MplusAutomation*. Sequential model modifications can be implemented by using modification indices (MI) or the procedure proposed by Saris et al. (2009).

## Usage

```
CFA(
  model,
  data,
  categorical = NULL,
  estimator = NULL,
  software = NULL,
  mimic.mplus = TRUE,
  mplus.path = NULL,
  rm.files = FALSE,
  max.iter = 10000,
  mod = "none",
  mod.args = list(stepwise = TRUE, mod.error = TRUE, MI.cut = 3.841459, MI.power =
    0.75, SEPC.lambda = 0.4, SEPC.psi = 0.1, max.mods = 50),
  suppressMessages = TRUE
)
```

## Arguments

| | |
|---|---|
| model | Either a character string containing the user-specified CFA model using either the *lavaan* or the *Mplus* syntax, or a list containing the design Lambda matrix (i.e., factor loadings), Phi matrix (i.e., factor correlations), and Psi matrix (i.e., error correlations), in that specific order. In the design matrices, free and fixed parameters must be indicated with 1 and 0, respectively. |
| data | A *N* individuals x *J* observed variables matrix or data.frame. Missing values need to be coded as NA. |
| categorical | A vector containing the number or name of the observed variables that need to be treated as categorical. Default is NULL, which treates all observed variables as continuous. |
| estimator | A character indicating the estimator to be used to fit the model. It can be "ML", "MLM", "MLMV", "MLMVS", "MLF", "MLR", "GLS", "DWLS", "WLS", "WLSM", "WLSMV", "ULS","ULSM", and "ULSMV". Check ?lavaan::lavOptions or the *Mplus User's Guide* for more information. Default is NULL, which uses "ML" if all observed variables are continuous and "WLSMV" if any observed variable is categorical. |
| software | A character indicating the software to use for fitting the CFA model. It can be "lavaan" or "mplus". Only required if design matrices are used in the model argument; the software is automatically detected if a syntax is used. Default is NULL. |
| mimic.mplus | A logical indicating whether *lavaan* should mimic *Mplus* output. Only applicable to *lavaan*. Default is TRUE. |
| mplus.path | A character indicating the path where the the files generated by *Mplus* will be stored, including the name of the files. Only applicable to *Mplus*. Default is NULL, which stores the files in the current working directory with the name *wrapCFA*. |
| rm.files | A logical indicating whether the stored files should be deleted. The default is FALSE. |
| max.iter | A double indicating the maximum number of iterations. Default is 10000. |
| mod | A character indicating whether sequential model modifications are permitted. It can be "none" for no modifications, "MI" for modifications based solely on modification indices, or "Saris" for the Saris procedure (Saris et al., 2009). "Saris" is only applicable to *lavaan*. Default is "none". |
| mod.args | A list of arguments for specifying the sequential model modification search: |
| | stepwise A logical indicating whether model modifications should be included sequentially (rather than in a single step). Default is TRUE. |
| | mod.error A logical indicating whether error correlations should be permitted. Default is TRUE. |
| | MI.cut A numeric indicating the MI cutoff point. Default is 3.841459. |
| | MI.power A numeric indicating the cutoff for the power associated to the MI in the Saris procedure. Default is 0.75. |
| | SEPC.lambda A numeric indicating the cutoff for the standardized expected parameter change (SEPC) associated to factor loadings in the Saris procedure. Default is 0.4. |
| | SEPC.psi A numeric indicating the cutoff for the SEPC associated to error correlations in the Saris procedure. Default is 0.1. |

max.mods A double indicating the maximum number of modifications permitted. Default is 50.

suppressMessages
A logical indicating whether the messages from MplusAutomation should be silent. Default is TRUE.

## Value

CFA returns an object of class wrapFA.

model.fit Model fit indices (vector).

lambda The standardized estimated factor loading matrix (matrix).

phi The standardized estimated factor correlation matrix (matrix).

psi The standardized estimated error correlation matrix (matrix).

lambda.p The p-value associated to each free parameter in the estimated factor loading matrix (matrix).

phi.p The p-value associated to each free parameter in the estimated factor correlation matrix (matrix).

psi.p The p-value associated to each free parameter in the estimated error correlation matrix (matrix).

MI Information about the MI and SEPC of the fixed parameters (matrix).

mods The modifications included in the model ((matrix)).} \item{outThe complete output provided by either *lavaan* or *Mplus* (list).

specifications Specifications used to run the function (list).

## References

Saris, W. E., Satorra, A., & van der Veld, W. M. (2009). Testing structural equation models or detection of misspecifications?. *Structural Equation Modeling*, *16*, 561-582. https://doi.org/10.1080/10705510903203433

## Examples

```
library(MBESS)
data(HS)
HS <- HS[, -c(1:8)]
colnames(HS) <- paste0("t", 1:26)
modHS <- data.frame(spatial = c(rep(1, 4), rep(0, 20), rep(1, 2)),
                    verbal = c(rep(0, 4), rep(1, 5), rep(0, 17)),
                    speed = c(rep(0, 9), rep(1, 4), rep(0, 13)),
                    memory = c(rep(0, 13), rep(1, 6), rep(0, 7)),
                   maths = c(rep(0, 19), rep(1, 5), rep(0, 2)), row.names = colnames(HS))
modHS.lav <- modFA(lambda = modHS, software = "lavaan", keep.names = TRUE)
cfa.lav <- CFA(model = modHS.lav$CFA, data = HS)
```

detFA *Determinacy of factor score estimates*

## Description

Computes the determinacy of factor score estimates. Apart from the estimated determinacy (using the implied matrices), the true determinacy can be computed if the generating matrices are provided (e.g., in simulation studies). Both the observed correlation matrix and the implied correlation matrix (Beauducel, 2011) are used to calculate the determinacy.

## Usage

```
detFA(
  data = NULL,
  r = NULL,
  est.lambda,
  est.phi,
  est.psi,
  true.lambda = NULL,
  true.phi = NULL,
  align = "auto"
)
```

## Arguments

| | |
|---|---|
| data | A *N* individuals x *J* observed variables matrix or data.frame. Missing values need to be coded as NA. It can be NULL if the correlation matrix (r) is provided. Default is NULL. |
| r | A *J* observed variables x *J* observed variables correlation matrix or data.frame. It can be NULL if the data are provided. Default is NULL. |
| est.lambda | A *J* observed variables x *K* latent variables estimated factor loading matrix or data.frame. |
| est.phi | A *K* latent variables x *K* latent variables estimated factor correlation matrix or data.frame. |
| est.psi | A *J* observed variables x *J* observed variables estimated residual correlation matrix or data.frame. |
| true.lambda | A *J* observed variables x *K* latent variables generating factor loading matrix or data.frame. Required to compute the true determinacy of factor score estimates. Useful for simulation studies. Default is NULL. |
| true.phi | A *K* latent variables x *K* latent variables generating factor correlation matrix or data.frame. Required to compute the true determinacy of factor score estimates. Useful for simulation studies. Default is NULL. |
| align | Align factor scores? It can be either 'auto' or a numeric vector. If 'auto', factor scores will be aligned based on true.lambda (not aligned if true.lambda = NULL). Otherwise, the numeric vector must identify the order for the factor scores. Default is 'auto'. |

## Value

detFA returns an object of class detFA.

est.r The estimated determinacy based on the observed correlation matrix (matrix).

est.sigma The estimated determinacy based on the implied correlation matrix (matrix).

true.r The true determinacy based on the observed correlation matrix. Only if true.lambda and true.phi have been provided (matrix).

true.r The true determinacy based on the implied correlation matrix. Only if true.lambda and true.phi have been provided (matrix).

## References

Beauducel, A. (2011). Indeterminacy of factor score estimates in slightly misspecified confirmatory factor models. *Journal of Modern Applied Statistical Methods*, *10*, 583-598. https://doi.org/10.22237/jmasm/1320120900

## Examples

```
library(MBESS)
data(HS)
HS <- HS[, -c(1:8)]
colnames(HS) <- paste0("t", 1:26)
modHS <- data.frame(spatial = c(rep(1, 4), rep(0, 20), rep(1, 2)),
                    verbal = c(rep(0, 4), rep(1, 5), rep(0, 17)),
                    speed = c(rep(0, 9), rep(1, 4), rep(0, 13)),
                    memory = c(rep(0, 13), rep(1, 6), rep(0, 7)),
                   maths = c(rep(0, 19), rep(1, 5), rep(0, 2)), row.names = colnames(HS))
modHS.lav <- modFA(lambda = modHS, software = "lavaan", keep.names = TRUE)
efa_geomin <- EFA(model = modHS.lav$EFA, data = HS)
fd_efa <- detFA(HS, est.lambda = efa_geomin$lambda, est.phi = efa_geomin$phi, est.psi = efa_geomin$psi)
```

---

# ECFA *EFA-based confirmatory factor analysis (ECFA)*

---

## Description

Fit an ECFA model (Nájera et al., 2022) via *lavaan* or *MplusAutomation*.

## Usage

```
ECFA(
  fit = NULL,
  model = NULL,
  data,
  method = "r2",
  r2.args = list(phi.grid = seq(0.7, 0.99, 0.01), selector = "BIC"),
  categorical = NULL,
  estimator = NULL,
  software = NULL,
  mimic.mplus = TRUE,
  mplus.path = NULL,
  rm.files = FALSE,
```

```
  max.iter = 10000,
  suppressMessages = TRUE
)
```

## Arguments

| | |
|---|---|
| fit | A wrapFA object from EFA() or BSEM() functions. Default is NULL, which means that a user-specified model will be used. |
| model | A character string containing the EFA model using either the *Mplus* or the *lavaan* syntax. Default is NULL, which means that the argument fit will be used. |
| data | A *N* individuals x *J* observed variables matrix or data.frame. Missing values need to be coded as NA. |
| method | A character indicating the method to detect the relevant loadings. It can be "r2" for the R-squared procedure (Citation, Year), "nominal" for statistically significant loadings with *alpha* = 0.05, or "bonferroni" for statistically significant loadings using the Bonferroni correction. Default is "r2". |
| r2.args | A list of arguments about the R-squared procedure:<br>phi.grid A numeric vector indicating the phi parameter space to search. Default is from 0.70 to 0.99, in steps of 0.01.<br>selector A character indicating fit index to use to select the final model. It can be "ChiSq/df", "CFI", "TLI", "AIC", "BIC", "aBIC", "RMSEA", "SRMR", and "AICC". Default is "BIC". |
| categorical | A vector containing the number or name of the observed variables that need to be treated as categorical. Default is NULL, which treates all observed variables as continuous. |
| estimator | A character indicating the estimator to be used to fit the model. It can be "ML", "MLM", "MLMV", "MLMVS", "MLF", "MLR", "GLS", "DWLS", "WLS", "WLSM", "WLSMV", "ULS","ULSM", and "ULSMV". Check ?lavaan::lavOptions or the Mplus User's Guide for more information. Default is NULL, which uses "ML" if all observed variables are continuous and "WLSMV" if any observed variable is categorical. |
| software | A character indicating the software to use for fitting the CFA model. It can be "lavaan" or "mplus". Only required if design matrices are used in the model argument; the software is automatically detected if a syntax is used. Default is NULL. |
| mimic.mplus | A logical indicating whether *lavaan* should mimic *Mplus* output. Only applicable to *lavaan*. Default is TRUE. |
| mplus.path | A character indicating the path where the the files generated by Mplus will be stored, including the name of the files. Default is NULL, which stores the files in the current working directory with the name *wrapECFA*. |
| rm.files | A logical indicating whether the stored files should be deleted. The default is FALSE. |
| max.iter | A double indicating the maximum number of iterations. Default is 10000. |
| suppressMessages | A logical indicating whether the messages from MplusAutomation should be silent. Default is TRUE. |

## Value

ECFA returns an object of class wrapFA.

model.fit Model fit indices (vector).

lambda The standardized estimated factor loading matrix (matrix).

phi The standardized estimated factor correlation matrix (matrix).

psi The standardized estimated error correlation matrix (matrix).

lambda.p The p-value associated to each free parameter in the estimated factor loading matrix (matrix).

phi.p The p-value associated to each free parameter in the estimated factor correlation matrix (matrix).

psi.p The p-value associated to each free parameter in the estimated error correlation matrix (matrix).

MI Information about the MI and SEPC of the fixed parameters (matrix).

out The complete output provided by either *lavaan* or *Mplus* (list).

specifications Specifications used to run the function (list).

## References

Nájera, P., Abad, F. J., & Sorrel, M. A. (2022). Is EFA always to be preferred? A systematic comparison of factor analytic techniques throughout the confirmatory-exploratory continuum. *Manuscript submitted for publication*.

## Examples

```
library(MBESS)
data(HS)
HS <- HS[, -c(1:8)]
colnames(HS) <- paste0("t", 1:26)
modHS <- data.frame(spatial = c(rep(1, 4), rep(0, 20), rep(1, 2)),
                    verbal = c(rep(0, 4), rep(1, 5), rep(0, 17)),
                    speed = c(rep(0, 9), rep(1, 4), rep(0, 13)),
                    memory = c(rep(0, 13), rep(1, 6), rep(0, 7)),
                   maths = c(rep(0, 19), rep(1, 5), rep(0, 2)), row.names = colnames(HS))
modHS.lav <- modFA(lambda = modHS, software = "lavaan", keep.names = TRUE)
efa_geomin <- EFA(model = modHS.lav$EFA, data = HS)
ecfa_r2 <- ECFA(fit = efa_geomin, data = HS, method = "r2", r2.args = list(phi.grid = seq(0.85, 0.95, 0.01), sele
```

---

EFA *Exploratory factor analysis (EFA)*

---

## Description

Fit an EFA model via *lavaan* or *MplusAutomation*. Correlated residuals can be included in the model as in exploratory structural equation modeling (ESEM; Asparouhov & Muthén, 2009). Several rotation procedures are supported. The objectively refined target matrix (RETAM) procedure (Lorenzo-Seva & Ferrando, 2020) is also available if *MplusAutomation* is used. Moreover, sequential model modifications can be implemented by using modification indices (MI) or the procedure proposed by Saris et al. (2009).

## Usage

```
EFA(
  model = NULL,
  data,
  n.factors = NULL,
  categorical = NULL,
  estimator = NULL,
  rotation = "Geomin",
  software = NULL,
  mimic.mplus = TRUE,
  mplus.path = NULL,
  rm.files = FALSE,
  max.iter = 10000,
  mod = "none",
  mod.args = list(stepwise = TRUE, mod.error = TRUE, MI.cut = 3.841459, MI.power =
    0.75, SEPC.lambda = 0.4, SEPC.psi = 0.1, max.mods = 50),
  suppressMessages = TRUE
)
```

## Arguments

| | |
|---|---|
| model | A character string containing the EFA model using either the *lavaan* or *Mplus* syntax. It can be null if the number of factors is specified instead in the argument n.factors. |
| data | A *N* individuals x *J* observed variables matrix or data.frame. Missing values need to be coded as NA. |
| n.factors | A double indicating the number of factors to extract. It can be null if model is specified instead. |
| categorical | A vector containing the number or name of the observed variables that need to be treated as categorical. Default is NULL, which treates all observed variables as continuous. |
| estimator | A character indicating the estimator to be used to fit the model. It can be "ML", "MLM", "MLMV", "MLMVS", "MLF", "MLR", "GLS", "DWLS", "WLS", "WLSM", "WLSMV", "ULS","ULSM", and "ULSMV". Check ?lavaan::lavOptions or the *Mplus* User's Guide for more information. Default is NULL, which uses "ML" if all observed variables are continuous and "WLSMV" if any observed variable is categorical. |
| rotation | A character indicating the rotation procedure to be used. It can be "Geomin", "Target", "RETAM", "Oblimin", "Promax", "Varimax", and "Quartimin". "Target" and "RETAM" are only available for *Mplus*. |
| software | A character indicating the software to use for fitting the EFA model. It can be "lavaan" or "mplus". Only required if design matrices are used in the model argument; the software is automatically detected if a syntax is used. Default is NULL. |
| mimic.mplus | A logical indicating whether *lavaan* should mimic *Mplus* output. Only applicable to *lavaan*. Default is TRUE. |
| mplus.path | A character indicating the path where the the files generated by *Mplus* will be stored, including the name of the files. Default is NULL, which stores the files in the current working directory with the name *wrapEFA*. |

rm.files A logical indicating whether the stored files should be deleted. The default is FALSE.

max.iter A double indicating the maximum number of iterations. Default is 10000.

mod A character indicating whether sequential model modifications are permitted. It can be "none" for no modifications, "MI" for modifications based solely on modification indices, or "Saris" for the Saris procedure (Saris et al., 2009). "Saris" is only applicable to *lavaan*. Default is "none".

mod.args A list of arguments for specifying the sequential model modification search:

- stepwise A logical indicating whether model modifications should be included sequentially (rather than in a single step). Default is TRUE.
- mod.error A logical indicating whether error correlations should be permitted. Default is TRUE.
- MI.cut A numeric indicating the MI cutoff point. Default is 3.841459.
- MI.power A numeric indicating the cutoff for the power associated to the MI in the Saris procedure. Default is 0.75.
- SEPC.lambda A numeric indicating the cutoff for the standardized expected parameter change (SEPC) associated to factor loadings in the Saris procedure. Default is 0.4.
- SEPC.psi A numeric indicating the cutoff for the SEPC associated to error correlations in the Saris procedure. Default is 0.1.
- max.mods A double indicating the maximum number of modifications permitted. Default is 50.

suppressMessages A logical indicating whether the messages from MplusAutomation should be silent. Default is TRUE.

## Value

EFA returns an object of class wrapFA.

model.fit Model fit indices (vector).

lambda The standardized estimated factor loading matrix (matrix).

phi The standardized estimated factor correlation matrix (matrix).

psi The standardized estimated error correlation matrix (matrix).

lambda.p The p-value associated to each free parameter in the estimated factor loading matrix (matrix).

phi.p The p-value associated to each free parameter in the estimated factor correlation matrix (matrix).

psi.p The p-value associated to each free parameter in the estimated error correlation matrix (matrix).

MI Information about the MI and SEPC of the fixed parameters (matrix).

mods The modifications included in the model ((matrix)).} \item{outThe complete output provided by either *lavaan* or *Mplus* (list).

specifications Specifications used to run the function (list).

## References

Asparouhov, T., & Muthén, B. O. (2009). Exploratory structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, *16*(3), 397-438. https://doi.org/10.1080/10705510903008204 Lorenzo-Seva, U., & Ferrando, P. J. (2020). Unrestricted factor analysis of multidimensional test items based on an objectively refined target matrix. *Behavior Research Methods*, *52*, 116-130. https://doi.org/10.3758/s13428-019-01209-1 Saris, W. E., Satorra, A., & van der Veld, W. M. (2009). Testing structural equation models or detection of misspecifications?. *Structural Equation Modeling*, *16*, 561-582. https://doi.org/10.1080/10705510903203433

## Examples

```
library(MBESS)
data(HS)
HS <- HS[, -c(1:8)]
colnames(HS) <- paste0("t", 1:26)
modHS <- data.frame(spatial = c(rep(1, 4), rep(0, 20), rep(1, 2)),
                    verbal = c(rep(0, 4), rep(1, 5), rep(0, 17)),
                    speed = c(rep(0, 9), rep(1, 4), rep(0, 13)),
                    memory = c(rep(0, 13), rep(1, 6), rep(0, 7)),
                   maths = c(rep(0, 19), rep(1, 5), rep(0, 2)), row.names = colnames(HS))
modHS.lav <- modFA(lambda = modHS, software = "lavaan", keep.names = TRUE)
modHS.mpl <- modFA(lambda = modHS, software = "mplus", keep.names = TRUE)
efa_geomin <- EFA(model = modHS.lav$EFA, data = HS)
efa_target <- EFA(model = modHS.mpl$EFAt, data = HS, rotation = "TARGET")
```

---

# modFA *Translate design matrices into syntax for lavaan or Mplus models*

---

## Description

Translates design matrices (factor loadings, factor correlations, residual correlations) into a syntax that is compatible with either *lavaan* and *Mplus*. Namely, CFA and EFA are supported for *lavaan*, and CFA, EFA, EFA with target rotation, and BSEM are supported for *Mplus*.

## Usage

```
modFA(lambda, phi = NULL, psi = NULL, software, keep.names = TRUE)
```

## Arguments

| | |
|---|---|
| `lambda` | Factor loading `matrix` (or `data.frame`). |
| `phi` | Factor correlation `matrix` (or `data.frame`). Default is `NULL`, which implies that all factor correlations are included in the model. |
| `psi` | Residual correlation `matrix` (or `data.frame`). Default is `NULL`, which implies that no residual correlations are included in the model. |
| `software` | A `character` indicating the software to use for translating the design matrices into syntax. It can be `"lavaan"`, `"mplus"`, or `"lslx"`. |
| `keep.names` | A `logical` indicating whether the names of the observed and latent variables (taken from `lambda`) should be retained or not. Default is `TRUE`. If `FALSE`, `x1,x2,...` is used for observed variables and `F1,F2,...` for latent variables. |

## Value

modFA returns an object of class modFA.

CFA The syntax for fitting a CFA (character).

EFA The syntax for fitting an EFA with mechanical/analytical rotation (character).

EFAt The syntax for fitting an EFA with target rotation. Only available for *Mplus* (character).

BSEM The syntax for fitting a BSEM. Only available for *Mplus* (character).

specifications Specifications used to implement the modFA function (list).

## Examples

```
bfi <- psych::bfi
bfi <- bfi[,1:25]
modelbfi <- data.frame(agree = c(rep(1, 5), rep(0, 20)),
                       consc = c(rep(0, 5), rep(1, 5), rep(0, 15)),
                       extra = c(rep(0, 10), rep(1, 5), rep(0, 10)),
                       neuro = c(rep(0, 15), rep(1, 5), rep(0, 5)),
                       openn = c(rep(0, 20), rep(1, 5)), row.names = colnames(bfi))
modelbfi.lav <- modFA(lambda = modelbfi, software = "lavaan", keep.names = TRUE)
modelbfi.mpl <- modFA(lambda = modelbfi, software = "mplus", keep.names = TRUE)
```

# Index