Package 'EFA.dimensions'

September 30, 2026

Title Exploratory Factor Analysis Functions for Assessing Dimensionality

Type Package

Version 0.1.9.2

Date 2026-09-29

Author Brian P. O'Connor [aut, cre]

Maintainer Brian P. O'Connor <brian.oconnor@ubc.ca>

Description Functions for an assortment of factor analysis-related procedures, including eleven procedures for determining the number of factors; for factor analysis with multiple options for methods of extraction and rotation; for bi-factor analysis; for extension factor analysis; options for running the analyses using either raw data or correlation matrices as input and with options for conducting the analyses using Pearson correlations, Kendall correlations, Spearman correlations, gamma correlations, or polychoric correlations; wrapper 'lavaan'-based functions for factorial invariance and exploratory structural equation modeling; functions for the factor-ability of a correlation matrix, for the congruence between factors from different datasets, for the assessment of local independence, for the assessment of factor solution complexity, for internal consistency, and for correcting Pearson correlation coefficients for attenuation due to unreliability. Auerswald & Moshagen (2019, <doi:10.1037/met0000200>); Field, Miles, & Field (2012, ISBN:978-1-4462-0045-2); Mulaik (2010, ISBN:978-1-4200-9981-2); O'Connor (2000, <doi:10.3758/bf03200807>).

Imports stats, psych, polycor, EFAtools, utils, mirt, GPArotation, lavaan, semTools

Suggests lattice

LazyLoad yes

LazyData yes

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2026-09-30 15:10:17 UTC

Contents

EFA.dimensions-package . . . 3
BIFACTOR . . . 4
COMPLEXITY . . . 10
CONGRUENCE . . . 11
CORRECTED_CORRELS . . . 13
data_Field . . . 15
data_Harman . . . 16
data_HS_1939 . . . 17
data_NEOPIR . . . 17
data_RSE . . . 18
data_RSE_not_recoded . . . 19
data_RSE_sex . . . 19
data_SDT . . . 20
data_TabFid . . . 20
DIMTESTS . . . 21
EFA . . . 23
EFA.dimensions-defunct . . . 27
EFA_SCORES . . . 28
EMPKC . . . 31
ESEM . . . 33
EXTENSION_FA . . . 37
FACTORABILITY . . . 40
Factorial_Invariance . . . 42
INTERNAL_CONSISTENCY . . . 46
LOCALDEP . . . 48
MAP . . . 50
MISSING_INFO . . . 51
NEVALSGT1 . . . 52
OMEGA . . . 54
PARALLEL . . . 59
PCA . . . 61
PLOT_Invariance . . . 64
POLYCHORIC_R . . . 65
PROCRUSTES . . . 67
RAWPAR . . . 68
RECODE . . . 71
ROOTFIT . . . 72
SALIENT . . . 76
SCREE_PLOT . . . 79
SESCREE . . . 80
SMT . . . 81

Index **83**

EFA.dimensions-package *EFA.dimensions*

Description

This package provides exploratory factor analysis-related functions for an assortment of factor analysis-related procedures.

There are 11 functions for determining the number of factors (DIMTESTS, EMPKC, HULL, MAP, NEVALSGT1, PARALLEL, RAWPAR, ROOTFIT, SALIENT, SCREE_PLOT, SESCREE, and SMT).

There is a principal components analysis function (PCA), an exploratory factor analysis function (EFA), a bifactor analysis function (BIFACTOR), and an extension factor analysis function (EXTENSION_FA), all with 9 possible factor extraction methods and 12 possible factor rotation methods.

Most of the analyses can be conducted using raw data or correlation matrices as input.

The analyses can be conducted using Pearson correlations, Kendall correlations, Spearman correlations, Goodman-Kruskal gamma correlations (Thompson, 2006), or polychoric correlations (using the psych and polychor packages).

Additional functions include:

- COMPLEXITY, for the assessment of factor solution complexity
- CONGRUENCE, for the congruences between factors from different datasets
- CORRECTED_CORRELS, correcting Pearson correlation coefficients for attenuation due to unreliability
- ESEM, for exploratory structural equation measurement models
- FACTORABILITY, for the factorability of a correlation matrix
- Factorial_Invariance, for multiple groups factorial invariance
- PLOT_Invariance, for plots of multiple groups factorial invariance statistics
- LOCALDEP, for the assessment of local independence
- INTERNAL_CONSISTENCY, for internal consistency statistics

References

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. *Psychological Methods, 24(4),* 468-491.

Mulaik, S. A. (2010). *Foundations of factor analysis* (2nd ed.). Boca Raton, FL: Chapman and Hall/CRC Press, Taylor & Francis Group.

O'Connor, B. P. (2000). SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test. *Behavior Research Methods, Instrumentation, and Computers, 32,* 396-402.

Sellbom, M., & Tellegen, A. (2019). Factor analysis in psychological assessment research: Common pitfalls and recommendations. *Psychological Assessment, 31(12),* 1428-1441.

Watts, A. L., Greene, A. L., Ringwald, W., Forbes, M. K., Brandes, C. M., Levin-Aspenson, H. F., & Delawalla, C. (2023). Factor analysis in personality disorders research: Modern issues and illustrations of practical recommendations. *Personality Disorders: Theory, Research, and Treatment, 14(1),* 105-117. https://doi.org/10.1037/per0000581

`BIFACTOR` *Bifactor analysis*

Description

Bifactor analysis with multiple options for data entry and analysis

Usage

```
BIFACTOR(loadings = NULL,
         rawdata = NULL,
         cormat = NULL, Ncases = NULL, corkind = 'pearson',
         Nfactors = 4,
         bifactor_kind = 'bifactorT',
         EFA_options    = list(extraction = 'minres',
                               rotation = 'oblimin'),
         LV_options     = list(group_keys = NULL,
                               estimator='MLR',
                               rotation = 'bigeomin',
                               resid_correls=NULL,
                               LV_names=NULL,
                               ordered = FALSE),
         schmid_options = list(extraction = 'minres',
                               rotation = 'oblimin'),
         GPA_options    = list(delta = .01,
                               epsilon = .00001,
                               normalize = FALSE,
                               maxit = 1000,
                               randomStarts = 50),
         min_loading = .2,
         verbose = TRUE)
```

Arguments

`loadings`	(optional) loadings can be either a bifactor loading matrix or an EFA loading matrix. See the Details and Examples sections below.

`rawdata` (optional) An all-numeric dataframe where the rows are cases & the columns are the variables. Required only when loadings and cormat are not provided.

`Nfactors` (optional) The number of factors to extract, including the general factor. Nfactors is = 1 (for the general factor) + the number of group factors.

`cormat` (optional) A correlation matrix with ones on the diagonal. Required only when loadings and rawdata are not provided.

`Ncases` (optional) The number of cases. Required only when cormat is provided.

`corkind` (optional) The kind of correlation matrix to be used when rawdata is provided. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'.

`bifactor_kind` (optional) Specify one of the following possible bifactor methods: 'bifactorT', 'bifactorQ', 'bigeominT', 'bigeominQ', 'SL', 'SLiD', 'DSL', 'CFA', 'ESEM', or 'none'. Use 'bifactor_kind = "none"' only when loadings is (already) a bifactor loading matrix.
Example: bifactor_kind = 'bifactorT'

`EFA_options` (optional) A list with EFA options when rawdata is provided. The list elements include values for 'extraction' and 'rotation'.

The possibilities for extraction are: 'minres'(the default), 'alpha', 'fullinfo', 'gls', 'image', 'ml', 'ols', 'paf', 'uls', and 'wls'.

The possibilities for rotation are: 'varimax', 'bentlerT', 'entropy', 'equamax', 'geominT', 'quartimax', 'promax', 'bentlerQ', 'geominQ', 'oblimin' (the default, 'oblimax', 'quartimin', 'simplimax', and 'none'.

`LV_options` (optional) A list with cfa options when bifactor_kind = 'CFA'. The possible list elements are 'group_keys', 'N_group_factors', 'estimator', 'rotation', 'resid_correls', 'LV_names', and 'ordered'. See the Details section below for information on each element.

`schmid_options` (optional) A list with schmid function (from the psych package) options when bifactor_kind is one of 'SL', 'SLiD', or 'DSL'. The list elements include values for 'extraction' and 'rotation'.

The possibilities for extraction are: 'minres'(the default), 'paf', 'pc', and 'ml'.

The possibilities for rotation are: 'oblimin' (the default), 'simplimax', 'Promax', 'promax' and 'none'.

`GPA_options` (optional) A list with options for when a GPArotation is used. See the Details section below.

`min_loading` The minimum value of a group factor loading for an item to be considered to have a non-negligible contribution to a group factor. min_loading only plays a role in the computations for PUC and for some group factor statistics.

`verbose` Should detailed results be displayed in console? TRUE (default) or FALSE

Details

The bifactor analyses can be conducted for three possible data input scenarios.

- When "loadings" is provided and when it is (already) a bifactor loading matrix, then bifactor rotations are not conducted and the function merely provides a variety of bifactor model, factor, and item statistics. "bifactor_kind" must be set to "none" for the function to recognize that loadings is (already) a bifactor loading matrix.
- When "loadings" is provided and when it consists of EFA loadings (not bifactor loadings), then the function will conduct the requested kind of bifactor analysis and provide the bifactor loadings and statistics.
- if either "rawdata" or "cormat" are provided, the function will conduct the preliminary EFA, the bifactor analyses, and produce the bifactor loadings and statistics. "loadings" must be NULL for this to occur.

The factor extraction computations for the following methods are conducted using the psych package (Revelle, 2026): 'alpha', 'gls', 'minres', 'ols', 'uls', and 'wls'.

The factor extraction computations for 'fullinfo' are conducted using the mirt package (Chalmers, 2012). Full-information methods are considered more appropriate for item-level data than other factor extraction methods (Wirth & Edwards, 2007).

The factor rotation computations for the following methods are conducted using the GPArotation package (Bernaards & Jennrich, 2005, 2026): 'bentlerQ', 'bentlerT', 'entropy', 'geominQ', 'geominT', 'oblimax', 'oblimin', 'quartimax', 'quartimin', and 'simplimax'.

The 'bifactorT', 'bifactorQ', 'bigeominQ', and 'bigeominQ' computations are conducted using the GPArotation package (Bernaards & Jennrich, 2005, 2026). The 'SL' and 'DSL' bifactor computations are conducted using the psych package (Revelle, 2026). The 'SLiD' bifactor computations are conducted using code from Garcia-Garzon et al. (2021).

The only possible extraction methods for 'SL', 'SLiD', and 'DSL' bifactor computations are 'paf', 'minres', and 'ml'.

For reviews of bifactor analyses, see Bornovalova (2020), Garcia-Garzon et al. (2021), & Reise et al. (2018).

- **bifactorT** is an orthogonal bifactor rotation designed for situations where a single, overarching global dimension is expected alongside separate sub-domains, and when all factors should be uncorrelated.
- **bifactorQ** is an oblique bifactor rotation designed for when a strong, overarching global dimension is expected alongside separate sub-domains that are allowed to correlate with each other.
- **bigeominT** is an orthogonal bifactor rotation designed for situations where a single, overarching global dimension is expected alongside separate sub-domains, and when all factors should be uncorrelated.
- **bigeominQ** is an oblique bifactor rotation designed for when a strong, overarching global dimension is expected alongside separate sub-domains that are allowed to correlate with each other.
- **SL** is an oblique bifactor rotation designed for where there is a broad, overarching factor alongside sub-domains that are allowed to overlap.
- **SLiD** an orthogonal bifactor rotation method designed for exploratory bifactor analysis that tries to ensure that once the general factor's variance is pulled out, each item loads onto only one specific group factor.

- **DSL** entropy is an orthogonal factor rotation that pushes factor loadings to be either strongly dominant (close to 1.0) or cleanly absent (close to 0.0), reducing the overall informational "noise" of the matrix.
- **CFA** confirmatory bifactor analyses are conducted using the cfa function from the lavaan package. All items are assumed to load on the general factor. The items that load on the group factors are indicated using "group_keys" for the LV_options argument.
- **ESEM** rxploratory structural equation modeling bifactor analyses conducted using the cfa function from the lavaan package. All items are assumed to load on the general factor. The items that load on the group factors can be indicated using "group_keys" for the LV_options argument.

For the **LV_options** argument:

When bifactor_kind = 'CFA' or 'ESEM', then LV_options is a list with information for the theseanalyses. The possible list elements are 'group_keys', 'estimator', 'rotation', 'resid_correls', 'LV_names', and 'ordered'.

- **group_keys** is a vector indicating which item (i.e., which variable in data) goes with which group factor (which can be named or numbered).
  Example: group_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2)
- **estimator** is the name of the lavaan estimator to be used in the analyses. The options for basic estimators (continuous data) are ML, GLS, WLS, DWLS, ULS, DLS, and PML. The options for robust estimators are MLM, MLMVS, MLMV, MLF, MLR, WLSM, WLSMVS, WLSMV, ULSM, ULSMVS, and ULSMV.
  Example: estimator = 'ML'
- **rotation** is the kind of lavaan bifactor rotation. The two options are 'bigeomin' or 'biquartimin'.
- **resid_correls** is a vector with the pairs of correlated error terms (if any).
  Example: resid_correls = c('Q1 ~~ Q2', 'Q9 ~~ Q10')
- **LV_names** is a vector of names for the latent variables
- **ordered** are the variables in rawdata (if provided) ordered? For example, Likert scale responses are usually ordered. The function will produce notices when the data are not consistent with the "ordered" status.

Examples of LV_options:

```
LV_options = list(group_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2),
                  resid_correls = NULL, estimator = 'WLSMVS')


LV_options = list(group_keys = c(Q1=1, Q2=1, Q3_R=2, Q4=1,   Q5_R=2,
                                 Q6=1, Q7=1, Q8_R=2, Q9_R=2, Q10_R=2)),
                  resid_correls = c('Q1 ~~ Q2', 'Q9 ~~ Q10'),
                  estimator = 'WLSMVS')
```

For the **GPA_options** argument:

A list with options for when a GPArotation is used. The list elements are 'delta', 'epsilon', 'normalize', 'maxit', and 'randomStarts'.

- **delta** is a small positive number that is added to the squared factor loadings in the (geomin) optimization denominator to prevent division by zero and control the sharpness of the penalty near zero. (Note: In MPLus, delta is called "epsilon".)
- **epsilon** is the numeric rotational convergence tolerance threshold. It controls how precise or strict the stopping rule is for the rotation algorithm. The gradient projection algorithm stops optimizing and assumes the rotation has converged when the norm of the gradient falls below this value.
- **normalize** specifies whether and how the loadings matrix is normalized prior to rotation and denormalized after rotation. The possibilities are FALSE (the default) for no normalization, or TRUE, for the Kaiser/Horst normalization.
- **maxit** is the maximum number of iterations.
- **randomStarts** is an integer that specifies the number of random starting configurations used to optimize the factor rotation. It helps avoid getting trapped in local minima during factor rotation optimization. It runs the gradient projection algorithm multiple timeseach using a different random orthogonal or oblique starting rotation matrix (Tmat) generated via QR decomposition. It returns the rotated factor loadings matrix and results corresponding to the run with the lowest objective criterion value among all attempts.

See this web page for more detailed descriptions of the above extraction, rotation, and bifactor methods:

https://oconnor-psych.ok.ubc.ca/vignettes/EFA_BIFACTOR_vignettes.html

Value

A list with the bifactor loadings and statistics, including omegas, ECV, ARPB, FD, coef_H, and rmsr.

Author(s)

Brian P. O'Connor

References

Bernaards, C. A., & Jennrich, R. I. (2005). Gradient Projection Algorithms and Software for Arbitrary Rotation Criteria in Factor Analysis. *Educational and Psychological Measurement, 65(5),* 676-696. https://doi.org/10.1177/0013164404272507

Bernaards, C. A., & Jennrich, R. I. (2026). GPArotation: Gradient Projection Factor Rotation. R package version 2026.4-1, https://CRAN.R-project.org/package=GPArotation

Bornovalova, M. A., Choate, A. M., Fatimah, H., Petersen, K. J., & Wiernik, B. M. (2020). Appropriate Use of Bifactor Analysis in Psychopathology Research: Appreciating Benefits and Limitations. *Biological psychiatry, 88(1),* 1827.

Chalmers, R. P. (2012). mirt: A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software, 48(6),* 129. doi:10.18637/jss.v048.i06.

Garcia-Garzon, E., Abad, F. J., & Garrido, L. E. (2021). On omega hierarchical estimation: A Comparison of Exploratory Bi-Factor Analysis Algorithms. *Multivariate Behavioral Research, 56(1),* 101-119.

Giordano, C., Ones, D. S., Waller, N. G., & Stanek, K. C. (2020). Exploratory bifactor measurement models in vocational behavior research. *Journal of Vocational Behavior, 120,* Article 103430.

Jennrich, R. I. (2018). Rotation. In P. Irwing, T. Booth, & D. J. Hughes (Eds.), *The Wiley handbook of psychometric testing: A multidisciplinary reference on survey, scale and test development* (pp. 279304). Wiley Blackwell. https://doi.org/10.1002/9781118489772.ch10

Markon K. E. (2019). Bifactor and Hierarchical Models: Specification, Inference, and Interpretation. *Annual review of clinical psychology, 15,* 5169.

Mulaik, S. A. (2010). *Foundations of factor analysis* (2nd ed.). Boca Raton, FL: Chapman and Hall/CRC Press, Taylor & Francis Group.

Reise, S.P., Bonifay, W., & Haviland, M.G. (2018). Bifactor Modelling and the Evaluation of Scale Scores. In P. Irwing, T. Booth& D.J. Hughes (eds.), *The Wiley Handbook of Psychometric Testing*. John Wiley & Sons.

Revelle, W. (2026). psych: Procedures for Psychological, Psychometric, and Personality Research. R package version 2.6.5, https://CRAN.R-project.org/package=psych

Sellbom, M., & Tellegen, A. (2019). Factor analysis in psychological assessment research: Common pitfalls and recommendations. *Psychological Assessment, 31(12),* 1428-1441. https://doi.org/10.1037/pas0000623

Watts, A. L., Greene, A. L., Ringwald, W., Forbes, M. K., Brandes, C. M., Levin-Aspenson, H. F., & Delawalla, C. (2023). Factor analysis in personality disorders research: Modern issues and illustrations of practical recommendations. *Personality Disorders: Theory, Research, and Treatment, 14(1),* 105-117. https://doi.org/10.1037/per0000581

Wirth, R. J., & Edwards, M. C. (2007). Item factor analysis: current approaches and future directions. *Psychological methods, 12(1),* 58-79. https://doi.org/10.1037/1082-989X.12.1.58

Examples

```
# using only EFA loadings as input
efa_loadings <- EFA(data=data_RSE, Nfactors=3, extraction = 'ml', verbose=FALSE)$pattern

BIFACTOR(loadings = efa_loadings)


# when a correlation matrix is provided, but not loadings or rawdata
BIFACTOR(cormat = cor(data_RSE), Ncases = 800, Nfactors = 4,
         bifactor_kind = 'SL',
         schmid_options = list(extraction = 'minres',
                               rotation = 'oblimin'))

# when rawdata is provided, but not loadings or cormat
```

```
BIFACTOR(rawdata = data_RSE, Nfactors = 3,
         bifactor_kind = 'bifactorT',
         EFA_options = list(extraction = 'minres', rotation = 'oblimin'))

# confirmatory bifactor analyses
BIFACTOR(rawdata = data_RSE,
         bifactor_kind = 'CFA',
         LV_options = list(group_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2),
                           N_group_factors = NULL,
                           estimator = 'WLSMV',
                           rotation = 'bigeomin',
                           resid_correls = NULL,
                           LV_names=NULL,
                           ordered = TRUE),
         min_loading = .0001)

# See this web page for additional Examples:

# https://oconnor-psych.ok.ubc.ca/vignettes/EXAMPLES_vignettes.html
```

COMPLEXITY *Factor solution complexity*

Description

Provides Hoffman's (1978) complexity coefficient for each item and (optionally) the percent complexity in the factor solution using the procedure and code provided by Pettersson and Turkheimer (2014).

Usage

```
COMPLEXITY(loadings, percent=TRUE, degree.change=100, averaging.value=100, verbose=TRUE)
```

Arguments

loadings	The factor loading matrix.
percent	(logical) Should the percent complexity be computed? The default = TRUE.
degree.change	If percent=TRUE, the number of incremental changes toward simple structure. The default = 100.
averaging.value	If percent=TRUE, the number of repeats per unit of degree change. The default = 100.
verbose	(logical) Should detailed results be displayed in console? The default = TRUE.

Details

This function provides Hoffman's (1978) complexity coefficient for each item and (optionally) the percent complexity in the factor solution using the procedure and code provided by Pettersson and Turkheimer (2014). For the percent complexity coefficient, values closer to zero indicate greater consistency with simple structure.

Value

A list with the following elements:

comp_rows	The complexity coefficient for each item
percent	The percent complexity in the factor solution

Author(s)

Brian P. O'Connor

References

Hofmann, R. J. (1978). Complexity and simplicity as objective indices descriptive of factor solutions. *Multivariate Behavioral Research, 13,* 247-250.

Pettersson E, Turkheimer E. (2010) Item selection, evaluation, and simple structure in personality data. *Journal of research in personality, 44(4),* 407-420.

Pettersson, E., & Turkheimer, E. (2014). Self-reported personality pathology has complex structure and imposing simple structure degrades test information. *Multivariate Behavioral Research, 49(4),* 372-389.

Examples

```
# the Harman (1967) correlation matrix
PCAoutput <- PCA(data_Harman, Nfactors = 2, Ncases = 305, rotation='promax', verbose=FALSE)
COMPLEXITY(loadings=PCAoutput$structure, verbose=TRUE)

# Rosenberg Self-Esteem scale items
PCAoutput <- PCA(data_RSE, Nfactors = 2, rotation='promax', verbose=FALSE)
COMPLEXITY(loadings=PCAoutput$structure, verbose=TRUE)

# NEO-PI-R scales
PCAoutput <- PCA(data_NEOPIR, Nfactors = 5, rotation='promax', verbose=FALSE)
COMPLEXITY(loadings=PCAoutput$structure, verbose=TRUE)
```

CONGRUENCE *Factor solution congruence*

Description

Aligns two factor loading matrices and computes the factor solution congruence and the root mean square residual.

Usage

```
CONGRUENCE(target, loadings, verbose)
```

Arguments

target	The target loading matrix.
loadings	The loading matrix that will be aligned with the target.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

The function first searches for the alignment of the factors from the two loading matrices that has the highest factor solution congruence. It then aligns the factors in "loadings" with the factors in "target" without changing the loadings. The alignment is based solely on the positions and directions of the factors. The function then produces the Tucker-Wrigley-Neuhaus factor solution congruence coefficient as an index of the degree of similarity between between the aligned loading matrices (see Guadagnoli & Velicer, 1991; and ten Berge, 1986, for reviews).

An investigation by Lorenzo-Seva and ten Berge (2006) resulted in the following conclusion: A congruence coefficient "value in the range .85.94 corresponds to a fair similarity, while a value higher than .95 implies that the two factors or components compared can be considered equal."

Value

A list with the following elements:

rcBefore	The factor solution congruence before factor alignment
rcAfter	The factor solution congruence after factor alignment
rcFactors	The congruence for each factor
rmsr	The root mean square residual
residmat	The residual matrix
loadingsNew	The aligned loading matrix

Author(s)

Brian P. O'Connor

References

Guadagnoli, E., & Velicer, W. (1991). A comparison of pattern matching indices. *Multivariate Behavior Research, 26,* 323-343.

ten Berge, J. M. F. (1986). Some relationships between descriptive comparisons of components from different studies. *Multivariate Behavioral Research, 21,* 29-40.

Lorenzo-Seva, U., & ten Berge, J. M. F. (2006). Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences, 2(2),* 5764.

Examples

```
# Rosenberg Self-Esteem scale items
loadings <- PCA(data_RSE[1:150,],   corkind='pearson', Nfactors = 3,
                rotation='varimax', verbose=FALSE)

target   <- PCA(data_RSE[151:300,], corkind='pearson', Nfactors = 3,
                rotation='varimax', verbose=FALSE)
CONGRUENCE(target = target$loadingsROT, loadings = loadings$loadingsROT, verbose=TRUE)


# NEO-PI-R scales
loadings <- PCA(data_NEOPIR[1:500,], corkind='pearson', Nfactors = 3,
                rotation='varimax', verbose=FALSE)

target <- PCA(data_NEOPIR[501:1000,], corkind='pearson', Nfactors = 3,
              rotation='varimax', verbose=FALSE)

CONGRUENCE(target$loadingsROT, loadings$loadingsROT, verbose=TRUE)
```

CORRECTED_CORRELS *Corrected Pearson correlation coefficients*

Description

Corrects Pearson correlation coefficients for attenuation due to unreliability and tests for overall differences between the original and corrected correlation matrices.

Usage

```
CORRECTED_CORRELS(cormat, Ncases = 100, alphas,
                  systematic=FALSE, Nperms = 1000, verbose=TRUE)
```

Arguments

cormat	A square correlation matrix.
Ncases	The number of cases upon which cormat is based. Example: Ncases = 250
alphas	A vector of reliability coefficients (e.g., Cronbach alpha values) for the variables in cormat, in the same order as the variables in cormat. Example: alphas <- c(.81, .95, .64, .49, .36)
systematic	The kind of permutations for the tests for significance differences between the uncorrected and corrected correlation matrices. The options are TRUE (for all possible permutations), or FALSE (the default), in which case random matrix permutations will be conducted.
Nperms	(optional) The number of random matrix permutation for when systematic = FALSE. Example: Nperms = 1000

verbose	(optional) Should detailed results be displayed in console? TRUE (default) or FALSE

Details

This function uses the simple, traditional formula for correcting correlations for attentuation due to unreliability, as described in Schmidt and Hunter (1996), Schmitt (1996), and Trafimow (2016).

Spearman's rho and Kendall's tau correlations between the values in the entered and corrected correlation matrices are provided when there are three or more variables. These are rank correlations that do not assume that similarity increases are linear and which are more robust to outliers than are Pearson correlations. The corresponding permutation significance tests of overall equality between the entered and corrected correlation matrices are "Mantel" tests.

The Steiger (1980) and Jennrich (1970) tests of the overall equality of the entered and corrected correlation matrices are conducted using functions from the psych package (Revelle, 2023).

See the MatrixCorrelation package for additional tests of differences between correlation matrices.

Value

A list with the following elements:

cormat	The entered correlation matrix
alphas	The entered alphas
corxd	The matrix of corrected correlations
resids	The differences between the entered and corrected correlations
data_in_rows	The input data and results in row format
cor_spearman	The Spearman's rho correlation between the entered and corrected correlations
cor_kendall	The Kendall's tau correlation between the entered and corrected correlations
Steiger_test	Results of the Steiger test for equality of the entered and corrected correlation matrices
Jennrich_test	Results of the Jennrich test for equality of the entered and corrected correlation matrices

Author(s)

Brian P. O'Connor

References

Jennrich, R. I. (1970) An asymptotic test for the equality of two correlation matrices. *Journal of the American Statistical Association, 65,* 904-912.

Revelle, W. (2023). psych: Procedures for Psychological, Psychometric, and Personality Research. R package version 2.3.6, https://CRAN.R-project.org/package=psych

Schmidt, F. L., & Hunter, J. E. (1996). Measurement error in psychological research: Lessons from 26 research scenarios. *Psychological Methods, 1(2),* 199223.

Schmitt, N. (1996). Uses and abuses of coefficient alpha. *Psychological Assessment, 8(4),* 350353.

Steiger, J. H. (1980). Testing pattern hypotheses on correlation matrices: Alternative statistics and some empirical results. *Multivariate Behavioral Research, 15,* 335-352.

Trafimow, D. (2016). The attenuation of correlation coefficients: A statistical literacy issue. *Teaching Statistics, 38(1),* 2528.

Examples

```
# data from Schmitt (1996)
cormat <- '
 1.0
 .70  1.0
 .51  .47  1.0
 .52  .55  .45  1.0
 .32  .35  .31  .28  1.0'
cormat_noms <- c('v1', 'v2', 'v3', 'v4', 'v5')
cormat <- data.matrix( read.table(text=cormat, fill=TRUE, col.names=cormat_noms,
                                  row.names=cormat_noms ))
cormat[upper.tri(cormat)] <- t(cormat)[upper.tri(cormat)]

alphas <- c(.81, .95, .64, .49, .36)

CORRECTED_CORRELS(cormat=cormat, alphas=alphas,
                  Nperms = 10000, systematic=FALSE, verbose=TRUE)
```

`data_Field` *data_Field*

Description

A data frame with scores on 23 variables for 2571 cases. This is a simulated dataset that has the exact same correlational structure as the "R Anxiety Questionnaire" data used by Field et al. (2012) in their chapter on Exploratory Factor Analysis.

Usage

```
data(data_Field)
```

Source

Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using R.* Los Angeles, CA: Sage.

Examples

```
# MAP test
MAP(data_Field, corkind='pearson', verbose=TRUE)

# DIMTESTS
DIMTESTS(data_Field, corkind='pearson',
         tests = c('CD','EMPKC','HULL','RAWPAR','NEVALSGT1'), display=2)

# principal components analysis
PCA(data_Field, corkind='pearson', Nfactors=4, rotation='none', verbose=TRUE)
```

`data_Harman` *Correlation matrix from Harman (1967, p. 80).*

Description

The correlation matrix for eight physical variables for 305 cases from Harman (1967, p. 80).

Usage

```
data(data_Harman)
```

References

Harman, H. H. (1967). *Modern factor analysis (2nd. ed.).* Chicago: University of Chicago Press.

Examples

```
# MAP test on the Harman correlation matrix
MAP(data_Harman, verbose=TRUE)

# DIMTESTS on the Harman correlation matrix
DIMTESTS(data_Harman, tests = c('EMPKC','HULL','RAWPAR','NEVALSGT1'), Ncases=305, display=2)

# parallel analysis of the Harman correlation matrix
RAWPAR(data_Harman, extraction='PCA', Ndatasets=100, percentile=95,
       Ncases=305, verbose=TRUE)
```

data_HS_1939 *Holzinger and Swineford (1939) data for 9 mental ability test scores*

Description

Raw data for 9 mental ability test scores in 301 seventh- and eighth-grade children, from two schools. The data were provided by Holzinger and Swineford (1939), adapted from the MBESS package.

test1 = Visual perception

test2 = Cubes

test3 = Lozenges

test4 = Paragraph comprehension

test5 = Sentence completion

test6 = Word meaning

test7 = Speeded addition

test8 = Speeded counting of dots

test9 = Speeded discrimination straight and curved capitals

school = 'Grant-White' or 'Pasteur'

Usage

```
data(data_HS_1939)
```

References

Holzinger, K., and Swineford, F. (1939). A study in factor analysis: The stability of a bifactor solution. *Supplementary Educational Monograph, no. 48.* Chicago: University of Chicago Press.

Examples

```
head(data_HS_1939)
```

data_NEOPIR *data_NEOPIR*

Description

A data frame with scores for 1000 cases on 30 variables that have the same intercorrelations as those for the Big 5 facets on pp. 100-101 of the NEO-PI-R manual (Costa & McCrae, 1992).

Usage

```
data(data_NEOPIR)
```

References

Costa, P. T., & McCrae, R. R. (1992). *Revised NEO personality inventory (NEO-PIR) and NEO five-factor inventory (NEO-FFI): Professional manual.* Odessa, FL: Psychological Assessment Resources.

Examples

```
# MAP test on the data_NEOPIR data
MAP(data_NEOPIR, corkind='pearson', verbose=TRUE)

# DIMTESTS on the data_NEOPIR data
DIMTESTS(data_NEOPIR, tests = c('EMPKC','HULL','RAWPAR','NEVALSGT1'), Ncases=1000, display=2)

# parallel analysis of the data_NEOPIR data
RAWPAR(data_NEOPIR, extraction='PCA', Ndatasets=100, percentile=95,
       corkind='pearson', verbose=TRUE)
```

---

data_RSE *Item-level dataset for the Rosenberg Self-Esteem scale*

Description

A data frame with 800 observations on the 10 items from the Rosenberg Self-Esteem scale. Negatively-worded items were reverse-coded.

Usage

```
data(data_RSE)
```

References

Rosenberg, M. (1965). *Society and the adolescent self-image.* Princeton University Press.

Examples

```
head(data_RSE)
```

data_RSE_not_recoded *Item-level dataset for the Rosenberg Self-Esteem scale*

Description

A data frame with 300 observations on the 10 items from the Rosenberg Self-Esteem scale. Negatively-worded items are not reverse-coded.

Usage

```
data(data_RSE_not_recoded)
```

References

Rosenberg, M. (1965). *Society and the adolescent self-image.* Princeton University Press.

Examples

```
head(data_RSE_not_recoded)
```

data_RSE_sex *Item-level dataset for the Rosenberg Self-Esteem scale plus gender*

Description

A data frame with 400 observations for males and 400 observations for females on the 10 items from the Rosenberg Self-Esteem scale. Negatively-worded items were reverse-coded.

Usage

```
data(data_RSE_sex)
```

References

Rosenberg, M. (1965). *Society and the adolescent self-image.* Princeton University Press.

Examples

```
head(data_RSE_sex)
```

data_SDT *data_SDT*

Description

A data frame with scores for 366 cases on 12 variables that have the same correlational structure as those reported by Snchez-Oliva et al. (2017). The data were generated using the values in their Table 1, p. 179. There are four scores for each of three needs, pertaining to autonomy, competence, and relatedness at work.

Usage

```
data(data_SDT)
```

References

Snchez-Oliva, D., Morin, A. J. S., Teixeira, P. J., Carraa, E. V., Palmeira, A. L., & Silva, M. N. (2017). A bifactor exploratory structural equation modeling representation of the structure of the Basic Psychological Needs at Work Scale. *Journal of Vocational Behavior, 98,* 173187.

Examples

```
head(data_SDT)
```

data_TabFid *data_TabFid*

Description

A data frame with scores for 340 cases on 44 Bem Sex Role Inventory items, used by Tabacknick & Fidell (2013, p. 656) in their chapter on exploratory factor analysis.

Usage

```
data(data_TabFid)
```

References

Tabachnik, B. G., & Fidell, L. S. (2013). *Using multivariate statistics.* New York, NY: Pearson.

Examples

```
# MAP test on the data_TabFid data
MAP(data_TabFid, corkind='pearson', verbose=TRUE)

# parallel analysis of the data_TabFid data
RAWPAR(data_TabFid, extraction='PCA', Ndatasets=100, percentile=95,
       corkind='pearson', verbose=TRUE)

# DIMTESTS on the data_TabFid data
DIMTESTS(data_TabFid, tests = c('EMPKC','HULL','RAWPAR'), corkind='pearson', display=1)

# principal axis factor analysis of the data_TabFid data
EFA(data_TabFid, corkind='pearson', extraction='paf', Nfactors = 5, iterpaf = 50,
      rotation='promax', ppower = 4, verbose=TRUE)
```

DIMTESTS *Tests for the number of factors*

Description

Conducts multiple tests for the number of factors

Usage

```
DIMTESTS(data, tests, corkind, Ncases, HULL_method, HULL_gof, HULL_cor_method,
          CD_cor_method, display=2)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
tests	A vector of the names of the tests for the number of factors that should be conducted. The possibilities are CD, EMPKC, HULL, MAP, NEVALSGT1, RAWPAR, SALIENT, SESCREE, SMT. If tests is not specified, then tests = c('EMPKC', 'HULL', 'RAWPAR') is used as the default.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases. Required only if data is a correlation matrix.
HULL_method	From EFAtools: The estimation method to use. One of "PAF" (default), "ULS", or "ML", for principal axis factoring, unweighted least squares, and maximum likelihood

`HULL_gof` From EFAtools: The goodness of fit index to use. Either "CAF" (default), "CFI", or "RMSEA", or any combination of them. If method = "PAF" is used, only the CAF can be used as goodness of fit index. For details on the CAF, see Lorenzo-Seva, Timmerman, and Kiers (2011).

`HULL_cor_method` From EFAtools: The kind of correlation matrix to be used for the Hull method analyses. The options are 'pearson', 'kendall', and 'spearman'

`CD_cor_method` From EFAtools: The kind of correlation matrix to be used for the CD method analyses. The options are 'pearson', 'kendall', and 'spearman'

`display` The results to be displayed in the console: 0 = nothing; 1 = only the # of factors for each test; 2 (default) = detailed output for each test

Details

This is a convenience function for running possibly multiple tests for the number of factors.

See this web page for descriptions of the various tests:

https://oconnor-psych.ok.ubc.ca/vignettes/Number_of_factors_tests_vignettes.html

Value

A list with the following elements:

`dimtests` A matrix with the DIMTESTS results

`NfactorsDIMTESTS` The number of factors according to the first test method specified in the "tests" vector

Author(s)

Brian P. O'Connor

References

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. *Psychological Methods, 24(4),* 468-491.

Lorenzo-Seva, U., Timmerman, M. E., & Kiers, H. A. (2011). The Hull method for selecting the number of common factors. *Multivariate Behavioral Research, 46(2),* 340-364.

O'Connor, B. P. (2000). SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test. *Behavior Research Methods, Instrumentation, and Computers, 32,* 396-402.

Ruscio, J., & Roche, B. (2012). Determining the number of factors to retain in an exploratory factor analysis using comparison data of known factorial structure. *Psychological Assessment, 24,* 282292. doi: 10.1037/a0025697

Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychological Bulletin, 99,* 432-442.

Examples

```
# the Harman (1967) correlation matrix
DIMTESTS(data_Harman, tests = c('EMPKC','HULL','RAWPAR'), corkind='pearson',
                                Ncases = 305, display=2)

# Rosenberg Self-Esteem scale items, all possible DIMTESTS
DIMTESTS(data_RSE,
      tests = c('CD','EMPKC','HULL','MAP','NEVALSGT1','RAWPAR','SALIENT','SESCREE','SMT'),
      corkind='pearson', display=2)

# Rosenberg Self-Esteem scale items, using polychoric correlations
DIMTESTS(data_RSE, corkind='polychoric', display=2)

# NEO-PI-R scales
DIMTESTS(data_NEOPIR, tests = c('EMPKC','HULL','RAWPAR','NEVALSGT1'), display=2)
```

EFA *Exploratory factor analysis*

Description

Exploratory factor analysis with multiple options for factor extraction and rotation

Usage

```
EFA(data, Nfactors=NULL, extraction = 'paf', rotation='promax', corkind='pearson',
    Ncases=NULL, iterpaf=100, ppower = 3,
    GPA_options = list(delta = .01,
                       epsilon = .00001,
                       normalize = FALSE,
                       maxit = 1000,
                       randomStarts = 50),
    verbose=TRUE)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal.The function internally determines whether the data are a correlation matrix.
Nfactors	The number of factors to extract. If not specified, then the EMPKC procedure will be used to determine the number of factors.
extraction	The factor extraction method for the analysis. The options are 'paf' (the default), 'alpha', 'fullinfo', 'gls', 'image', 'minres', 'ml', 'ols', 'uls', and 'wls'.

rotation	The factor rotation method for the analysis. The orthogonal rotation options are: 'varimax' (the default), 'bentlerT', 'entropy', 'equamax', 'geominT', 'quartimax', and 'none'. The oblique rotation options are: 'promax' (the default), 'bentlerQ', 'geominQ', 'oblimin', 'oblimax', 'quartimin', 'simplimax', and 'none'.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases. Required only if data is a correlation matrix.
iterpaf	The maximum number of iterations for paf.
ppower	The power value to be used in a promax rotation (required only if rotation = 'promax'). Suggested value: 3
GPA_options	(optional) A list with options for when a GPArotation is used. See the Details section below.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

The factor extraction computations for the following methods are conducted using the psych package (Revelle, 2026): 'alpha', 'gls', 'minres', 'ols', 'uls', and 'wls'.

The factor extraction computations for 'fullinfo' are conducted using the mirt package (Chalmers, 2012). Full-information methods are considered more appropriate for item-level data than other factor extraction methods (Wirth & Edwards, 2007).

The factor rotation computations for the following methods are conducted using the GPArotation package (Bernaards & Jennrich, 2005, 2026): 'bentlerQ', 'bentlerT', 'entropy', 'geominQ', 'geominT', 'oblimax', 'oblimin', 'quartimax', 'quartimin', and 'simplimax'.

For factor extraction (see Mulaik, 2010, for a review):

- **alpha** is for an alpha factor analysis
- **gls** is for a generalized weighted least squares factor analysis
- **image** is for image factor analysis
- **minres** is for a minimum residual factor analysis
- **ml** is for maximum likelihood factor analysis
- **ols** is for an ordinary least squares factor analysis
- **paf** is for principal axis factor analysis
- **uls** is for an unweighted least squares factor analysis
- **wls** is for a weighted least squares factor analysis

For factor rotation (see Jennrich, 2018, for a review):

- **bentlerQ** is an oblique rotation based on Bentler's invariant pattern simplicity criterion.

- **bentlerT** is an orthogonal rotation based on Bentler's invariant pattern simplicity criterion
- **entropy** entropy is an orthogonal factor rotation that pushes factor loadings to be either strongly dominant (close to 1.0) or cleanly absent (close to 0.0), reducing the overall informational "noise" of the matrix.
- **equamax** is an orthogonal rotation designed as a mathematical compromise between varimax and quartimax.
- **geominQ** is an oblique factor rotation method that find a clean "simple structure" even when the data contains highly complex variables (variables that naturally load on multiple factors).
- **geominT** is an orthogonal factor rotation that combines the strict geometric constraint of uncorrelated factors with Brownes flexible, product-based geomin complexity criterion.
- **oblimax** is an oblique factor rotation method that maximizes the number of very high and very low (near-zero) factor loadings.
- **oblimin** is an oblique factor rotation that allows factors to correlate freely to achieve the simplest possible loading structure.
- **promax** is a two-stage oblique factor rotation method.
- **quartimax** is an orthogonal factor rotation designed to spread the explained variance evenly across the factors, actively preventing a single general factor from dominating.
- **simplimax** is an oblique factor rotation that attempts to recover complex or messy "simple structures" where traditional continuous methods like oblimin or geomin fail.
- **quartimin** is an oblique factor rotation that focuses on simplifying the rows of the loading matrix.
- **varimax** is an orthogonal factor rotation that maximizes the variance of the squared factor loadings within each individual factor column.

For **GPA_options:**

A list with options for when a GPArotation is used. The list elements are 'delta', 'epsilon', 'normalize', 'maxit', and 'randomStarts'.

- **delta** is a small positive number that is added to the squared factor loadings in the (geomin) optimization denominator to prevent division by zero and control the sharpness of the penalty near zero. (Note: In MPLus, delta is called "epsilon".)
- **epsilon** is the numeric rotational convergence tolerance threshold. It controls how precise or strict the stopping rule is for the rotation algorithm. The gradient projection algorithm stops optimizing and assumes the rotation has converged when the norm of the gradient falls below this value.
- **normalize** specifies whether and how the loadings matrix is normalized prior to rotation and denormalized after rotation. The possibilities are FALSE (the default) for no normalization, or TRUE, for the Kaiser/Horst normalization.
- **maxit** is the maximum number of iterations.
- **randomStarts** is an integer that specifies the number of random starting configurations used to optimize the factor rotation. It helps avoid getting trapped in local minima during factor rotation optimization. It runs the gradient projection algorithm multiple timeseach using a different random orthogonal or oblique starting rotation matrix (Tmat) generated via QR decomposition. It returns the rotated factor loadings matrix and results corresponding to the run with the lowest objective criterion value among all attempts.

See this web page for more detailed descriptions of the above extraction and rotation methods:
https://oconnor-psych.ok.ubc.ca/vignettes/EFA_BIFACTOR_vignettes.html

Value

A list with the following elements:

loadingsNOROT	The unrotated factor loadings
loadingsROT	The rotated factor loadings
pattern	The pattern matrix
structure	The structure matrix
phi	The correlations between the factors
varexplNOROT1	The initial eigenvalues and total variance explained
varexplNOROT2	The eigenvalues and total variance explained after factor extraction (no rotation)
varexplROT	The rotation sums of squared loadings and total variance explained for the rotated loadings
cormat_reprod	The reproduced correlation matrix, based on the rotated loadings
fit_coefs	Model fit coefficients
chisqMODEL	The model chi squared
dfMODEL	The model degrees of freedom
pvalue	The model p-value
chisqNULL	The null model chi squared
dfNULL	The null model degrees of freedom
communalities	The unrotated factor solution communalities
uniquenesses	The unrotated factor solution uniquenesses

Author(s)

Brian P. O'Connor

References

Bernaards, C. A., & Jennrich, R. I. (2005). Gradient Projection Algorithms and Software for Arbitrary Rotation Criteria in Factor Analysis. *Educational and Psychological Measurement, 65(5),* 676-696. https://doi.org/10.1177/0013164404272507

Bernaards, C. A., & Jennrich, R. I. (2026). GPArotation: Gradient Projection Factor Rotation. R package version 2026.4-1, https://CRAN.R-project.org/package=GPArotation

Chalmers, R. P. (2012). mirt: A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software, 48(6),* 129. doi:10.18637/jss.v048.i06.

Jennrich, R. I. (2018). Rotation. In P. Irwing, T. Booth, & D. J. Hughes (Eds.), *The Wiley handbook of psychometric testing: A multidisciplinary reference on survey, scale and test development* (pp.

279304). Wiley Blackwell. https://doi.org/10.1002/9781118489772.ch10

Mulaik, S. A. (2010). *Foundations of factor analysis* (2nd ed.). Boca Raton, FL: Chapman and Hall/CRC Press, Taylor & Francis Group.

Revelle, W. (2026). psych: Procedures for Psychological, Psychometric, and Personality Research. R package version 2.6.5, https://CRAN.R-project.org/package=psych

Sellbom, M., & Tellegen, A. (2019). Factor analysis in psychological assessment research: Common pitfalls and recommendations. *Psychological Assessment, 31(12),* 1428-1441. https://doi.org/10.1037/pas0000623

Watts, A. L., Greene, A. L., Ringwald, W., Forbes, M. K., Brandes, C. M., Levin-Aspenson, H. F., & Delawalla, C. (2023). Factor analysis in personality disorders research: Modern issues and illustrations of practical recommendations. *Personality Disorders: Theory, Research, and Treatment, 14(1),* 105-117. https://doi.org/10.1037/per0000581

Wirth, R. J., & Edwards, M. C. (2007). Item factor analysis: current approaches and future directions. *Psychological methods, 12(1),* 58-79. https://doi.org/10.1037/1082-989X.12.1.58

Examples

```
# the Harman (1967) correlation matrix
EFA(data=data_Harman, extraction = 'paf', Nfactors=2, Ncases=305, rotation='oblimin')

# Rosenberg Self-Esteem scale items, using ml extraction & polychoric correlations
EFA(data=data_RSE, Nfactors=2, extraction = 'ml', corkind='polychoric')

# NEO-PI-R scales
EFA(data=data_NEOPIR, Nfactors=5, extraction='minres', rotation='promax')
```

EFA.dimensions-defunct

Defunct functions

Description

The functions below are defunct and have been removed and replaced.

Details

- PA_FA has been replaced by EFA
- MAXLIKE_FA has been replaced by EFA
- IMAGE_FA has been replaced by EFA

Author(s)

Brian P. O'Connor <brian.oconnor@ubc.ca>

EFA_SCORES *Exploratory factor analysis scores*

Description

Factor scores, and factor score indeterminacy coefficients, for exploratory factor analysis

Usage

```
EFA_SCORES(loadings=NULL, loadings_type='structure', data=NULL, cormat=NULL,
           corkind='pearson', phi=NULL, method = 'Thurstone', verbose = TRUE)
```

Arguments

loadings	The factor loadings. Required for all methods except PCA.
loadings_type	(optional) The kind of factor loadings. The options are 'structure' (the default) or 'pattern'. Use 'structure' for orthogonal loadings.
data	(optional) An all-numeric dataframe where the rows are cases & the columns are the variables. Required if factor scores for cases are desired.
cormat	(optional) The item/variable correlation matrix. Not required when "data" is provided.
corkind	(optional) The kind of correlation matrix to be used. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. The kind of correlation should be the same as the kind that was used to produce the "loadings".
phi	(optional) The factor correlations.
method	(optional) The method to be used for computing the factor scores (e.g., method = 'Thurstone'). The options are: • **Thurstone** for Thurstone's' (1935) least squares regression approach (Grice Equation 5), which is conducted on the factor structure loadings • **Harman** for Harman's' (1976) idealized variables (Grice Equation 10), which is conducted on the factor pattern loadings. • **Bartlett** for Bartlett's (1937) method (Grice Equation 9), which is conducted on the factor pattern loadings. • **tenBerge** for ten Berge et al's (1999) method, which is conducted on the factor pattern loadings and which generates scores with correlations that are identical to those in **phi** (Grice Equation 8) • **Anderson** for Anderson & Rubin (1956; see Gorsuch, 1983, p 265) scores, which is conducted on the factor pattern loadings and which is only appropriate for orthogonal factor models • **PCA** for unrotated principal component scores (requires only data or cormat).
verbose	(optional) Should detailed results be displayed in console? TRUE (default) or FALSE

Details

Before using factor scores, it is important to establish that there is an acceptable degree of "determinacy" for the computed factor scores (Grice, 2001; Waller, 2023).

The following descriptions of factor score indeterminacy are either taken directly from, or adapted from, Grice (2001):

"As early as the 1920s researchers recognized that, even if the correlations among a set of ability tests could be reduced to a subset of factors, the scores on these factors would be indeterminate (Wilson, 1928). In other words, an infinite number of ways for scoring the individuals on the factors could be derived that would be consistent with the same factor loadings. Under certain conditions, for instance, an individual with a high ranking on g (general intelligence), according to one set of factor scores, could receive a low ranking on the same common factor according to another set of factor scores, and the researcher would have no way of deciding which ranking is "true" based on the results of the factor analysis. As startling as this possibility seems, it is a fact of the mathematics of the common factor model.

The indeterminacy problem is not that the factor scores cannot be directly and appropriately computed; it is that an infinite number of sets of such scores can be created for the same analysis that will all be equally consistent with the factor loadings.

The degree of indeterminacy will not be equivalent across studies and is related to the ratio between the number of items and factors in a particular design (Meyer, 1973; Schonemann, 1971). It may also be related to the magnitude of the communalities (Gorsuch, 1983). Small amounts of indeterminacy are obviously desirable, and the con- sequences associated with a high degree of indeterminacy are extremely unsettling. Least palatable is the fact that if the maximum possible proportion of indeterminacy in the scores for a particular factor meets or exceeds 50 struct two orthogonal or negatively correlated sets of factor scores that will be equally consistent with the same factor loadings (Guttman, 1955).

MULTR & RSQR MULTR is the multiple correlation between each factor and the original variables (Green, 1976; Mulaik, 1976). MULTR ranges from 0 to 1, with high values being desirable, and indicates the maximum possible degree of determinacy for factor scores. Some authors have suggested that MULTR values should be substantially higher than .707 which, when squared, would equal .50. RSQR is the square or MULTR and represents the maximum proportion of determinacy.

MINCOR

the minimum correlation that could be obtained between two sets of equally valid factor scores for each factor (Guttman, 1955; Mulaik, 1976; Schonemann, 1971). This index ranges from -1 to +1. High positive values are desirable. When MINCOR is zero, then two sets of competing factor scores can be constructed for the same common factor that are orthogonal or even negatively correlated. MINCOR values approaching zero are distressing, and negative values are disastrous. MINCOR values of zero or less occur when MULTR <= .707 (at least 50 indeterminacy). MULTR values that do not appreciably exceed .71 are therefore particularly problematic. High values that approach 1.0 indicate that the factors may be slightly indeterminate, but the infinite sets of factor scores that could be computed will yield highly similar rankings of the individuals. In other words, the practical impact of the indeterminacy is minimal. MINCOR is the "Guttman's Indeterminacy Index" that is provided by the fsIndeterminacy function in the fungible package.

VALIDITY

While the MULTR values represent the maximum correlation between the factor score estimates and the factors, the VALIDITY coefficients represent the actual correlations between the factor

score estimates and their respective factors, which may be lower than MULTR. The VALIDITY coefficients may range from -1 to +1. They should be interpreted in the same manner as MULTR. Gorsuch (1983, p. 260) recommended values of at least .80, but much larger values (>.90) may be necessary if the factor score estimates are to serve as adequate substitutes for the factors themselves.

Correlational Accuracy

If the factor score estimates are adequate representations of the factors, then the correlations between the factor scores should be similar to the correlations between the factors."

Value

A list with the following elements:

`FactorScores`	The factor scores
`FSCoef`	The factor score coefficients (W)
`MULTR`	The multiple correlation between each factor and the original variables
`RSQR`	The square or MULTR, representing the maximum proportion of determinacy
`MINCOR`	Guttmans indeterminacy index, the minimum correlation that could be obtained between two sets of equally valid factor scores for each factor.
`VALIDITY`	The correlations between the factor score estimates and their respective factors
`UNIVOCALITY`	The extent to which the estimated factor scores are excessively or insufficiently correlated with other factors in the same analysis
`FactorScore_Correls`	The correlations between the factor scores
`phi`	The correlations between the factors
`pattern`	The pattern matrix
`pattern`	The structure matrix

Author(s)

Brian P. O'Connor

References

Anderson, R. D., & Rubin, H. (1956). Statistical inference in factor analysis. *Proceedings of the Third Berkeley Symposium of Mathematical Statistics and Probability, 5,* 111-150.

Bartlett, M. S. (1937). The statistical conception of mental factors. *British Journal of Psychology, 28,* 97-104.

Grice, J. (2001). Computing and evaluating factor scores. *Psychological Methods, 6(4),* 430-450.

Harman, H. H. (1976). *Modern factor analysis.* University of Chicago press.

ten Berge, J. M. F., Krijnen, W. P., Wansbeek, T., and Shapiro, A. (1999). Some new results on correlation-preserving factor scores prediction methods. *Linear Algebra and its Applications, 289(1-3),* 311-318.

Thurstone, L. L. (1935). *The vectors of mind.* Chicago: University of Chicago Press.

Waller, N. G. (2023). Breaking our silence on factor score indeterminacy. *Journal of Educational and Behavioral Statistics, 48(2),* 244-261.

Examples

```
efa_out <- EFA(data=data_RSE, extraction = 'ml', Nfactors=2, rotation='promax')

EFA_SCORES(loadings=efa_out$structure, loadings_type='structure', data=data_RSE,
           phi=efa_out$phi, method = 'tenBerge')


# PCA scores
EFA_SCORES(data=data_NEOPIR, method = 'PCA')
```

EMPKC *The empirical Kaiser criterion method*

Description

A test for the number of common factors using the Empirical Kaiser Criterion method (Braeken & van Assen, 2017).

Usage

```
EMPKC(data, corkind='pearson', Ncases=NULL, verbose=TRUE)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases. Required only if data is a correlation matrix.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

The code for this function was adapted from the code provided by Auerswald & Moshagen (2019).

From Braeken & van Assen (2017):

"We developed a new factor retention method, the Empirical Kaiser Criterion, which is directly linked to statistical theory on eigenvalues and to researchers' goals to obtain reliable scales. EKC is easily visualized, and easy to compute and apply (no specialized software or simulations are needed). EKC can be seen as a sample-variant of the original Kaiser criterion (which is only effective at the population level), yet with a built-in empirical correction factor that is a function of the variables-to-sample-size ratio and the prior observed eigenvalues in the series. The links with statistical theory and practically relevant scales allowed us to derive conditions under which EKC accurately retrieves the number of acceptable scales, that is, sufficiently reliable scales and strong enough items.

"Our simulations verified our derivations, and showed that (a) EKC performs about as well as parallel analysis for data arising from the null, 1-factor, or orthogonal factors model; and (b) clearly outperforms parallel analysis for the specific case of oblique factors, particularly whenever inter-factor correlation is moderate to high and the number of variables per factor is small, which is characteristic of many applications these days. Moreover, additional simulations suggest that our method for predicting conditions of accurate factor retention also work for the more computer- intensive methods ... The ease-of-use and effectiveness of EKC make this method a prime candidate for replacing parallel analysis, and the original Kaiser criterion that, although it empirically does not perform too well, is still the number one method taught in introductory multivariate statistics courses and the default in many commercial software packages. Furthermore, the link to statistical theory opens up possibilities for generic power curves and sample size planning for exploratory factor analysis studies.

"Generally, the EKC accurately retrieved the number of factors in conditions whenever it was predicted to work well, and its performance was worse when it was not predicted to work well. More precisely, hit rate or power exceeded .8 in accordance with predictions under the null model, 1-factor model, the orthogonal factor model, and the oblique factor model with more than three variables per scale. Only in the case of minimal scales, that is, with three items per scale, did EKC sometimes not accurately retrieve the number of factors as predicted; dropping the restriction that eigenvalues should exceed 1 then mended EKC's performance. A general guideline for application that can be derived from our results (and would not need a study-specific power study), is that EKC will accurately retrieve the number of factors in samples of at least 100 persons, when there is no factor, one practically relevant scale, or up to five practically relevant uncorrelated scales with a reliability of at least .8." (pp. 463-464)

From Auerswald & Moshagen (2019):

"The Empirical Kaiser Criterion (EKC; Braeken & van Assen, 2017) is an approach that incorporates random sample variations of the eigenvalues in Kaiser's criterion. On a population level, the criterion is equivalent to Kaiser's criterion and extractions all factors with associated eigenvalues of the correlation matrix greater than one. However, on a sample level, the criterion takes the distribution of eigenvalues for normally distributed data into account." (p. 474)

Value

The number of factors according to the EMPKC test.

Author(s)

Brian P. O'Connor

References

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. *Psychological Methods, 24(4),* 468-491.

Braeken, J., & van Assen, M. A. (2017). An empirical Kaiser criterion. *Psychological Methods, 22,* 450 - 466.

Examples

```
# the Harman (1967) correlation matrix
EMPKC(data_Harman, Ncases = 305)

# Rosenberg Self-Esteem scale items, using polychoric correlations
EMPKC(data_RSE, corkind='polychoric')

# NEO-PI-R scales
EMPKC(data_NEOPIR)
```

ESEM *Exploratory Structural Equation Modeling*

Description

An easy-to-use function for exploratory structural equation measurement models via lavaan

Usage

```
ESEM(data, method = 'startvalues',
     Nfactors = NULL, extraction = 'ml',
     rotation_EFA = 'oblimin',
     rotation_LV  = 'geomin',
     corkind = 'pearson', anchors = NULL, estimator = 'ML',
     ordered = FALSE, target = NULL, target_keys = NULL,
     bifactor = FALSE, verbose = TRUE)
```

Arguments

data	A dataframe with the variables (and no other variables). Example: data = data_RSE

`method` When method = 'efa_blocks' (the default), the function uses the lavaan EFA blocks method. When method = 'startvalues', the function will first run an EFA (using the Nfactors, extraction, and rotation arguments) to obtain starting values and anchors for the ESEM analyses.

`Nfactors` The number of factors to extract. If Nfactors is not specified, and method = 'startvalues', then the EMPKC function is used to determine the number of factors.

`extraction` The factor extraction method for the analysis. The options are 'ml' (the default), 'paf', 'image', 'minres', 'uls', 'ols', 'wls', 'gls', 'alpha', and 'fullinfo'.

`rotation_EFA` The factor rotation method for the EFA analysis (if conducted).
The orthogonal rotation options are: 'varimax' (the default), 'quartimax', 'bentlerT', 'equamax', 'geominT', 'bigeominT', 'bifactorT', 'entropy', and 'none'.
The oblique rotation options are: 'promax' (the default), 'quartimin', 'oblimin', 'oblimax', 'simplimax', 'bentlerQ', 'geominQ', 'bigeominQ', 'bifactorQ', and 'none'.

`rotation_LV` The factor rotation method for the lavaan cfa analysis (if conducted). The options are varimax, quartimax, orthomax, oblimin, quartimin, geomin (the default), promax, entropy, mccammon, infomax, tandem2, oblimax, bentler, simplimax, target, cf, crawford-ferguson, cf-quartimax, cf-varimax, cf-equamax, cf-parsimax, and cf-facparsim.

`corkind` The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson' (the default), 'kendall', 'spearman', 'gamma', and 'polychoric'.

`anchors` (optional) The names of the anchor variables in data, if any.
Example: anchors = c('v1', 'v5', 'v8')

`estimator` The name of the lavaan estimator to be used in the analyses.
The options for basic estimators (continuous data) are ML, GLS, WLS, DWLS, ULS, DLS, and PML.
The options for robust estimators are MLM, MLMVS, MLMV, MLF, MLR, WLSM, WLSMVS, WLSMV, ULSM, ULSMVS, and ULSMV.
Example: estimator = 'ML'

`ordered` (optional) Set ordered = TRUE for ordered, categorical data (e.g., binary or Likert scale responses). The default is FALSE, for continuous data.

`target` (optional) A target loading matrix to be used instead of a data-driven EFA loading matrix.

`target_keys` (optional) A vector of numbers that will be used to create a target loading matrix (instead of using the above "target" argument). The length of target_keys should be equal to the number of variables in data. Each value in target_keys indicates the factor that the ith variable should be free to load on. For example, c(2, 1, 3) would indicate that the first variable loads on the second factor, the second variable loads on the first factor, and the third variable loads on the third factor. The target matrix that is generated by target_keys will have 'NA' for all of the loading variables, which indicates that the actual loadings should be estimated in the analyses. All other elements in the created target will be zeros.
Example: target_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2)

	target_keys is used as the group factor keys when bifactor = TRUE.
`bifactor`	(logical) Should bifactor ESEM analyses be conducted? TRUE (default) or FALSE
`verbose`	(optional) Should detailed results be displayed in console? TRUE or FALSE (the default)

Details

This function is for lavaan measurement models and not for structural models.

The function can be run with just a single line of code, as in

ESEM(data=data_RSE, Nfactors = 2)

or

ESEM(data=data_RSE, target_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2))

It is the ESEM model code that distinguishes ESEM from a regular CFA. In a typical CFA, an equation is written for each latent variable specifying the items or variables that load onto each latent variable. Cross-loadings are usually not provided or are not possible in CFA. In ESEM, longer, more elaborate equations for each latent variable are created. These equations contain coefficients/loadings for every item/variable on each latent variable. The coefficients come from either a preliminary exploratory factor analysis or from a specific target loading matrix.

Inside the ESEM function, there are two broad steps in the analyses: preparing ESEM model code, and then running the lavaan cfa function (Rosseel, 2012) using the prepared ESEM model code.

method = 'startvalues' provides more user options (via the extraction and rotation arguments) than does the default method = 'efa_blocks'. The results from the two methods are usually quite similar.

When method = 'startvalues', the preliminary exploratory factor analysis is empirically driven. The extraction method is usually ML, or a robust version of ML. The rotation method is usually oblique geomin, which has an epsilon value that researchers can adjust to reduce the size of cross-loadings and factor correlations. Morin (2023) recommended using an epsilon value of .5 to maximally reduce factor correlations, which in turn helps to obtain more accurate estimates of relations between constructs. The ESEM function in this package uses ML as the default factor extraction method and geominT (oblique) as the default rotation method.

When a target loading matrix is provided (for either method = 'startvalues' or method = 'efa_blocks'), the analyses are more deliberately guided by the researcher's theoretical assumptions or perhaps by a prior factor model for the data. "When using a target rotation, researchers are able to specify the main indicators for each construct, allowing these loadings to be freely estimated, but 'targeting' all cross-loadings to be as close to zero as possible while allowing them to be freely estimated. When based on a target rotation procedure, ESEM can be considered a fundamentally confirmatory method" (Swami et al., 2023, p. 7). For the ESEM function in this package, simply use the "target" argument, or the "target_keys"" argument, to provide a target loading matrix.

(When method = 'startvalues', the function uses the GPArotation package to conduct a targeted rotation of the EFA loadings instead of using geomin rotation. The rotation must be either 'targetQ' (oblique) or 'targetT' (orthogonal)).

When method = 'startvalues' and a target loading matrix is not provided, the ESEM function identifies the highest loading item/variable on each factor and uses these items as anchors in the subsequent CFA. The rotated loadings for non-anchor variables, in contrast, are treated as starting-point loadings in the CFA. The automatic anchor-finding method is by-passed if anchor variables are provided by the user on the "anchors" argument.

Value

A list with the loadings, the anchors, the target matrix, and the esem_model output.

Author(s)

Brian P. O'Connor

References

Afolabi, K. & Konold, T. R. (2024). The circle of methods for evaluating latent variable measurement models: EFA, CFA, and ESEM. *Practical Assessment, Research, and Evaluation, 29,* Article 15

Morin, A. J. (2023). Exploratory structural equation modeling. In R. Hoyle (Ed.), *Handbook of structural equation modeling* (2nd ed.). Guilford.

Prokofieva, M., Zarate, D., Parker, A., Palikara, O., & Stavropoulos, V. (2023). Exploratory structural equation modeling: A streamlined step by step approach using the R Project software. *BMC Psychiatry, 23(1),* Article 546. https://doi.org/10.1186/s12888-023-05028-9

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48,* 136.

Swami, V., Maano, C., & Morin, A. J. S. (2023). A guide to exploratory structural equation modeling (ESEM) and bifactor-ESEM in body image research. *Body Image, 47,* Article 101641. https://doi.org/10.1016/j.bodyim.2023.101641

Examples

```
ESEM(data=data_RSE, method = 'startvalues', Nfactors = 2)

# creating and using a target loading matrix for the ESEM
F1_target <- c(NA, NA, 0, NA, 0, NA, NA, 0, 0, 0)
F2_target <- c(0, 0, NA, 0, NA, 0, 0, NA, NA, NA)
RSE_target <- cbind(F1_target, F2_target)
rownames(RSE_target) <- names(data_RSE)
RSE_target

ESEM(data=data_RSE, method = 'startvalues', target = RSE_target)

# more simply: use target_keys to create a target loading matrix for the ESEM
ESEM(data=data_RSE, target_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2))
```

```
# See this web page for additional Examples:

# https://oconnor-psych.ok.ubc.ca/vignettes/EXAMPLES_vignettes.html
```

EXTENSION_FA *Extension factor analysis*

Description

Extension factor analysis, which provides correlations between nonfactored items and the factors that exist in a set of core items. The extension item correlations are then used to decide which factor, if any, a prospective item belongs to.

Usage

```
EXTENSION_FA(data, Ncore, Next, higherorder, roottest,
             corkind,
             extraction, rotation,
             Nfactors, NfactorsHO,
             Ndatasets, percentile,
             salvalue, numsals,
             iterpaf, ppower,
             verbose, factormodel, rotate)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables.
Ncore	An integer indicating the number of core variables. The function will run the factor analysis on the data that appear in column #1 to column #Ncore of the data matrix.
Next	An integer indicting the number of extension variables, if any. The function will run extension factor analyses on the remaining columns in data, i.e., using column #Ncore+1 to the last column in data. Enter zero if there are no extension variables.
higherorder	Should a higher-order factor analysis be conducted? The options are TRUE or FALSE.
roottest	The method for determining the number of factors. The options are: 'Nsalient' for number of salient loadings (see salvalue & numsals below); 'parallel' for parallel analysis (see Ndatasets & percentile below); 'MAP' for Velicer's minimum average partial test; 'SEscree' for the standard error scree test; 'nevals>1' for the number of eigenvalues > 1; and 'user' for a user-specified number of factors (see Nfactors & NfactorsHO below).
corkind	The kind of correlation matrix to be used. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'.

extraction	The factor extraction method. The options are: 'PAF' for principal axis / common factor analysis; 'PCA' for principal components analysis; 'ML' for maximum likelihood.
rotation	The factor rotation method. The options are: 'promax', 'varimax', and 'none'.
Nfactors	An integer indicating the user-determined number of factors (required only if roottest = 'user').
NfactorsHO	An integer indicating the user-determined number of higher order factors (required only if roottest = 'user' and higherorder = TRUE).
Ndatasets	An integer indicating the # of random data sets for parallel analyses (required only if roottest = 'parallel').
percentile	An integer indicating the percentile from the distribution of parallel analysis random eigenvalues to be used in determining the # of factors (required only if roottest = 'parallel'). Suggested value: 95
salvalue	The minimum value for a loading to be considered salient (required only if roottest = 'Nsalient'). Suggested value: .40
numsals	The number of salient loadings required for the existence of a factor i.e., the number of loadings > or = to salvalue (see above) for the function to identify a factor. Required only if roottest = 'Nsalient'. Gorsuch (1995a, p. 545) suggests: 3
iterpaf	The maximum # of iterations for a principal axis / common factor analysis (required only if extraction = 'PAF'). Suggested value: 100
ppower	The power value to be used in a promax rotation (required only if rotation = 'promax'). Suggested value: 3
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE
factormodel	(Deprecated.) Use 'extraction' instead.
rotate	(Deprecated.) Use 'rotation' instead.

Details

Traditional scale development statistics can produce results that are baffling or misunderstood by many users, which can lead to inappropriate substantive interpretations and item selection decisions. High internal consistencies do not indicate unidimensionality; item-total correlations are inflated because each item is correlated with its own error as well as the common variance among items; and the default number-of-eigenvalues-greater-than-one rule, followed by principal components analysis and varimax rotation, produces inflated loadings and the possible appearance of numerous uncorrelated factors for items that measure the same construct (Gorsuch, 1997a, 1997b). Concerned investigators may then neglect the higher order general factor in their data as they use misleading statistical output to trim items and fashion unidimensional scales.

These problems can be circumvented in exploratory factor analysis by using more appropriate factor analytic procedures and by using extension analysis as the basis for adding items to scales. Extension analysis provides correlations between nonfactored items and the factors that exist in a set of core items. The extension item correlations are then used to decide which factor, if any, a prospective item belongs to. The decisions are unbiased because factors are defined without being influenced by the extension items. One can also examine correlations between extension items and any higher order factor(s) in the core items. The end result is a comprehensive, undisturbed, and

informative picture of the correlational structure that exists in a set of core items and of the potential contribution and location of additional items to the structure.

Extension analysis is rarely used, at least partly because of limited software availability. Furthermore, when it is used, both traditional extension analysis and its variants (e.g., correlations between estimated factor scores and extension items) are prone to the same problems as the procedures mentioned above (Gorsuch, 1997a, 1997b). However, Gorusch (1997b) described how diagonal component analysis can be used to bypass the problems and uncover the noninflated and unbiased extension variable correlations – all without computing factor scores.

Value

A list with the following elements:

`fits1`	eigenvalues & fit coefficients for the first set of core variables
`rff`	factor intercorrelations
`corelding`	core variable loadings on the factors
`extcorrel`	extension variable correlations with the factors
`fits2`	eigenvalues & fit coefficients for the higher order factor analysis
`rfflding`	factor intercorrelations from the first factor analysis and the loadings on the higher order factor(s)
`ldingsef`	variable loadings on the lower order factors and their correlations with the higher order factor(s)
`extsef`	extension variable correlations with the lower order factor(s) and their correlations with the higher order factor(s)

Author(s)

Brian P. O'Connor

References

Dwyer, P. S. (1937) The determination of the factor loadings of a given test from the known factor loadings of other tests. *Psychometrika, 3,* 173-178.

Gorsuch, R. L. (1997a). Exploratory factor analysis: Its role in item analysis. *Journal of Personality Assessment, 68,* 532-560.

Gorsuch, R. L. (1997b). New procedure for extension analysis in exploratory factor analysis. *Educational and Psychological Measurement, 57,* 725-740.

Horn, J. L. (1973) On extension analysis and its relation to correlations between variables and factor scores. *Multivariate Behavioral Research, 8(4),* 477-489.

O'Connor, B. P. (2001). EXTENSION: SAS, SPSS, and MATLAB programs for extension analysis. *Applied Psychological Measurement, 25,* p. 88.

Examples

```
EXTENSION_FA(data_RSE, Ncore=7, Next=3, higherorder=TRUE,
             roottest='MAP',
             corkind='pearson',
             extraction='PCA', rotation='promax',
             Nfactors=2, NfactorsHO=1,
             Ndatasets=100, percentile=95,
             salvalue=.40, numsals=3,
             iterpaf=200,
             ppower=4,
             verbose=TRUE)

EXTENSION_FA(data_NEOPIR, Ncore=12, Next=6, higherorder=TRUE,
             roottest='MAP',
             corkind='pearson',
             extraction='PCA', rotation='promax',
             Nfactors=4, NfactorsHO=1,
             Ndatasets=100, percentile=95,
             salvalue=.40, numsals=3,
             iterpaf=200,
             ppower=4,
             verbose=TRUE)
```

FACTORABILITY *Factorability of a correlation matrix*

Description

Three methods for assessing the factorability of a correlation matrix

Usage

```
FACTORABILITY(data, corkind='pearson', Ncases=NULL, verbose=TRUE)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases for a correlation matrix. Required only if the entered data is a correlation matrix.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

This function provides results from three methods of assessing whether a dataset or correlation matrix is suitable for factor analysis:

1 – whether the determinant of the correlation matrix is > 0.00001;

2 – Bartlett's test of whether a correlation matrix is significantly different an identity matrix; and

3 – the Kaiser-Meyer-Olkin measure of sampling adequacy.

Value

A list with the following elements:

`chisq`	The chi-squared value for Bartlett,s test
`df`	The degrees of freedom for Bartlett,s test
`pvalue`	The significance level for Bartlett,s test
`Rimage`	The image correlation matrix
`KMO`	The overall KMO value
`KMOvars`	The KMO values for the variables

Author(s)

Brian P. O'Connor

References

Bartlett, M. S. (1951). The effect of standardization on a chi square approximation in factor analysis, *Biometrika, 38,* 337-344.

Cerny, C. A., & Kaiser, H. F. (1977). A study of a measure of sampling adequacy for factor-analytic correlation matrices. *Multivariate Behavioral Research, 12(1),* 43-47.

Dziuban, C. D., & Shirkey, E. C. (1974). When is a correlation matrix appropriate for factor analysis? *Psychological Bulletin, 81,* 358-361.

Kaiser, H. F., & Rice, J. (1974). Little Jiffy, Mark IV. *Educational and Psychological Measurement, 34,* 111-117.

Examples

```
FACTORABILITY(data_RSE, corkind='pearson')

FACTORABILITY(data_Field, corkind='pearson')
```

Factorial_Invariance *Factor Model Invariance*

Description

Assesses the equivalence of measurement and structural parameters of a CFA or ESEM factor model across multiple groups

Usage

```
Factorial_Invariance(data,
                     group,
                     LV_model=NULL,
                     LV_keys=NULL,
                     LV_names=NULL,
                     LV_resid_correls=NULL,
                     estimator = 'ML',
                     ordered = FALSE,
                     disp_options = list(models = c('Configural', 'Scalar'),
                                         coefs = c('std', 'LV_means')),
                     verbose=TRUE)
```

Arguments

data	A dataframe with the model variables and the group factor. Example: data = data_RSE_sex
group	The name of the group variable in data. Example: group = 'gender'
LV_model	(optional) lavaan package CFA model syntax. Example: model = ' pos_items =~ Q1 + Q2 + Q4 + Q6 + Q7 neg_items =~ Q3_R + Q5_R + Q8_R + Q9_R + Q10_R'
LV_keys	(optional) A vector or dataframe indicating which item (i.e., which variable in data) goes with which latent variable (which can be named or numbered). LV_keys is required when LV_model is not provided. If LV_keys is a dataframe, then the variable names must be in the first column and the factor memberships must be in the second column. Example of a vector entry: LV_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2)
LV_names	(optional) A vector with the preferred names for the latent variables. Example: LV_names = c('pos', 'neg')
LV_resid_correls	(optional) A vector with the pairs of correlated error terms (if any). Example: LV_resid_correls = c('Q1 ~~ Q2', 'Q9 ~~ Q10')
estimator	The name of the lavaan estimator to be used in the analyses.

The options for basic estimators (continuous data) are ML, GLS, WLS, DWLS, ULS, DLS, and PML.

The options for robust estimators are MLM, MLMVS, MLMV, MLF, MLR, WLSM, WLSMVS, WLSMV, ULSM, ULSMVS, and ULSMV.

Example: estimator = 'ML'

`ordered` (optional) are the variables in data (if provided) ordered? For example, Likert scale responses are usually ordered. The possible values for ordered are TRUE, FALSE, or a vector with the names of the ordered variables.

`disp_options` (optional) A list with display (print) options for model coefficients. The list elements are 'models' and 'coefs'.

The possible entries for 'models' are: 'mod_Configural', 'mod_Metric', 'mod_Scalar', 'mod_Strict', 'mod_LV_vars', 'mod_LV_covars', & 'mod_LV_means'.

The possible entries for 'coefs' are: 'raw', 'std', 'phi', & 'LV_means'

See the Details section below for information on each element.

`verbose` (optional) Should detailed results be displayed in console?
TRUE (default) or FALSE

Details

This function assesses the equivalence of the measurement and structural parameters of a confirmatory factor analysis (CFA) model, or of an exploratory structural equation model (ESEM), across multiple groups. It is a wrapper function that uses functions from the lavaan (Rosseel, 2012) and semTools (Jorgensen et al., 2026) packages.

The measurement model pertains to the measurement characteristics of the indicators (observed measures) and thus consists of the factor loadings, intercepts, and residual variances. The evaluations of the equivalence of these parameters across groups are tests of measurement invariance (Brown, 2015).

Tests of the structural parameters involve evaluations of the latent variables themselves and focus on the factor variances, covariances, and latent means. These parameters are (estimated) features of the population from which a sample was drawn. The evaluations of the equivalence of these parameters across groups are tests of population heterogeneity: Does the dispersion, the interrelationships, and the levels of the factors vary across groups?

Brown (2015) stated that it is more prudent for model evaluation to work upward from the least restricted solution (equal form) to determine if further tests of measurement invariance and population heterogeneity are warranted. Here is the recommended sequence of multiple groups invariance evaluations that is used in the present function, with the function model names in bold:

- **Configural** invariance, or "equal form" invariance, tests whether the number of factors and the pattern of indicator factor loadings are identical across groups.
- **Metric** invariance, or "equal factor loadings" invariance, tests the equality of the factor loadings across groups.

- **Scalar** invariance, "equal intercepts" invariance, or "strong factorial" invariance, tests whether the indicator intercepts are equal across groups.
- **Strict** invariance, or "equal residual" invariance, tests whether the indicator residual variances are equal across groups.
- **LV_vars** tests whether the factor variances are equal across groups.
- **LV_covars** tests whether the factor covariances are equal across groups.
- **LV_means** tests whether the latent variable means are equal across groups.

Value

A list containing the following components:

`lavaan_model_syntax`
: The lavaan CFA model syntax

`fits_by_group`
: The model fit coefficients for each group separately

`mod_Configural`
: The lavaan cfa output object for the configural invariance model

`mod_Metric`
: The lavaan cfa output object for the metric invariance model

`mod_Scalar`
: The lavaan cfa output object for the scalar invariance model

`mod_Strict`
: The lavaan cfa output object for the strict invariance model

`mod_LV_vars`
: The lavaan cfa output object for the model that restricts the latent variable variances to be equal

`mod_LV_covars`
: The lavaan cfa output object for the model that restricts the latent variable covariances to be equal

`mod_LV_means`
: The lavaan cfa output object for the model that restricts the latent variable means to be equal

`model_fits`
: The invariance model fit coefficients

`model_fit_diffs`
: Differences in the invariance model fit coefficients

`chisq_diffs`
: Chi-Squared model differences tests

`LV_scores`
: A list with scores on the latent variables for each of the models in disp_options

Author(s)

Brian P. O'Connor

References

Brown, T. A. (2015). *Confirmatory Factor Analysis for Applied Research (2nd ed.)*. New York: The Guilford Press.

Jorgensen, T. D., Pornprasertmanit, S., Schoemann, A. M., & Rosseel, Y. (2026). semTools: Useful tools for structural equation modeling. R package version 0.5-8. Retrieved from https://CRAN.R-project.org/package=semTools

Milfont, T. L., & Fischer, R. (2015). Testing measurement invariance across groups: Applications in cross-cultural research. *International Journal of Psychological Research, 3,* 111130.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48,* 136.

Stark, S., Chernyshenko, O. S., & Drasgow, F. (2006). Detecting differential item functioning with confirmatory factor analysis and item response theory: Toward a unified strategy. *Journal of Applied Psychology, 91(6),* 12921306.

Stein, J. A., Lee, J. W., & Jones, P. S. (2006). Assessing cross-cultural differences through use of multiple-group invariance analyses. *Journal of Personality Assessment, 87(3),* 249258.

Tan, T. (2024) Frequentist and Bayesian factorial invariance using R. *Practical Assessment, Research, and Evaluation. 29(8),* 1-34.

Examples

```
model_RSE <- '
   pos_items =~ Q1 + Q2 + Q4 + Q6 + Q7
   neg_items =~ Q3_R + Q5_R + Q8_R + Q9_R + Q10_R '

# using LV_model
Factorial_Invariance(data = data_RSE_sex, group = 'gender', LV_model = model_RSE)

# using LV_keys
Factorial_Invariance(data = data_RSE_sex, group = 'gender',
                     LV_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2),
                     LV_names = c('pos', 'neg'))

# another way of using LV_keys
Factorial_Invariance(data = data_RSE_sex, group = 'gender',
                     LV_keys = c( Q1 = 1, Q2 = 1, Q3_R = 2, Q4 = 1,   Q5_R = 2,
                                  Q6 = 1, Q7 = 1, Q8_R = 2, Q9_R = 2, Q10_R = 2) )

# for an ESEM model
# first, use the ESEM function to obtain the esem_model_syntax (without the group variable, gender)
esem_output <- ESEM(data = subset(data_RSE_sex, select = -c(gender)), Nfactors = 2)

Factorial_Invariance(data = data_RSE_sex, group = 'gender',
                     LV_model = esem_output$esem_model_syntax)


# See this web page for additional Examples:

# https://oconnor-psych.ok.ubc.ca/vignettes/EXAMPLES_vignettes.html
```

INTERNAL_CONSISTENCY *Internal consistency reliability coefficients*

Description

Internal consistency reliability coefficients

Usage

```
INTERNAL_CONSISTENCY(data, extraction = 'minres', reverse_these = NULL,
                         auto_reverse = TRUE, verbose=TRUE, factormodel)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables.
extraction	(optional) The factor extraction method to be used in the omega computations. The options are: 'ML' for maximum likelihood (the default); and 'PAF' for principal axis / common factor analysis.
reverse_these	(optional) A vector of the names of items that should be reverse-coded
auto_reverse	(optional) Should reverse-coding of items be conducted when warranted? TRUE (default) or FALSE
verbose	(optional) Should detailed results be displayed in console? TRUE (default) or FALSE
factormodel	(Deprecated.) Use 'extraction' instead.

Details

When 'auto_reverse = TRUE', the item loadings on the first principal component are computed and items with negative loadings are reverse-coded.

If error messages are produced, try using 'auto_reverse = FALSE'.

If item names are provided for the 'reverse_these' argument, then auto_reverse is not conducted.

The Omega-total coefficient in the output is McDonald's omega total coefficient computed using three factors (one general + 2 group factors), minres extraction, and oblimin rotation. See the OMEGA or the BIFACTOR functions in this package for alternative omega options.

See this web page for more detailed descriptions of the alpha, omega, and other coefficients produced by this function:

https://oconnor-psych.ok.ubc.ca/vignettes/Coefficient_descriptions_vignettes.html

Value

A list with the following elements:

`int.consist_scale`
: A vector with the scale omega, Cronbach's alpha, standardized Cronbach's alpha, the mean of the off-diagonal correlations, the median of the off-diagonal correlations, and the rmsr fit coefficient for a 1-factor model

`int.consist_dropped`
: A matrix of the int.consist_scale values for when each item, in turn, is int.consist_dropped from the analyses

`item_stats`
: The item means, standard deviations, and item-total correlations

`resp_opt_freqs`
: The response option frequencies

`resp_opt_props`
: The response option proportions

`new_data`
: The data that was used for the analyses, including any item reverse-codings

Author(s)

Brian P. O'Connor

References

Flora, D. B. (2020). Your coefficient alpha is probably wrong, but which coefficient omega is right? A tutorial on using R to obtain better reliability estimates. *Advances in Methods and Practices in Psychological Science, 3(4),* 484501.

McNeish, D. (2018). Thanks coefficient alpha, we'll take it from here. *Psychological Methods, 23(3),* 412433.

Revelle, W., & Condon, D. M. (2019). Reliability from alpha to omega: A tutorial. *Psychological Assessment, 31(12),* 13951411.

Examples

```
# Rosenberg Self-Esteem scale items -- without reverse-coding
INTERNAL_CONSISTENCY(data_RSE_not_recoded, extraction = 'minres',
                     reverse_these = NULL, auto_reverse = FALSE, verbose=TRUE)

# Rosenberg Self-Esteem scale items -- with auto_reverse-coding
INTERNAL_CONSISTENCY(data_RSE_not_recoded, extraction = 'minres',
                     reverse_these = NULL, auto_reverse = TRUE, verbose=TRUE)

# Rosenberg Self-Esteem scale items -- another way of reverse-coding
INTERNAL_CONSISTENCY(data_RSE_not_recoded, extraction = 'minres',
                     reverse_these = c('Q1','Q2','Q4','Q6','Q7'), verbose=TRUE)
```

LOCALDEP *Local dependence*

Description

Provides the residual correlations after partialling latent trait scores out of an inter-item correlation matrix, along with local dependence statistics.

Usage

```
LOCALDEP(data, corkind, item_type, thetas, theta_type, verbose)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables.
corkind	The kind of correlation matrix to be used for the analyses. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
item_type	(optional) The type of items for the IRT analyses. If item_type is not specified, then it is assumed that the items follow a graded or 2PL model. The options for item_type are those that can be used in the mirt function from the mirt package, which include 'Rasch', '2PL', '3PL', '3PLu', '4PL', 'graded', 'grsm', 'grsmIRT', 'gpcm', 'gpcmIRT', 'rsm', 'nominal', 'ideal', item_type 'ggum', among other possibilities.
thetas	(optional) A vector of the latent trait scores that will be partialled out of the item correlations and used in computing other local dependence statistics. If thetas are not supplied, then they will be estimated internally using the fscores function from the mirt package.
theta_type	(optional) If thetas are not supplied, then they will be estimated internally using the fscores function from the mirt package with the following options: "EAP" (default), "MAP", "ML", "WLE", "EAPsum", "plausible", and "classify".
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

Item response theory models are based on the assumption that the items display local independence. The latent trait is presumed to be responsible for the associations between the items. Once the latent trait is partialled out, the residual correlations between pairs of items should be negligible. Local dependence exists when there is additional systematic covariance among the items. It can occur when pairs of items have highly similar content or between sequentially presented items in a test. Local dependence distorts IRT parameter estimates, it can artificially increase scale information, and it distorts the latent trait, which becomes too heavily defined by the locally dependent items. Examining the residual (partial) correlations is a preliminary, exploratory method of determining whether local dependence exists. The function also displays the local dependence Q3 statistic values described by Yen (1984), the X2 statistic values described by Chen and Thissen (1997), the G2

statistic values described by Chen and Thissen (1997), and the jack-knife statistic values described by Edwards et al. (2018). The Q3, X2, G2, and jack-knife statistic values are obtained using the mirt function from the mirt package (Chalmers, 2012).

Value

A list with the following elements:

`correlations`	The correlation matrix
`residcor`	The residualized (partial) correlation matrix
`eigenvalues`	The eigenvalues
`resid_Q3`	A matrix with the Q3 statistic values described by Yen (1984)
`resid_LD`	A matrix with the X2 statistic values described by Chen and Thissen (1997)
`resid_LDG2`	A matrix with the G2 statistic values described by Chen and Thissen (1997)
`resid_JSI`	A matrix with the jack-knife statistic values described by Edwards et al. (2018)
`localdep_stats`	All of the above local dependence statistics in long format

Author(s)

Brian P. O'Connor

References

Chalmers, R., P. (2012). mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software, 48(6)*, 1-29.

Chen, W. H. & Thissen, D. (1997). Local dependence indices for item pairs using item response theory. *Journal of Educational and Behavioral Statistics, 22*, 265-289.

Edwards, M. C., Houts, C. R. & Cai, L. (2018). A diagnostic procedure to detect departures from local independence in item response theory models. *Psychological Methods, 23*, 138-149.

Yen, W. (1984). Effects of local item dependence on the fit and equating performance of the three parameter logistic model. *Applied Psychological Measurement, 8*, 125-145.

Examples

```
# Rosenberg Self-Esteem scale items
LOCALDEP(data_RSE)
```

MAP *Velicer's minimum average partial (MAP) test*

Description

Velicer's minimum average partial (MAP) test for determining the number of components, which focuses on the common variance in a correlation matrix.

Usage

```
MAP(data, corkind, Ncases, verbose)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases. Required only if data is a correlation matrix.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

This method for determining the number of components focuses on the common variance in a correlation matrix. It involves a complete principal components analysis followed by the examination of a series of matrices of partial correlations. Specifically, on the first step, the first principal component is partialled out of the correlations between the variables of interest, and the average squared coefficient in the off-diagonals of the resulting partial correlation matrix is computed. On the second step, the first two principal components are partialled out of the original correlation matrix and the average squared partial correlation is again computed. These computations are conducted for k (the number of variables) minus one steps. The average squared partial correlations from these steps are then lined up, and the number of components is determined by the step number in the analyses that resulted in the lowest average squared partial correlation. The average squared coefficient in the original correlation matrix is also computed, and if this coefficient happens to be lower than the lowest average squared partial correlation, then no components should be extractioned from the correlation matrix. Statistically, components are retained as long as the variance in the correlation matrix represents systematic variance. Components are no longer retained when there is proportionately more unsystematic variance than systematic variance (see O'Connor, 2000, p. 397).

Value

A list with the following elements:

totvarexplNOROT
: The eigenvalues and total variance explained

avgsqrs
: Velicers average squared correlations

NfactorsMAP
: The number of components according to the original (1976) MAP test

NfactorsMAP4
: The number of components according to the revised (2000) MAP test

Author(s)

Brian P. O'Connor

References

O'Connor, B. P. (2000). SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test. *Behavior Research Methods, Instrumentation, and Computers, 32,* 396-402.

Velicer, W. F. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika, 41,* 321-327.

Velicer, W. F., Eaton, C. A., and Fava, J. L. (2000). Construct explication through factor or component analysis: A review and evaluation of alternative procedures for determining the number of factors or components. In R. D. Goffin & E. Helmes, eds., *Problems and solutions in human assessment (p.p. 41-71).* Boston: Kluwer.

Examples

```
# the Harman (1967) correlation matrix
MAP(data_Harman, corkind='pearson', Ncases = 305, verbose=TRUE)

# Rosenberg Self-Esteem scale items, using Pearson correlations
MAP(data_RSE, corkind='pearson', verbose=TRUE)

# Rosenberg Self-Esteem scale items, using polychoric correlations
MAP(data_RSE, corkind='polychoric', verbose=TRUE)

# NEO-PI-R scales
MAP(data_NEOPIR, verbose=TRUE)
```

MISSING_INFO *Missing value statistics*

Description

Frequencies and proportions of missing values

Usage

```
MISSING_INFO(data, verbose)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables.
verbose	(optional) Should detailed results be displayed in console? TRUE (default) or FALSE

Details

Provides the number of cases with each of N missing values (NA values), along with the proportions, cumulative proportions, and the cumulative Ns.

Value

A matrix with the following columns:

N_cases	The number of cases
N_missing	The number of missing values
Proportion	The proportion of missing values
Cum_Proportion	The cumulative proportion of missing values
Cum_N	The cumulative number of cases

Author(s)

Brian P. O'Connor

Examples

```
MISSING_INFO(airquality)

# add NA values to the Rosenberg Self-Esteem scale items, for illustration
data_RSE_missing <- data_RSE
data_RSE_missing[matrix(rbinom(prod(dim(data_RSE_missing)), size=1, prob=.3)==1,
                 nrow=dim(data_RSE_missing)[1])] <- NA

MISSING_INFO(data_RSE_missing)
```

NEVALSGT1 *The number of eigenvalues greater than 1*

Description

Returns the count of the number of eigenvalues greater than 1 in a correlation matrix. This value is often referred to as the "Kaiser", "Kaiser-Guttman", or "Guttman-Kaiser" rule for determining the number of components or factors in a correlation matrix.

Usage

```
NEVALSGT1(data, corkind, Ncases, verbose=TRUE)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases. Required only if data is a correlation matrix.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

The rationale for this traditional procedure for determining the number of components or factors is that a component with an eigenvalue of 1 accounts for as much variance as a single variable. Extracting components with eigenvalues of 1 or less than 1 would defeat the usual purpose of component and factor analyses. Furthermore, the reliability of a component will always be nonnegative when its eigenvalue is greater than 1. This rule is the default retention criteria in SPSS and SAS.

There are a number of problems with this rule of thumb. Monte Carlo investigations have found that its accuracy rate is not acceptably high (Zwick & Velicer, 1986)). The rule was originally intended to be an upper bound for the number of components to be retained, but it is most often used as the criterion to determine the exact number of components or factors. Guttman's original proof applies only to the population correlation matrix and the sampling error that occurs in specific samples results in the rule often overestimating the number of components. The rule is also considered overly mechanical, e.g., a component with an eigenvalue of 1.01 achieves factor status whereas a component with an eigenvalue of .999 does not.

This function is included in this package for curiosity and research purposes.

Value

A list with the following elements:

NfactorsNEVALSGT1
: The number of eigenvalues greater than 1.

totvarexplNOROT
: The eigenvalues and total variance explained

Author(s)

Brian P. O'Connor

References

Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., & Strahan, E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. *Psychological Methods, 4,* 272-299.

Guttman, L. (1954). Some necessary conditions for common factor analysis. *Psychometrika, 19,* 149-161.

Hayton, J. C., Allen, D. G., Scarpello, V. (2004). Factor retention decisions in exploratory factor analysis: A tutorial on parallel analysis. *Organizational Research Methods, 7,* 191-205.

Kaiser, H. F. (1960). The application of electronic computer to factor analysis. *Educational and Psychological Measurement, 20,* 141-151.

Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychological Bulletin, 99,* 432-442.

Examples

```
# the Harman (1967) correlation matrix
NEVALSGT1(data_Harman, corkind='pearson', Ncases = 305, verbose=TRUE)

# Rosenberg Self-Esteem scale items, using Pearson correlations
NEVALSGT1(data_RSE, corkind='pearson', verbose=TRUE)

# Rosenberg Self-Esteem scale items, using polychoric correlations
NEVALSGT1(data_RSE, corkind='polychoric', verbose=TRUE)

# NEO-PI-R scales
NEVALSGT1(data_NEOPIR, corkind='pearson', verbose=TRUE)
```

OMEGA *Omega internal consistency reliability coefficients*

Description

Total and hierarchical omega internal consistency reliability coefficients computed using multiple possible methods

Usage

```
OMEGA(data, corkind = 'pearson',
      Nfactors = 4,
      bifactor_kind = c('SL', 'SLiD', 'DSL', 'bifactorT', 'bigeominT'),
      EFA_options = list(extraction = 'minres', rotation = 'oblimin'),
      LV_options = list(group_keys = NULL,
                        estimator = 'MLR',
                        rotation = 'bigeomin',
```

```
                    resid_correls = NULL,
                    LV_names = NULL,
                    ordered = FALSE),
      schmid_options = list(extraction = 'minres', rotation = 'oblimin'),
      GPA_options = list(delta = .01,
                     epsilon = .00001,
                     normalize = FALSE,
                     maxit = 1000,
                     randomStarts = 50),
      min_loading = .2, display = 2)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'.
Nfactors	(optional) The number of factors to extract, including the general factor. Nfactors is = 1 (for the general factor) + the number of group factors.
bifactor_kind	The bifactor method(s) to be used. The options are 'SL', 'SLiD', 'DSL', 'bifactorQ', 'bifactorT', 'bigeominQ', 'bigeominT', 'CFA', and 'ESEM'. Multiple methods can be specified.
EFA_options	(optional) A list with EFA options when rawdata is provided. The list elements include values for 'extraction' and 'rotation'. The possibilities for extraction are: 'minres'(the default), 'alpha', 'fullinfo', 'gls', 'image', 'ml', 'ols', 'paf', 'uls', and 'wls'. The possibilities for rotation are: 'varimax', 'bentlerT', 'entropy', 'equamax', 'geominT', 'quartimax', 'promax', 'bentlerQ', 'geominQ', 'oblimin' (the default, 'oblimax', 'quartimin', 'simplimax', and 'none'.
LV_options	(optional) A list with cfa options when bifactor_kind = 'CFA'. The possible list elements are 'group_keys', 'estimator', 'rotation', 'resid_correls', 'LV_names', and 'ordered'. See the Details section below for information on each element.
schmid_options	(optional) A list with schmid function (from the psych package) options when bifactor_kind is one of 'SL', 'SLiD', or 'DSL'. The list elements include values for 'extraction' and 'rotation'. The possibilities for extraction are: 'minres'(the default), 'paf', 'pc', and 'ml'. The possibilities for rotation are: 'oblimin' (the default), 'simplimax', 'Promax', 'promax' and 'none'.
GPA_options	(optional) A list with options for when a GPArotation is used. See the Details section below.

`min_loading` The minimum value of a group factor loading for an item to be considered to have a non-negligible contribution to a group factor. min_loading only plays a role in the computations for PUC and for some group factor statistics.

`display` The results to be displayed in the console:
0 = nothing
1 = only the omega and scale coefficients
2 (default) = detailed output, including the bifactor model loadings and statistics

Details

For the **bifactor_kind** argument:

- **bifactorT** is an orthogonal bifactor rotation designed for situations where a single, overarching global dimension is expected alongside separate sub-domains, and when all factors should be uncorrelated.
- **bifactorQ** is an oblique bifactor rotation designed for when a strong, overarching global dimension is expected alongside separate sub-domains that are allowed to correlate with each other.
- **bigeominT** is an orthogonal bifactor rotation designed for situations where a single, overarching global dimension is expected alongside separate sub-domains, and when all factors should be uncorrelated.
- **bigeominQ** is an oblique bifactor rotation designed for when a strong, overarching global dimension is expected alongside separate sub-domains that are allowed to correlate with each other.
- **SL** is an oblique bifactor rotation designed for where there is a broad, overarching factor alongside sub-domains that are allowed to overlap.
- **SLiD** an orthogonal bifactor rotation method designed for exploratory bifactor analysis that tries to ensure that once the general factor's variance is pulled out, each item loads onto only one specific group factor.
- **DSL** entropy is an orthogonal factor rotation that pushes factor loadings to be either strongly dominant (close to 1.0) or cleanly absent (close to 0.0), reducing the overall informational "noise" of the matrix.
- **CFA** confirmatory bifactor analyses are conducted using the cfa function from the lavaan package. All items are assumed to load on the general factor. The items that load on the group factors are indicated using "group_keys" for the LV_options argument.
- **ESEM** rxploratory structural equation modeling bifactor analyses conducted using the cfa function from the lavaan package. All items are assumed to load on the general factor. The items that load on the group factors can be indicated using "group_keys" for the LV_options argument.

For the **LV_options** argument:

When bifactor_kind = 'CFA' or 'ESEM', then LV_options is a list with information for these analyses. The possible list elements are 'group_keys', 'estimator', 'rotation', 'resid_correls', 'LV_names', and 'ordered'.

- **group_keys** is a vector indicating which item (i.e., which variable in data) goes with which group factor (which can be named or numbered).
  Example: group_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2)
- **estimator** is the name of the lavaan estimator to be used in the analyses. The options for basic estimators (continuous data) are ML, GLS, WLS, DWLS, ULS, DLS, and PML. The options for robust estimators are MLM, MLMVS, MLMV, MLF, MLR, WLSM, WLSMVS, WLSMV, ULSM, ULSMVS, and ULSMV.
  Example: estimator = 'ML'
- **rotation** is the kind of lavaan bifactor rotation. The two options are 'bigeomin' or 'biquartimin'.
- **resid_correls** is a vector with the pairs of correlated error terms (if any).
  Example: resid_correls = c('Q1 ~~ Q2', 'Q9 ~~ Q10')
- **LV_names** is a vector of names for the latent variables
- **ordered** are the variables in rawdata (if provided) ordered? For example, Likert scale responses are usually ordered. The function will produce notices when the data are not consistent with the "ordered" status.

Examples of LV_options:

```
LV_options = list(group_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2),
                  resid_correls = NULL, estimator = 'WLSMVS')


LV_options = list(group_keys = c(Q1=1, Q2=1, Q3_R=2, Q4=1,   Q5_R=2,
                                 Q6=1, Q7=1, Q8_R=2, Q9_R=2, Q10_R=2)),
                  resid_correls = c('Q1 ~~ Q2', 'Q9 ~~ Q10'),
                  estimator = 'WLSMVS')
```

For the **GPA_options** argument:

A list with options for when a GPArotation is used. The list elements are 'delta', 'epsilon', 'normalize', 'maxit', and 'randomStarts'.

- **delta** is a small positive number that is added to the squared factor loadings in the (geomin) optimization denominator to prevent division by zero and control the sharpness of the penalty near zero. (Note: In MPLus, delta is called "epsilon".)
- **epsilon** is the numeric rotational convergence tolerance threshold. It controls how precise or strict the stopping rule is for the rotation algorithm. The gradient projection algorithm stops optimizing and assumes the rotation has converged when the norm of the gradient falls below this value.
- **normalize** specifies whether and how the loadings matrix is normalized prior to rotation and denormalized after rotation. The possibilities are FALSE (the default) for no normalization, or TRUE, for the Kaiser/Horst normalization.
- **maxit** is the maximum number of iterations.

- **randomStarts** is an integer that specifies the number of random starting configurations used to optimize the factor rotation. It helps avoid getting trapped in local minima during factor rotation optimization. It runs the gradient projection algorithm multiple timeseach using a different random orthogonal or oblique starting rotation matrix (Tmat) generated via QR decomposition. It returns the rotated factor loadings matrix and results corresponding to the run with the lowest objective criterion value among all attempts.

See this web page for more detailed descriptions of the omega-total and omega-hierarchical internal consistency reliability coefficients and other statistics produced by this function:

https://oconnor-psych.ok.ubc.ca/vignettes/Coefficient_descriptions_vignettes.html

Value

A list with the omega coefficients, the factor loadings, and model fit statistics.

Author(s)

Brian P. O'Connor

References

Bernaards, C. A., & Jennrich, R. I. (2005). Gradient Projection Algorithms and Software for Arbitrary Rotation Criteria in Factor Analysis. *Educational and Psychological Measurement, 65(5),* 676-696.

Bernaards, C. A., & Jennrich, R. I. (2026). GPArotation: Gradient Projection Factor Rotation. R package version 2026.4-1, https://CRAN.R-project.org/package=GPArotation

Educational Content Team. (2026). McDonald's Omega. Cogn-IQ Encyclopedia. https://pubscience.org/cqep.2025.0048

Flora, D. B. (2020). Your coefficient alpha is probably wrong, but which coefficient omega is right? A tutorial on using R to obtain better reliability estimates. *Advances in Methods and Practices in Psychological Science, 3(4),* 484501.

Garcia-Garzon, E., Abad, F. J., & Garrido, L. E. (2021). On omega hierarchical estimation: A Comparison of Exploratory Bi-Factor Analysis Algorithms. *Multivariate Behavioral Research, 56(1),* 101-119.

Jennrich, R. I. (2018). Rotation. In P. Irwing, T. Booth, & D. J. Hughes (Eds.), *The Wiley handbook of psychometric testing: A multidisciplinary reference on survey, scale and test development* (pp. 279304). Wiley Blackwell. https://doi.org/10.1002/9781118489772.ch10

Kalkbrenner, M. T. (2024). Choosing Between Cronbachs Coefficient Alpha, McDonalds Coefficient Omega, and Coefficient H: Confidence Intervals and the Advantages and Drawbacks of Interpretive Guidelines. *Measurement and Evaluation in Counseling and Development, 57(2),* 93105.

McNeish, D. (2018). Thanks coefficient alpha, we'll take it from here. *Psychological Methods, 23(3),* 412433.

Revelle, W. (2026). psych: Procedures for Psychological, Psychometric, and Personality Research. R package version 2.6.5, https://CRAN.R-project.org/package=psych

Revelle, W., & Condon, D. M. (2019). Reliability from alpha to omega: A tutorial. *Psychological Assessment, 31(12),* 13951411.

Waller, N. G. (2018) Direct Schmid-Leiman Transformations and Rank-Deficient Loadings Matrices. *Psychometrika, 83(4),* 858870.

Examples

```
OMEGA(data_RSE, Nfactors = 3, display = 1)
```

PARALLEL *Parallel analysis of eigenvalues (random data only)*

Description

Generates eigenvalues and corresponding percentile values for random data sets with specified numbers of variables and cases.

Usage

```
PARALLEL(Nvars=20, Ncases=300, Ndatasets=100, extraction='PCA', percentile='95',
         corkind='pearson', verbose=TRUE, factormodel)
```

Arguments

Nvars	The number of variables.
Ncases	The number of cases.
Ndatasets	An integer indicating the # of random data sets for parallel analyses.
extraction	The factor extraction method. The options are: 'PAF' for principal axis / common factor analysis; 'PCA' for principal components analysis. 'image' for image analysis.
percentile	An integer indicating the percentile from the distribution of parallel analysis random eigenvalues. Suggested value: 95
corkind	The kind of correlation matrix to be used for the random data. The options are 'pearson', 'kendall', and 'spearman'.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE
factormodel	(Deprecated.) Use 'extraction' instead.

Details

This procedure for determining the number of components or factors involves comparing the eigenvalues derived from an actual data set to the eigenvalues derived from the random data. In Horn's original description of this procedure, the mean eigenvalues from the random data served as the comparison baseline, whereas the more common current practice is to use the eigenvalues that correspond to the desired percentile (typically the 95th) of the distribution of random data eigenvalues. Factors or components are retained as long as the ith eigenvalue from the actual data is greater than the ith eigenvalue from the random data. This function produces only random data eigenvalues and it does not take real data as input. See the RAWPAR function in this package for parallel analyses that also involve real data.

The PARALLEL function permits users to specify PCA or PAF or image as the factor extraction method. Principal components eigenvalues are often used to determine the number of common factors. This is the default in most statistical software packages, and it is the primary practice in the literature. It is also the method used by many factor analysis experts, including Cattell, who often examined principal components eigenvalues in his scree plots to determine the number of common factors. Principal components eigenvalues are based on all of the variance in correlation matrices, including both the variance that is shared among variables and the variances that are unique to the variables. In contrast, principal axis eigenvalues are based solely on the shared variance among the variables. The procedures are qualitatively different. Some therefore claim that the eigenvalues from one extraction method should not be used to determine the number of factors for another extraction method. The PAF option in the extraction argument for the PARALLEL function was included solely for research purposes. It is best to use PCA as the extraction method for regular data analyses.

Value

Random data eigenvalues

Author(s)

Brian P. O'Connor

References

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika, 30,* 179-185.

O'Connor, B. P. (2000). SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test. *Behavior Research Methods, Instrumentation, and Computers, 32,* 396-402.

Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychological Bulletin, 99,* 432-442.

Examples

```
PARALLEL(Nvars=15, Ncases=250, Ndatasets=100, extraction='PCA', percentile=95,
         corkind='pearson', verbose=TRUE)
```

PCA *Principal components analysis*

Description

Principal components analysis

Usage

```
PCA(data, Nfactors=NULL, rotation='promax', corkind='pearson',
          Ncases=NULL, ppower = 3,
          GPA_options = list(delta = .01,
                             epsilon = .00001,
                             normalize = FALSE,
                             maxit = 1000,
                             randomStarts = 50),
          verbose=TRUE, rotate)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal.The function internally determines whether the data are a correlation matrix.
Nfactors	The number of components to extraction. If not specified, then the EMPKC procedure will be used to determine the number of components.
rotation	The factor rotation method for the analysis. The orthogonal rotation options are: 'varimax' (the default), 'quartimax', 'bentlerT', 'equamax', 'geominT', 'entropy', and 'none'. The oblique rotation options are: 'promax' (the default), 'quartimin', 'oblimin', 'oblimax', 'simplimax', 'bentlerQ', 'geominQ', and 'none'.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases. Required only if data is a correlation matrix.
ppower	The power value to be used in a promax rotation (required only if rotation = 'promax'). Suggested value: 3
GPA_options	(optional) A list with options for when a GPArotation is used. See the Details section below.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE
rotate	(Deprecated.) Use 'rotation' instead.

Details

The factor rotation computations for the following methods are conducted using the GPArotation package (Bernaards & Jennrich, 2005, 2026): 'bentlerQ', 'bentlerT', 'entropy', 'geominQ', 'geominT', 'oblimax', 'oblimin', 'quartimax', 'quartimin', and 'simplimax'.

For factor rotation (see Jennrich, 2018, for a review):

- **bentlerQ** is an oblique rotation based on Bentler's invariant pattern simplicity criterion.
- **bentlerT** is an orthogonal rotation based on Bentler's invariant pattern simplicity criterion
- **bifactorQ** is an oblique bifactor rotation designed for when a strong, overarching global dimension is expected alongside separate sub-domains that are allowed to correlate with each other.
- **bifactorT** is an orthogonal bifactor rotation designed for situations where a single, overarching global dimension is expected alongside separate sub-domains, and when all factors should be uncorrelated.
- **bigeominQ** is an oblique bifactor rotation designed for where there is a broad, overarching factor alongside sub-domains that are allowed to overlap.
- **bigeominT** an orthogonal bifactor rotation method designed for exploratory bifactor analysis that tries to ensure that once the general factor's variance is pulled out, each item loads onto only one specific group factor.
- **entropy** entropy is an orthogonal factor rotation that pushes factor loadings to be either strongly dominant (close to 1.0) or cleanly absent (close to 0.0), reducing the overall informational "noise" of the matrix.
- **equamax** is an orthogonal rotation designed as a mathematical compromise between varimax and quartimax.
- **geominQ** is an oblique factor rotation method that find a clean "simple structure" even when the data contains highly complex variables (variables that naturally load on multiple factors).
- **geominT** is an orthogonal factor rotation that combines the strict geometric constraint of uncorrelated factors with Brownes flexible, product-based geomin complexity criterion.
- **oblimax** is an oblique factor rotation method that maximizes the number of very high and very low (near-zero) factor loadings.
- **oblimin** is an oblique factor rotation that allows factors to correlate freely to achieve the simplest possible loading structure.
- **promax** is a two-stage oblique factor rotation method.
- **quartimax** is an orthogonal factor rotation designed to spread the explained variance evenly across the factors, actively preventing a single general factor from dominating.
- **simplimax** is an oblique factor rotation that attempts to recover complex or messy "simple structures" where traditional continuous methods like oblimin or geomin fail.
- **quartimin** is an oblique factor rotation that focuses on simplifying the rows of the loading matrix.
- **varimax** is an orthogonal factor rotation that maximizes the variance of the squared factor loadings within each individual factor column.

For **GPA_options:**

A list with options for when a GPArotation is used. The list elements are 'delta', 'epsilon', 'normalize', 'maxit', and 'randomStarts'.

- **delta** is a small positive number that is added to the squared factor loadings in the (geomin) optimization denominator to prevent division by zero and control the sharpness of the penalty near zero. (Note: In MPLus, delta is called "epsilon".)
- **epsilon** is the numeric rotational convergence tolerance threshold. It controls how precise or strict the stopping rule is for the rotation algorithm. The gradient projection algorithm stops optimizing and assumes the rotation has converged when the norm of the gradient falls below this value.
- **normalize** specifies whether and how the loadings matrix is normalized prior to rotation and denormalized after rotation. The possibilities are FALSE (the default) for no normalization, or TRUE, for the Kaiser/Horst normalization.
- **maxit** is the maximum number of iterations.
- **randomStarts** is an integer that specifies the number of random starting configurations used to optimize the factor rotation. It helps avoid getting trapped in local minima during factor rotation optimization. It runs the gradient projection algorithm multiple timeseach using a different random orthogonal or oblique starting rotation matrix (Tmat) generated via QR decomposition. It returns the rotated factor loadings matrix and results corresponding to the run with the lowest objective criterion value among all attempts.

See this web page for more detailed descriptions of the above rotation methods:

https://oconnor-psych.ok.ubc.ca/vignettes/EFA_BIFACTOR_vignettes.html

Value

A list with the following elements:

`loadingsNOROT`	The unrotated factor loadings
`loadingsROT`	The rotated factor loadings
`pattern`	The pattern matrix
`structure`	The structure matrix
`phi`	The correlations between the factors
`varexplNOROT1`	The initial eigenvalues and total variance explained
`varexplROT`	The rotation sums of squared loadings and total variance explained for the rotated loadings
`cormat_reprod`	The reproduced correlation matrix, based on the rotated loadings
`fit_coeffs`	Model fit coefficients
`communalities`	The unrotated factor solution communalities
`uniquenesses`	The unrotated factor solution uniquenesses

Author(s)

Brian P. O'Connor

Examples

```
# the Harman (1967) correlation matrix
PCA(data_Harman, Nfactors=2, Ncases=305, rotation='oblimin')

# Rosenberg Self-Esteem scale items
PCA(data_RSE, corkind='polychoric', Nfactors=2)

# NEO-PI-R scales
PCA(data_NEOPIR, Nfactors=5, rotation='promax')
```

PLOT_Invariance *Plots of multiple groups factor analysis invariance statistics*

Description

Plots factor loadings, slopes and intercepts, and/or latent variable distributions from multiple groups confirmatory factor analysis (CFA) models and exploratory structural equation models (ESEM)

Usage

```
PLOT_Invariance(model_object,
                invar_model = 'Scalar',
                plot_types = c('loadings', 'ints_slopes', 'LV_distribs'),
                loadings_std = TRUE, loadings_by_factor = FALSE)
```

Arguments

model_object	A "Factorial_Invariance" function output object, or a lavaan multiple groups cfa output object.
invar_model	The kind of invariance model for the plots when model_object is from the Factorial_Invariance function. The options are: 'Configural', 'Metric', 'Scalar', 'Strict', 'LV_vars', 'LV_covars', and 'LV_means'. (This argument can be ignored when model_object is not from the Factorial_Invariance function.)
plot_types	The desired kind(s) of plots. The possibilities are (1) 'loadings', for factor loadings, (2) 'ints_slopes', for intercepts and slopes, and/or (3) 'LV_distribs', for latent variable distributions.
loadings_std	Should standardized loadings be used for 'loadings' plots? TRUE or FALSE
loadings_by_factor	For 'loadings' plots, should a separate plot be produced for each factor (TRUE)? If FALSE, then the loadings for different factors will appear in the same plots.

Value

A list containing the following components (if plotted using this function): loadings, estimates, & LV_scores.

Author(s)

Brian P. O'Connor

Examples

```
RSE_output <-
Factorial_Invariance(data = data_RSE_sex, group = 'gender',
                     LV_keys = c(1, 1, 2, 1, 2, 1, 1, 2, 2, 2),
                     LV_names = c('pos', 'neg'))

PLOT_Invariance(model_object = RSE_output,
                invar_model = 'Configural',
                plot_types = c('loadings', 'ints_slopes'),
                loadings_std = TRUE, loadings_by_factor = TRUE)

HS_1939_output <-
  Factorial_Invariance(data = data_HS_1939, group = 'school',
                       LV_keys = c(1,1,1, 2,2,2, 3,3,3),
                       LV_names = c('visual', 'textual', 'speed'))

PLOT_Invariance(model_object = HS_1939_output,
                invar_model = 'Scalar',
                plot_types = c('LV_distribs'))
```

POLYCHORIC_R *Polychoric correlation matrix*

Description

Produces a polychoric correlation matrix

Usage

```
POLYCHORIC_R(data, method, verbose)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables. All values should be integers, as in the values for Likert rating scales.
method	(optional) The source package used to estimate the polychoric correlations: 'Revelle' for the psych package (the default); 'Fox' for the polycor package.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

Applying familiar factor analysis procedures to item-level data can produce misleading or uninterpretable results. Common factor analysis, maximum likelihood factor analysis, and principal components analysis produce meaningful results only if the data are continuous and multivariate normal. Item-level data almost never meet these requirements.

The correlation between any two items is affected by both their substantive (content-based) similarity and by the similarities of their statistical distributions. Items with similar distributions tend to correlate more strongly with one another than do with items with dissimilar distributions. Easy or commonly endorsed items tend to form factors that are distinct from difficult or less commonly endorsed items, even when all of the items measure the same unidimensional latent variable. Item-level factor analyses using traditional methods are almost guaranteed to produce at least some factors that are based solely on item distribution similarity. The items may appear multidimensional when in fact they are not. Conceptual interpretations of the nature of item-based factors will often be erroneous.

A common, expert recommendation is that factor analyses of item-level data (e.g., for binary response options or for ordered response option categories) or should be conducted on matrices of polychoric correlations. Factor analyses of polychoric correlation matrices are essentially factor analyses of the relations among latent response variables that are assumed to underlie the data and that are assumed to be continuous and normally distributed.

This is a cpu-intensive function. It is probably not necessary when there are > 8 item response categories.

By default, the function uses the polychoric function from William Revelle's' psych package to produce a full matrix of polychoric correlations. The function uses John Fox's hetcor function from the polycor package when requested or when the number of item response categories is > 8.

Value

The polychoric correlation matrix

Author(s)

Brian P. O'Connor

Examples

```
# Revelle polychoric correlation matrix for the Rosenberg Self-Esteem Scale (RSE)
POLYCHORIC_R(data_RSE, method = 'Revelle')

# Fox polychoric correlation matrix for the Rosenberg Self-Esteem Scale (RSE)
POLYCHORIC_R(data_RSE, method = 'Fox')
```

PROCRUSTES *Procrustes factor rotation*

Description

Conducts Procrustes rotations of a factor loading matrix to a target factor matrix, and it computes the factor solution congruence and the root mean square residual (based on comparisons of the entered factor loading matrix with the Procrustes-rotated matrix).

Usage

```
PROCRUSTES(loadings, target, type, verbose)
```

Arguments

loadings	The loading matrix that will be aligned with the target.
target	The target loading matrix.
type	The options are 'orthogonal' or 'oblique' rotation.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

This function conducts Procrustes rotations of a factor loading matrix to a target factor matrix, and it computes the factor solution congruence and the root mean square residual (based on comparisons of the entered factor loading matrix with the Procrustes-rotated matrix). The orthogonal Procrustes rotation is based on Schonemann (1966; see also McCrae et al., 1996). The oblique Procrustes rotation is based on Hurley and Cattell (1962). The factor solution congruence is the Tucker-Wrigley-Neuhaus factor solution congruence coefficient (see Guadagnoli & Velicer, 1991; and ten Berge, 1986, for reviews).

Value

A list with the following elements:

loadingsPROC	The Procrustes-rotated loadings
congruence	The factor solution congruence after factor Procrustes rotation
rmsr	The root mean square residual
residmat	The residual matrix after factor Procrustes rotation

Author(s)

Brian P. O'Connor

References

Guadagnoli, E., & Velicer, W. (1991). A comparison of pattern matching indices. *Multivariate Behavior Research, 26,* 323-343.

Hurley, J. R., & Cattell, R. B. (1962). The Procrustes program: Producing direct rotation to test a hypothesized factor structure. *Behavioral Science, 7,* 258-262.

McCrae, R. R., Zonderman, A. B., Costa, P. T. Jr., Bond, M. H., & Paunonen, S. V. (1996). Evaluating replicability of factors in the revised NEO personality inventory: Confirmatory factor analysis versus Procrustes rotation. *Journal of Personality and Social Psychology, 70,* 552-566.

Schonemann, P. H. (1966). A generalized solution of the orthogonal Procrustes problem. *Psychometrika, 31,* 1-10.

ten Berge, J. M. F. (1986). Some relationships between descriptive comparisons of components from different studies. *Multivariate Behavioral Research, 21,* 29-40.

Examples

```
# RSE data
PCAoutput_1 <- PCA(data_RSE[1:150,],   Nfactors = 2, rotation='promax', verbose=FALSE)

PCAoutput_2 <- PCA(data_RSE[151:300,], Nfactors = 2, rotation='promax', verbose=FALSE)

PROCRUSTES(target=PCAoutput_1$pattern, loadings=PCAoutput_2$pattern,
           type = 'orthogonal', verbose=TRUE)
```

RAWPAR *Parallel analysis of eigenvalues (for raw data)*

Description

Parallel analysis of eigenvalues, with real data as input, for deciding on the number of components or factors.

Usage

```
RAWPAR(data, randtype, extraction, Ndatasets, percentile,
       corkind, corkindRAND, Ncases=NULL, eval_plot=TRUE, verbose, factormodel)
```

Arguments

`data`	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.

randtype	The kind of random data to be used in the parallel analysis: 'generated' for random normal data generation; 'permuted' for permutations of the raw data matrix.
extraction	The factor extraction method. The options are: 'PAF' for principal axis / common factor analysis; 'PCA' for principal components analysis. 'image' for image analysis.
Ndatasets	An integer indicating the # of random data sets for parallel analyses.
percentile	An integer indicating the percentile from the distribution of parallel analysis random eigenvalues to be used in determining the # of factors. Suggested value: 95
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
corkindRAND	The kind of correlation matrix to be used for the random data analyses. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. The default is 'pearson'.
Ncases	The number of cases upon which a correlation matrix is based. Required only if data is a correlation matrix.
eval_plot	Should a plot of the real and parallel analysis eigenvalues be produced? The options are TRUE (the default) or FALSE.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE
factormodel	(Deprecated.) Use 'extraction' instead.

Details

The parallel analysis procedure for deciding on the number of components or factors involves extracting eigenvalues from random data sets that parallel the actual data set with regard to the number of cases and variables. For example, if the original data set consists of 305 observations for each of 8 variables, then a series of random data matrices of this size (305 by 8) would be generated, and eigenvalues would be computed for the correlation matrices for the original, real data and for each of the random data sets. The eigenvalues derived from the actual data are then compared to the eigenvalues derived from the random data. In Horn's original description of this procedure, the mean eigenvalues from the random data served as the comparison baseline, whereas the more common current practice is to use the eigenvalues that correspond to the desired percentile (typically the 95th) of the distribution of random data eigenvalues. Factors or components are retained as long as the ith eigenvalue from the actual data is greater than the ith eigenvalue from the random data.

The RAWPAR function permits users to specify PCA or PAF or image as the factor extraction method. Principal components eigenvalues are often used to determine the number of common factors. This is the default in most statistical software packages, and it is the primary practice in the literature. It is also the method used by many factor analysis experts, including Cattell, who often examined principal components eigenvalues in his scree plots to determine the number of common factors. Principal components eigenvalues are based on all of the variance in correlation matrices, including both the variance that is shared among variables and the variances that are unique to the variables. In contrast, principal axis eigenvalues are based solely on the shared variance among the variables. The procedures are qualitatively different. Some therefore claim that the eigenvalues from one extraction method should not be used to determine the number of factors for another

extraction method. The PAF option in the extraction argument for the RAWPAR and PARALLEL function was included solely for research purposes. It is best to use PCA as the extraction method for regular data analyses.

Polychoric correlations are time-consuming to compute. While polychoric correlations should probably be specified for the real data eigenvalues when data consists of item-level responses, polychoric correlations probably should not be specified for the random data computations, even for item-level data. The procedure would take much time and it is unnecessary. Polychoric correlations are estimates of what the Pearson correlations would be had the real data been continuous. For item-level data, specify polychoric correlations for the real data eigenvalues (corkind='polychoric') and use the default for the random data eigenvalues (corkindRAND='pearson'). The option for using polychoric correlations for the random data computations (corkindRAND='polychoric') was provided solely for research purposes.

Value

A list with:

`eigenvalues`	the eigenvalues for the real and random data
`NfactorsPA`	the number of factors based on the parallel analysis

Author(s)

Brian P. O'Connor

References

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika, 30,* 179-185.

O'Connor, B. P. (2000). SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test. *Behavior Research Methods, Instrumentation, and Computers, 32,* 396-402.

Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychological Bulletin, 99,* 432-442.

Examples

```
# WISC data
RAWPAR(data_TabFid, randtype='generated', extraction='PCA', Ndatasets=100,
       percentile=95, corkind='pearson', verbose=TRUE)

# the Harman (1967) correlation matrix
RAWPAR(data_Harman, randtype='generated', extraction='PCA', Ndatasets=100,
       percentile=95, corkind='pearson', Ncases=305, verbose=TRUE)

# Rosenberg Self-Esteem scale items, using Pearson correlations
RAWPAR(data_RSE, randtype='permuted', extraction='PCA', Ndatasets=100,
       percentile=95, corkind='pearson', corkindRAND='pearson', verbose=TRUE)
```

```
# Rosenberg Self-Esteem scale items, using polychoric correlations
RAWPAR(data_RSE, randtype='generated', extraction='PCA', Ndatasets=100,
       percentile=95, corkind='polychoric', verbose=TRUE)

# NEO-PI-R scales
RAWPAR(data_NEOPIR, randtype='generated', extraction='PCA', Ndatasets=100,
       percentile=95, corkind='pearson', Ncases=305, verbose=TRUE)
```

RECODE *Recode values in a vector*

Description

Options for changing the numeric values in a vector to new numeric values

Usage

```
RECODE(data, old = NULL, new = NULL, type = 'reverse', max_value = NULL,
       real_min = NULL, real_max = NULL, new_min = NULL, new_max = NULL)
```

Arguments

data	A numeric vector, typically consisting of item responses.
old	(optional) A vector of the values in data to be recoded, e.g., old = c(1,2,3,4).
new	(optional) A vector of the values that should replace the old values, e.g., new = c(0,0,1,1).
type	(optional) The type of recoding if "old" and "new" are not specified. The options are 'reverse' (for reverse coding) and 'new_range' (for changing the metric/range).
max_value	(optional) For type = 'reverse' coding only. It is the maximum possible value for data (an item). This option is included for when max(data) is not the maximum possible value for an item (e.g., when the highest response option was never used).
real_min	(optional) For type = 'new_range' coding only. The minimum possible value for data.
real_max	(optional) For type = 'new_range' coding only. The maximum possible value for data.
new_min	(optional) For type = 'new_range' coding only. The desired, new minimum possible value for data.
new_max	(optional) For type = 'new_range' coding only. The desired, new maximum possible value for data.

Details

When 'old' and 'new' are specified, the data values in the 'old' vector are replaced with the values in the same ordinal position in the 'new' vector, e.g., occurrences of the second value in 'old' in data are replaced with the second value in 'new' in data.

Regarding the type = 'new_range' option: Sometimes the items in a pool have different response option ranges, e.g., some on a 5-point scale and others on a 6-point scale. The type = 'new_range' option changes the metric/range of a specified item to a desired metric, e.g., so that scales scores based on all of the items in the pool can be computed. This alters item scores and the new item values may not be integers. Specifically, for each item response, the percent value on the real/used item is computed. Then the corresponding value on the desired new item metric for the same percentage is found.

Value

The recoded data values

Author(s)

Brian P. O'Connor

Examples

```
data <- c(1,2,3,4,1,2,3,4)
print(RECODE(data, old = c(1,2,3,4), new = c(1,1,2,2)) )

print(RECODE(data, type = 'reverse'))

# reversing coding the third item (Q3) of the Rosenberg Self-Esteem scale data
data_RSE_rev <- RECODE(data_RSE_not_recoded[,'Q3'], type = 'reverse')
table(data_RSE_rev); table(data_RSE_not_recoded[,'Q3'])

# changing the third item (Q3) responses for the Rosenberg Self-Esteem scale data
# from 0-to-4 to 1-to-5
data_RSE_rev <- RECODE(data_RSE_not_recoded[,'Q3'], old = c(0,1,2,3,4), new = c(1,2,3,4,5))
table(data_RSE_rev); table(data_RSE_not_recoded[,'Q3'])

# changing the metric/range of the third item (Q3) responses for the
# Rosenberg Self-Esteem scale data
data_RSE_rev <- RECODE(data_RSE_not_recoded[,'Q3'], type = 'new_range',
                 real_min = 1, real_max = 4, new_min = 1, new_max = 5 )
table(data_RSE_rev); table(data_RSE_not_recoded[,'Q3'])
```

ROOTFIT *Factor fit coefficients*

Description

A variety of fit coefficients for the possible N-factor solutions in exploratory factor analysis

Usage

```
ROOTFIT(data, corkind='pearson', Ncases=NULL, extraction='PAF', verbose, factormodel)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal.The function internally determines whether the data are a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases upon which a correlation matrix is based. Required only if data is a correlation matrix.
extraction	The factor extraction method. The options are: 'PAF' for principal axis / common factor analysis; 'PCA' for principal components analysis. 'ML' for maximum likelihood estimation.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE
factormodel	(Deprecated.) Use 'extraction' instead.

Details

Eigenvalue

An eigenvalue is the variance of the factor. More specifically, an eigenvalue is the the variance of the linear combination of the variables for a factor. There are as many eigenvalues for a correlation or covariance matrix as there are variables in the matrix. The sum of the eigenvalues is equal to the number of variables. An eigenvalue of one means that a factor explains as much variance as one variable.

RMSR – Root Mean Square Residual (absolute fit)

RMSR (or perhaps more commonly, RMR) is an index of the overall badness-of-fit. It is the square root of the mean of the squared residuals (the residuals being the simple differences between original correlations and the correlations implied by the N-factor model). RMSR is 0 when there is perfect model fit. A value less than .08 is generally considered a good fit. A standardized version of the RMSR is often recommended over the RMSR in structural equation modeling analyses. This is because the values in covariance matrices are scale-dependent. However, the RMSR coefficient that is provided in this package is based on correlation coefficients (not covariances) and therefore does not have this problem.

GFI (absolute fit)

The GFI (McDonald, 1999) is an index of how closely a correlation matrix is reproduced by the factor solution. It is equal to 1.0 - mean-squared residual / mean-squared correlation, ignoring the diagonals.

CAF (common part accounted for)

Lorenzo-Seva, Timmerman, & Kiers (2011): "We now propose an alternative goodness-of-fit index that can be used with any extraction method. This index expresses the extent to which the common variance in the data is captured in the common factor model. The index is denoted as CAF (common part accounted for)."

"A measure that expresses the amount of common variance in a matrix is found in the KMO (Kaiser, Meyer, Olkin) index (see Kaiser, 1970; Kaiser & Rice, 1974). The KMO index is commonly used to assess whether a particular correlation matrix R is suitable for common factor analysis (i.e., if there is enough common variance to justify a factor analysis)."

"Now, we propose to express the common part accounted for by a common factor model with q common factors as 1 minus the KMO index of the estimated residual matrix."

"The values of CAF are in the range [0, 1] and if they are close to zero it means that a substantial amount of common variance is still present in the residual matrix after the q factors have been extractioned (implying that more factors should be extractioned). Values of CAF close to one mean that the residual matrix is free of common variance after the q factors have been extractioned (i.e., no more factors should be extractioned)."

RMSEA - Root Mean Square Error of Approximation (absolute fit)

Schermelleh-Engel (2003): "The Root Mean Square Error of Approximation (RMSEA; Steiger, 1990) is a measure of approximate fit in the population and is therefore concerned with the discrepancy due to approximation. Steiger (1990) as well as Browne and Cudeck (1993) define a "close fit" as a RMSEA value <= .05. According to Browne and Cudeck (1993), RMSEA values <= .05 can be considered as a good fit, values between .05 and .08 as an adequate fit, and values between .08 and .10 as a mediocre fit, whereas values > .10 are not acceptable. Although there is general agreement that the value of RMSEA for a good model should be less than .05, Hu and Bentler (1999) suggested an RMSEA of less than .06 as a cutoff criterion."

Kenny (2020): "The measure is positively biased (i.e., tends to be too large) and the amount of the bias depends on smallness of sample size and df, primarily the latter. The RMSEA is currently the most popular measure of model fit. MacCallum, Browne and Sugawara (1996) have used 0.01, 0.05, and 0.08 to indicate excellent, good, and mediocre fit respectively. However, others have suggested 0.10 as the cutoff for poor fitting models. These are definitions for the population. That is, a given model may have a population value of 0.05 (which would not be known), but in the sample it might be greater than 0.10. There is greater sampling error for small df and low N models, especially for the former. Thus, models with small df and low N can have artificially large values of the RMSEA. For instance, a chi square of 2.098 (a value not statistically significant), with a df of 1 and N of 70 yields an RMSEA of 0.126. For this reason, Kenny, Kaniskan, and McCoach (2014) argue to not even compute the RMSEA for low df models."

Hooper (2008): "In recent years it has become regarded as "one of the most informative fit indices" (Diamantopoulos and Siguaw, 2000: 85) due to its sensitivity to the number of estimated parameters in the model. In other words, the RMSEA favours parsimony in that it will choose the model with the lesser number of parameters."

TLI – Tucker Lewis Index (incremental fit)

The Tucker-Lewis index, TLI, is also sometimes called the non-normed fit index, NNFI, or the Bentler-Bonett non-normed fit index, or RHO2. The TLI penalizes for model complexity.

Schermelleh-Engel (2003): "The (TLI or) NNFI ranges in general from zero to one, but as this index is not normed, values can sometimes leave this range, with higher (TLI or) NNFI values indimessageing better fit. A rule of thumb for this index is that .97 is indimessageive of good fit relative to the independence model, whereas values greater than .95 may be interpreted as an acceptable fit. An advantage of the (TLI or) NNFI is that it is one of the fit indices less affected by sample size (Bentler, 1990; Bollen, 1990; Hu & Bentler, 1995, 1998)."

Kenny (2020): "The TLI (and the CFI) depends on the average size of the correlations in the data. If the average correlation between variables is not high, then the TLI will not be very high."

CFI - Comparative Fit Index (incremental fit)

Schermelleh-Engel (2003): "The CFI ranges from zero to one with higher values indimessageing better fit. A rule of thumb for this index is that .97 is indicative of good fit relative to the independence model, while values greater than .95 may be interpreted as an acceptable fit. Again a value of .97 seems to be more reasonable as an indimessageion of a good model fit than the often stated cutoff value of .95. Comparable to the NNFI, the CFI is one of the fit indices less affected by sample size."

Hooper (2008): "A cut-off criterion of CFI >= 0.90 was initially advanced however, recent studies have shown that a value greater than 0.90 is needed in order to ensure that misspecified models are not accepted (Hu and Bentler, 1999). From this, a value of CFI >= 0.95 is presently recognised as indicative of good fit (Hu and Bentler, 1999). Today this index is included in all SEM programs and is one of the most popularly reported fit indices due to being one of the measures least effected by sample size (Fan et al, 1999)."

Kenny (2020): "Because the TLI and CFI are highly correlated only one of the two should be reported. The CFI is reported more often than the TLI, but I think the CFI,s penalty for complexity of just 1 is too low and so I prefer the TLI even though the CFI is reported much more frequently than the TLI."

MFI – (absolute fit)

An absolute fit index proposed by MacDonald and Marsh (1990) that does not depend on a comparison with another model.

AIC – Akaike Information Criterion (degree of parsimony index)

Kenny (2020): "The AIC is a comparative measure of fit and so it is meaningful only when two different models are estimated. Lower values indicate a better fit and so the model with the lowest AIC is the best fitting model. There are somewhat different formulas given for the AIC in the literature, but those differences are not really meaningful as it is the difference in AIC that really matters. The AIC makes the researcher pay a penalty of two for every parameter that is estimated. One advantage of the AIC, BIC, and SABIC measures is that they can be computed for models with zero degrees of freedom, i.e., saturated or just-identified models."

CAIC – Consistent Akaike Information Criterion (degree of parsimony index)

A version of AIC that adjusts for sample size. Lower values indicate a better fit.

BIC – Bayesian Information Criterion (degree of parsimony index)

Lower values indicate a better fit.

Kenny (2020): "Whereas the AIC has a penalty of 2 for every parameter estimated, the BIC increases the penalty as sample size increases. The BIC places a high value on parsimony (perhaps too high)."

SABIC – Sample-Size Adjusted BIC (degree of parsimony index)

Kenny (2020): "Like the BIC, the sample-size adjusted BIC or SABIC places a penalty for adding parameters based on sample size, but not as high a penalty as the BIC. Several recent simulation studies (Enders & Tofighi, 2008; Tofighi, & Enders, 2007) have suggested that the SABIC is a useful tool in comparing models.

Value

A list with eigenvalues & fit coefficients.

Author(s)

Brian P. O'Connor

References

Hooper, D., Coughlan, J., & Mullen, M. (2008). Structural Equation Modelling: Guidelines for Determining Model Fit. *Electronic Journal of Business Research Methods, 6(1),* 53-60.

Kenny, D. A. (2020). *Measuring model fit.* http://davidaKenny.net/cm/fit.htm

McDonald, R. P. (1999). *Test theory: A unified treatment.* Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.

Lorenzo-Seva, U., Timmerman, M. E., & Kiers, H. A. (2011). The Hull method for selecting the number of common factors. *Multivariate Behavioral Research, 46,* 340-364.

Schermelleh-Engel, K., & Moosbrugger, H. (2003). Evaluating the fit of structural equation models: Tests of significance and descriptive goodness-of-fit measures. *Methods of Psychological Research Online, Vol.8(2),* pp. 23-74.

Tabachnick, B. G., & Fidell, L. S. (2019). *Using multivariate statistics* (pp. 560-564). New York, NY: Pearson.

Examples

```
# the Harman (1967) correlation matrix
ROOTFIT(data_Harman, Ncases = 305, extraction='ml')
ROOTFIT(data_Harman, Ncases = 305, extraction='paf')
ROOTFIT(data_Harman, Ncases = 305, extraction='pca')

# RSE data
ROOTFIT(data_RSE, corkind='pearson', extraction='ml')
ROOTFIT(data_RSE, corkind='pearson', extraction='paf')
ROOTFIT(data_RSE, corkind='pearson', extraction='pca')

# NEO-PI-R scales
ROOTFIT(data_NEOPIR, corkind='pearson', extraction='ml')
ROOTFIT(data_NEOPIR, corkind='pearson', extraction='paf')
ROOTFIT(data_NEOPIR, corkind='pearson', extraction='pca')
```

SALIENT *Salient loadings criterion for the number of factors*

Description

Salient loadings criterion for determining the number of factors, as recommended by Gorsuch. Factors are retained when they consist of a specified minimum number (or more) variables that have a specified minimum (or higher) loading value.

Usage

```
SALIENT(data, salvalue = .4, numsals = 3, max_cross = NULL, min_eigval = .7,
              corkind = 'pearson', extraction = 'paf', rotation = 'promax',
              loading_mat = 'structure', ppower = 3, iterpaf = 100,
              Ncases = NULL, verbose = TRUE, factormodel, rotate)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
salvalue	(optional) The loading value that is considered salient. Default = .40. This can also be a vector of up to three values, e.g., salvalue = c(.4, .5, .6).
numsals	(optional) The required number of salient loadings for a factor. Default = 3. This can also be a vector of up to three values, e.g., numsals = c(3, 2, 1).
max_cross	(optional) The maximum value for cross-loadings.
min_eigval	(optional) The minimum eigenvalue for including a factor in the analyses. Default = .7
corkind	(optional) The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
extraction	(optional) The factor extraction method for the analysis. The options are 'pca', 'paf' (the default), 'ml', 'image', 'minres', 'uls', 'ols', 'wls', 'gls', 'alpha', and 'fullinfo'.
rotation	(optional) The factor rotation method for the analysis. The orthogonal rotation options are: 'varimax' (the default), 'quartimax', 'bentlerT', 'equamax', 'geominT', 'bifactorT', 'entropy', and 'none'. The oblique rotation options are: 'promax' (the default), 'quartimin', 'oblimin', 'oblimax', 'simplimax', 'bentlerQ', 'geominQ', 'bifactorQ', and 'none'.
iterpaf	(optional) The maximum number of iterations for paf. Default value = 100
loading_mat	(optional) The kind of factor rotation matrix for an oblique rotation. The options are 'structure' (the default) or 'pattern'.
ppower	(optional) The power value to be used in a promax rotation (required only if rotation = 'promax'). Default value = 3
Ncases	The number of cases. Required only if data is a correlation matrix.
verbose	(optional) Should detailed results be displayed in console? TRUE (default) or FALSE
factormodel	(Deprecated.) Use 'extraction' instead.
rotate	(Deprecated.) Use 'rotation' instead.

Details

In this procedure for determining the number of factors, factors are retained when each factor has at least a specified minimum number variables (e.g., 3) that have loadings that are greater than or

equal to a specified minimum loading value (e.g., .40). Factor are considered trivial when they do not contain a sufficient number of salient loadings (Gorsuch, 1997, 2015; Boyd, 2011).

The procedure begins by extracting and rotating (if requested) an excessive number of factors. If the initial factor loadings do not meet the specified criteria, then the factor analysis is conducted again with one less factor and the loadings are again examined to determine whether the factor loadings meet the specified criteria. The procedure stops when a loading matrix meets the criteria, in which case the number of columns in the loading matrix is the number of factors according to the salient loadings criteria.

The initial, excessive number of factors for the procedure is determined using the **min_eigval** argument (for minimum eigenvalue). The default is .70, which can be adjusted (raised) when analyses produce an error caused by there being too few variables.

Although there is no consensus on what constitutes a 'salient' loading, an absolute value of .40 is common in the literature.

There are different versions of the salient loadings criterion method, which has not been extensively tested to date. The procedure is nevertheless considered promising by Gorsuch and others.

Some versions involve the use of multiple salient loading values, each with its own minimum number of variables. This can be done in the SALIENT function by providing a vector of values for the **salvalue** argument and a corresponding vector of values for the **numsals** argument. The maximum number of possible values is three, and there should be a logical order in the values, i.e., increasing values for salvalue and decreasing values for numsals.

It is also possible to place a restriction of the maximum value of the cross-loadings for the salient variables, e.g., requiring that a salient loading is not accompanied by cross-loadings on other variables that are greater than .15. Use the **max_cross** argument for this purpose, although it may be difficult to claim that cross-loadings should be small when the factors are correlated.

Value

The number of factors according to the salient loadings criterion.

Author(s)

Brian P. O'Connor

References

Boyd, K. C. (2011). Factor analysis. In M. Stausberg & S. Engler (Eds.), *The Routledge Handbook of Research Methods in the Study of Religion* (pp. 204-216). New York: Routledge.

Gorsuch, R. L. (1997). Exploratory factor analysis: Its role in item analysis. *Journal of Personality Assessment, 68,* 532-560.

Gorsuch, R. L. (2015). *Factor analysis*. Routledge/Taylor & Francis Group.

Examples

```
# the Harman (1967) correlation matrix
SALIENT(data_Harman, salvalue=.4, numsals=3, Ncases=305)
```

```
# Rosenberg Self-Esteem scale items, using Pearson correlations
SALIENT(data_RSE, salvalue=.4, numsals=3, corkind='pearson')

# NEO-PI-R scales
SALIENT(data_NEOPIR, salvalue = c(.4, .5, .6), numsals = c(3, 2, 1), extraction = 'paf',
        rotation='promax', loading_mat = 'pattern')
```

SCREE_PLOT *Scree plot of eigenvalues*

Description

Produces a scree plot of eigenvalues for raw data or for a correlation matrix.

Usage

```
SCREE_PLOT(data, corkind, Ncases, verbose)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal.The function internally determines whether the data are a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases for a correlation matrix. Required only if the entered data is a correlation matrix.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Value

totvarexpl	The eigenvalues and total variance explained

Author(s)

Brian P. O'Connor

Examples

```
# Field's RAQ factor analysis data
SCREE_PLOT(data_Field, corkind='pearson')

# the Harman (1967) correlation matrix
SCREE_PLOT(data_Harman)

# Rosenberg Self-Esteem scale items
```

```
SCREE_PLOT(data_RSE, corkind='polychoric')

# NEO-PI-R scales
SCREE_PLOT(data_RSE)
```

SESCREE *Standard Error Scree test*

Description

This is a linear regression operationalization of the scree test for determining the number of components. The results are purportedly identical to those from the visual scree test. The test is based on the standard error of estimate values that are computed for the set of eigenvalues in a scree plot. The number of components to retain is the point where the standard error exceeds 1/m, where m is the numbers of variables.

Usage

```
SESCREE(data, Ncases=NULL, corkind, verbose=TRUE)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
Ncases	The number of cases. Required only if data is a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Value

The number of components according to the Standard Error Scree test.

Author(s)

Brian P. O'Connor

References

Zoski, K., & Jurs, S. (1996). An objective counterpart to the visual scree test for factor analysis: the standard error scree test. *Educational and Psychological Measurement, 56(3),* 443-451.

Examples

```
# the Harman correlation matrix
SESCREE(data_Harman, Ncases=305, verbose=TRUE)

# the Rosenberg Self-Esteem Scale (RSE) using Pearson correlations
SESCREE(data_RSE, corkind='pearson', verbose=TRUE)

# the Rosenberg Self-Esteem Scale (RSE) using polychoric correlations
SESCREE(data_RSE, corkind='polychoric', verbose=TRUE)

# the NEO-PI-R scales
SESCREE(data_NEOPIR, verbose=TRUE)
```

SMT *Sequential chi-square model tests*

Description

A test for the number of common factors using the likelihood ratio test statistic values from maximum likelihood factor analysis estimations.

Usage

```
SMT(data, corkind, Ncases=NULL, verbose)
```

Arguments

data	An all-numeric dataframe where the rows are cases & the columns are the variables, or a correlation matrix with ones on the diagonal. The function internally determines whether the data are a correlation matrix.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', 'gamma', and 'polychoric'. Required only if the entered data is not a correlation matrix.
Ncases	The number of cases. Required only if data is a correlation matrix.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

Details

From Auerswald & Moshagen (2019):

"The fit of common factor models is often assessed with the likelihood ratio test statistic (Lawley, 1940) using maximum likelihood estimation (ML), which tests whether the model-implied covariance matrix is equal to the population covariance matrix. The associated test statistic asymptotically follows a Chi-Square distribution if the observed variables follow a multivariate normal distribution and other assumptions are met (e.g., Bollen, 1989). This test can be sequentially applied to factor models with increasing numbers of factors, starting with a zero-factor model. If the Chi-Square test statistic is statistically significant (with e.g., $p < .05$), a model with one additional factor, in

this case a unidimensional factor model, is estimated and tested. The procedure continues until a nonsignificant result is obtained, at which point the number of common factors is identified.

"Simulation studies investigating the performance of sequential Chi-Square model tests (SMT) as an extraction criterion have shown conflicting results. Whereas some studies have shown that SMT has a tendency to overextraction (e.g., Linn, 1968; Ruscio & Roche, 2012; Schonemann & Wang, 1972), others have indicated that the SMT has a tendency to underextraction (e.g., Green et al., 2015; Hakstian et al., 1982; Humphreys & Montanelli, 1975; Zwick & Velicer, 1986). Hayashi, Bentler, and Yuan (2007) demonstrated that overextraction tendencies are due to violations of regularity assumptions if the number of factors for the test exceeds the true number of factors. For example, if a test of three factors is applied to samples from a population with two underlying factors, the likelihood ratio test statistic will no longer follow a Chi-Square distribution. Note that the tests are applied sequentially, so a three-factor test is only employed if the two-factor test was incorrectly significant. Therefore, this violation of regularity assumptions does not decrease the accuracy of SMT, but leads to (further) overextractions if a previous test was erroneously significant. Additionally, this overextraction tendency might be counteracted by the lack of power in simulation studies with smaller sample sizes. The performance of SMT has not yet been assessed for non-normally distributed data or in comparison to most of the other modern techniques presented thus far in a larger simulation design." (p. 475)

Value

A list with the following elements:

`NfactorsSMT`	number of factors according to the SMT
`pvalues`	eigenvalues, chi-square values, & pvalues

Author(s)

Brian P. O'Connor

References

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. *Psychological Methods, 24(4),* 468-491.

Examples

```
# the Harman (1967) correlation matrix
SMT(data_Harman, Ncases=305, verbose=TRUE)

# Rosenberg Self-Esteem scale items, using Pearson correlations
SMT(data_RSE, corkind='polychoric', verbose=TRUE)

# NEO-PI-R scales
SMT(data_NEOPIR, verbose=TRUE)
```

Index

BIFACTOR, 4

COMPLEXITY, 10
CONGRUENCE, 11
CORRECTED_CORRELS, 13

data_Field, 15
data_Harman, 16
data_HS_1939, 17
data_NEOPIR, 17
data_RSE, 18
data_RSE_not_recoded, 19
data_RSE_sex, 19
data_SDT, 20
data_TabFid, 20
DIMTESTS, 21

EFA, 23, *27*
EFA.dimensions-defunct, 27
EFA.dimensions-package, 3
EFA_SCORES, 28
EMPKC, 31
ESEM, 33
EXTENSION_FA, 37

FACTORABILITY, 40
Factorial_Invariance, 42

IMAGE_FA *(*EFA.dimensions-defunct*)*, 27
INTERNAL.CONSISTENCY
*(*INTERNAL_CONSISTENCY*)*, 46
INTERNAL_CONSISTENCY, 46

LOCALDEP, 48

MAP, 50
MAXLIKE_FA *(*EFA.dimensions-defunct*)*, 27
MISSING_INFO, 51

NEVALSGT1, 52

OMEGA, 54

PA_FA *(*EFA.dimensions-defunct*)*, 27
PARALLEL, 59
PCA, 61
PLOT_Invariance, 64
POLYCHORIC_R, 65
PROCRUSTES, 67

RAWPAR, 68
RECODE, 71
ROOTFIT, 72

SALIENT, 76
SCREE_PLOT, 79
SESCREE, 80
SMT, 81