

# Package ‘EFA.dimensions’

March 17, 2023

**Title** Exploratory Factor Analysis Functions for Assessing Dimensionality

**Type** Package

**Version** 0.1.7.7

**Date** 2023-03-16

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

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

**Description** Functions for eleven procedures for determining the number of factors, including functions for parallel analysis and the minimum average partial test. There are also functions for conducting principal components analysis, principal axis factor analysis, maximum likelihood factor analysis, image factor analysis, and extension factor analysis, all of which can take 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. Varimax rotation, promax rotation, and Procrustes rotations can be performed. Additional functions focus on the factorability of a correlation matrix, the congruences between factors from different datasets, the assessment of local independence, the assessment of factor solution complexity, and internal consistency. Auerwald & Moshagen (2019, ISSN:1939-1463); Fabrigar & Wegener (2012, ISBN:978-0-19-973417-7); Field, Miles, & Field (2012, ISBN:978-1-4462-0045-2); O'Connor (2000, <doi:10.3758/bf03200807>); O'Connor (2001, ISSN:0146-6216).

**Imports** stats, psych, polycor, EFAtools, utils, mirt

**Suggests** lattice

**LazyLoad** yes

**LazyData** yes

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-03-17 11:30:18 UTC

**R topics documented:**

EFA.dimensions-package . . . . .	2
COMPLEXITY . . . . .	3
CONGRUENCE . . . . .	5
data_Field . . . . .	7
data_Harman . . . . .	7
data_NEOPIR . . . . .	8
data_RSE . . . . .	9
data_TabFid . . . . .	9
DIMTESTS . . . . .	10
EMPKC . . . . .	13
EXTENSION_FA . . . . .	15
FACTORABILITY . . . . .	18
IMAGE_FA . . . . .	19
INTERNAL.CONSISTENCY . . . . .	21
LOCALDEP . . . . .	25
MAP . . . . .	27
MAXLIKE_FA . . . . .	28
NEVALSGT1 . . . . .	30
PARALLEL . . . . .	32
PA_FA . . . . .	34
PCA . . . . .	35
POLYCHORIC_R . . . . .	36
PROCRUSTES . . . . .	38
PROMAX . . . . .	39
RAWPAR . . . . .	40
RECODE . . . . .	43
ROOTFIT . . . . .	44
SALIENT . . . . .	49
SCREE_PLOT . . . . .	50
SESCREE . . . . .	51
SMT . . . . .	52
VARIMAX . . . . .	54
<b>Index</b>	<b>56</b>

---

EFA.dimensions-package

*EFA.dimensions*

---

**Description**

This package provides exploratory factor analysis-related functions for assessing dimensionality.

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 are also functions for conducting principal components analysis (PCA), principal axis factor analysis (PA\_FA), maximum likelihood factor analysis (MAXLIKE\_FA), image factor analysis (IMAGE\_FA), and extension factor analysis (EXTENSION\_FA),

Varimax rotation (VARIMAX), promax rotation (PROMAX), and Procrustes rotations (PROCRUSTES) can be performed.

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 focus on the factorability of a correlation matrix (FACTORABILITY), the congruences between factors from different datasets (CONGRUENCE), the assessment of local independence (LOCALDEP), the assessment of factor solution complexity (COMPLEXITY), and internal consistency (INTERNAL.CONSISTENCY).

## 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.
- Fabrigar, L. R., & Wegener, D. T. (2012). *Exploratory factor analysis*. New York, NY: Oxford University Press. ISBN:978-0-19-973417-7
- Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using R*. Los Angeles, CA: Sage. ISBN:978-1-4462-0045-2
- 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.
- O'Connor, B. P. (2001). EXTENSION: SAS, SPSS, and MATLAB programs for extension analysis. *Applied Psychological Measurement, 25*, p. 88. doi:10.1177/01466216010251011.
- Thompson, L. A. 2007. R (and S-PLUS) Manual to Accompany Agresti's Categorical Data Analysis (2002) 2nd edition.

---

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 Petterson 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, rotate='PROMAX', verbose=FALSE)
COMPLEXITY(loadings=PCAoutput$structure, verbose=TRUE)

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

# NEO-PI-R scales
PCAoutput <- PCA(data_NEOPIR, Nfactors = 5, rotate='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 Berg (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,
               rotate='VARIMAX', verbose=FALSE)

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

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

target <- PCA(data_NEOPIR[501:1000,], corkind='pearson', Nfactors = 3,
              rotate='VARIMAX', verbose=FALSE)
CONGRUENCE(target$loadingsV, loadings$loadingsV, 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, rotate='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, factormodel='PCA', Ndatasets=100, percentile=95,
        Ncases=305, verbose=TRUE)
```

---

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, factormodel='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 300 observations on the 10 items from the Rosenberg Self-Esteem scale.

**Usage**

```
data(data_RSE)
```

**References**

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

**Examples**

```
# MAP test on the Rosenberg Self-Esteem Scale (RSE) item data
MAP(data_RSE, corkind='polychoric', verbose=TRUE)

# DIMTESTS on the Rosenberg Self-Esteem Scale (RSE) item data
DIMTESTS(data_RSE, tests = c('CD', 'EMPKC', 'HULL', 'RAWPAR', 'NEVALSGT1'), Ncases=1000, display=2)

# parallel analysis of the Rosenberg Self-Esteem Scale (RSE) item data
RAWPAR(data_RSE, factormodel='PCA', Ndatasets=100, percentile=95,
        corkind='pearson', verbose=TRUE)
```

---

`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, factormodel='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
PA_FA(data_TabFid, corkind='pearson', Nfactors = 5, iterpaf = 50,
       rotate='PROMAX', ppower = 4, verbose=TRUE)
```

---

DIMTESTS	<i>Tests for the number of factors</i>
----------	--

---

**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)
```

**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, SESCOREE, 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 tests for the number of factors.

The HULL method option uses the HULL function (and its defaults) in the EFAtools package.

From Auerswald & Moshagen (2019):

"The Hull method (Lorenzo-Seva et al., 2011) is an approach based on the Hull heuristic used in other areas of model selection (e.g., Ceulemans & Kiers, 2006). Similar to nongraphical variants of Cattell's scree plot, the Hull method attempts to find an elbow as justification for the number of common factors. However, instead of using the eigenvalues relative to the number of factors, the Hull method relies on goodness-of-fit indices relative to the model degrees of freedom of the proposed model."

The CD (comparison data) method option uses the CD function (and its defaults) in the EFAtools package. The CD method can only be conducted on raw data and not on correlation matrices.

From Auerswald & Moshagen (2019):

"Ruscio and Roche (2012) suggested an approach that finds the number of factors by determining the solution that reproduces the pattern of eigenvalues best (comparison data, CD). CD takes previous factors into account by generating comparison data of a known factorial structure in an iterative procedure. Initially, CD compares whether the simulated comparison data with one underlying factor ( $j = 1$ ) reproduce the pattern of empirical eigenvalues significantly worse compared with a two-factor solution ( $j + 1$ ). If this is the case, CD increases  $j$  until further improvements are nonsignificant or a preset maximum of factors is reached."

"No single extraction criterion performed best for every factor model. In unidimensional and orthogonal models, traditional PA, EKC, and Hull consistently displayed high hit rates even in small samples. Models with correlated factors were more challenging, where CD and SMT outperformed other methods, especially for shorter scales. Whereas the presence of cross-loadings generally increased accuracy, non-normality had virtually no effect on most criteria. We suggest researchers use a combination of SMT and either Hull, the EKC, or traditional PA, because the number of factors was almost always correctly retrieved if those methods converged. When the results of this combination rule are inconclusive, traditional PA, CD, and the EKC performed comparatively well. However, disagreement also suggests that factors will be harder to detect, increasing sample size requirements to  $N \geq 500$ ."

The recommended tests for the number of factors are: EMPKC, HULL, and RAWPAR. The MAP test is also recommended for principal components analyses. Other possible methods (e.g., NEVALSGT1, SALIENT, SESCREEN) are less well-validated and are included for research purposes.

**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)
```

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 extracts 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)
```

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,
             factormodel, rotate,
             Nfactors, NfactorsHO,
             Ndatasets, percentile,
             salvalue, numsals,
             iterpaf, ppower,
             verbose)
```

**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: 'Nsaliend' 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'.
factormodel	The factor extraction method. The options are: 'PAF' for principal axis / common factor analysis; 'PCA' for principal components analysis; 'ML' for maximum likelihood.

rotate	The factor rotation method. The options are: 'PROMAX', , and 'none'.
Nfactors	An integer indicating the user-determined number of factors (required only if roottest = 'user').
NfactorsH0	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 factormodel = 'PAF'). Suggested value: 100
ppower	The power value to be used in a promax rotation (required only if rotate = 'PROMAX'). Suggested value: 3
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

### 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, Gorsuch (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',
              factormodel='PCA', rotate='PROMAX',
```

```
Nfactors=2, NfactorsH0=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',
  factormodel='PCA', rotate='PROMAX',
  Nfactors=4, NfactorsH0=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')
```

---

IMAGE\_FA

*Image factor analysis*

---

**Description**

Image factor analysis

**Usage**

```
IMAGE_FA(data, corkind, Nfactors=NULL, Ncases=NULL, rotate, ppower, 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.
Nfactors	The number of factors to extract.
Ncases	The number of cases. Required only if data is a correlation matrix.
rotate	The factor rotation method. The options are: 'PROMAX', , and 'none'.
ppower	The power value to be used in a promax rotation (required only if rotate = 'PROMAX'). Suggested value: 3
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

**Details**

"Image analysis is a principal factor variant in its usual application. As in the case of the principal axes with estimated communalities procedure, it is a principal factor variant in the sense that, after appropriate alterations are made to the correlation matrix, that matrix can be submitted to a principal factor program to find the desired factors. The program will then minimize the residuals of the particular matrix submitted. The differences in image analysis and other principal factor variants lie primarily in the alterations of the correlation matrix before the factors are extracted. Image factors can also be extracted by maximum likelihood procedures (Joreskog, 1969b)." (Gorsuch, 1974, p. 103)

Image analysis is a common factor analysis of the predictable variance of a set of variables. Each variable,  $X_i$ , is regressed on all other variables to obtain the predicted  $X_i$  values. The covariance matrix of the predicted  $X_i$ s is then factored to produce an image analysis. The image of a variable is the portion of a variable which is predicted from other variables.

The present function is an implementation of the Harris (1962) variant of this procedure (see also Velicer, 1974), which is a noniterative approximation to canonical component analysis. The results are identical to the image factor analysis results that are produced by SAS and SPSS.

**Value**

A list with the following elements:

totvarexplNOROT	The eigenvalues and total variance explained
totvarexplROT	The rotation sums of squared loadings and total variance explained for the rotated loadings
loadingsNOROT	The unrotated factor loadings
loadingsROT	The rotated factor loadings (for varimax rotation)
structure	The structure matrix (for promax rotation)
pattern	The pattern matrix (for promax rotation)
correls	The correlations between the factors (for promax rotation)

cormat\_reproduced

The reproduced correlation matrix, based on the rotated loadings

### Author(s)

Brian P. O'Connor

### References

Gorsuch, R. L. (1983). *Factor analysis ( 2nd ed. )*. Hillsdale, N.J.: Erlbaum, 1983.

Guttman, L. (1953). Image theory for the structure of quantitative variates. *Psychometrika* 18, 277-296.

Harris, C. W. (1962). Some Rao-Guttman relationships. *Psychometrika*, 27, 247-63.

Velicer, W. F. (1974). A comparison of the stability of factor analysis, principal component analysis, and rescaled image analysis. *Educational and Psychological Measurement*, 34(3), 563-572.

### Examples

```
# the Harman (1967) correlation matrix
IMAGE_FA(data_Harman, Nfactors=2, Ncases=305, rotate='PROMAX', ppower = 4, verbose=TRUE)

# Rosenberg Self-Esteem scale items
IMAGE_FA(data_RSE, corkind='gamma', Nfactors=2, rotate='PROMAX', ppower = 4, verbose=TRUE)

# NEO-PI-R scales
IMAGE_FA(data_NEOPIR, corkind='pearson', Nfactors=5, rotate='PROMAX', ppower = 4, verbose=TRUE)
```

---

INTERNAL.CONSISTENCY *Internal consistency reliability coefficients*

---

### Description

Internal consistency reliability coefficients

### Usage

```
INTERNAL.CONSISTENCY(data, factormodel = 'ML', reverse_these = NULL,
                      auto_reverse = TRUE, verbose)
```

**Arguments**

data	An all-numeric dataframe where the rows are cases & the columns are the variables.
factormodel	(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

**Details**

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

If item names are provided for the 'reverse\_these' argument, then auto\_reverse is not conducted.

The following helpful descriptions of Cronbach's alpha and of omega total are direct quotes from McNeish (2017, pp. 414-417):

**Cronbach's Alpha**

"One can interpret the value of Cronbach's alpha in one of many different ways:

1. Cronbach's alpha is the correlation of the scale of interest with another scale of the same length that intends to measure the same construct, with different items, taken from the same hypothetical pool of items (Kline, 1986).
2. The square root of Cronbach's alpha is an estimate of the correlation between observed scores and true scores (Nunnally & Bernstein, 1994).
3. Cronbach's alpha is the proportion of the variance of the scale that can be attributed to a common source (DeVellis, 1991).
4. Cronbach's alpha is the average of all possible split-half reliabilities from the set of items (Pedhazur & Schmelkin, 1991). (It is important to note the correlation between the two parts is not the split half reliability, but is used to find the split half reliability found by the Spearman-Brown prophecy formula.)

Under certain assumptions, Cronbach's alpha is a consistent estimate of the population internal consistency; however, these assumptions are quite rigid and are precisely why methodologists have argued against the use of Cronbach's alpha.

The assumptions of Cronbach's alpha are:

1. The scale adheres to tau equivalence, i.e., that each item on a scale contributes equally to the total scale score. Tau equivalence tends to be unlikely for most scales that are used in empirical research some items strongly relate to the construct while some are more weakly related.
2. Scale items are on a continuous scale and normally distributed. Cronbach's alpha is largely based on the observed covariances (or correlations) between items. In most software implementations of Cronbach's alpha (such as in SAS and SPSS), these item covariances are calculated using a

Pearson covariance matrix. A well-known assumption of Pearson covariance matrices is that all variables are continuous in nature. Otherwise, the elements of the matrix can be substantially biased downward. However, it is particularly common for psychological scales to contain items that are discrete (e.g., Likert or binary response scales), which violates this assumption. If discrete items are treated as continuous, the covariance estimates will be attenuated, which ultimately results in underestimation of Cronbach's alpha because the relations between items will appear smaller than they actually are. To accommodate items that are not on a continuous scale, the covariances between items can instead be estimated with a polychoric covariance (or correlation) matrix rather than with a Pearson covariance matrix. Polychoric covariance matrices assume that there is an underlying normal distribution to discrete responses.

3. The errors of the items do not covary. Correlated errors occur when sources other than the construct being measured cause item responses to be related to one another.

4. The scale is unidimensional. Though Cronbach's alpha is sometimes thought to be a measure of unidimensionality because its colloquial definition is that it measures how well items stick together, unidimensionality is an assumption that needs to be verified prior to calculating Cronbach's alpha rather than being the focus of what Cronbach's alpha measures. Internal consistency is necessary for unidimensionality but that internal consistency is not sufficient for demonstrating unidimensionality. That is, items that measure different things can still have a high degree of interrelatedness, so a large Cronbach's alpha value does not necessarily guarantee that the scale measures a single construct. As a result, violations of unidimensionality do not necessarily bias estimates of Cronbach's alpha. In the presence of a multidimensional scale, Cronbach's alpha may still estimate the interrelatedness of the items accurately and the interrelatedness of multidimensional items can in fact be quite high."

### **Omega total**

"Omega total is an internal consistency coefficient that assumes that the scale is unidimensional. Omega estimates the reliability for the composite of items on the scale (which is conceptually similar to Cronbach's alpha). Under the assumption that the construct variance is constrained to 1 and that there are no error covariances, omega total is calculated from factor analysis output (loadings and error/uniqueness values). Tau equivalence is no longer assumed and the potentially differential contribution of each item to the scale must be assessed. Omega total is a more general version of Cronbach's alpha and actually subsumes Cronbach's alpha as a special case. More simply, if tau equivalence is met, omega total will yield the same result as Cronbach's alpha but omega total has the flexibility to accommodate congeneric scales, unlike Cronbach's alpha."

### **Root Mean Square Residual (rmsr)**

rmsr 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. The rmsr coefficient is included in the internal consistency output as an index of the degree of fit of a one-factor model to the item data.

### **Standardized Cronbach's Alpha**

Standardized alpha should be used when items have different scale ranges, e.g., some items are 1-to-7, and other items are 1-to-4, or 1-to-100. Regular alpha is based on covariances, whereas standardized alpha is based on correlations, wherein the items have identical standard deviations. Items in different metrics should be standardized before computing scale scores.

**Value**

A list with the following elements:

<code>int.consist_scale</code>	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
<code>int.consist_dropped</code>	A matrix of the <code>int.consist_scale</code> values for when each item, in turn, is <code>int.consist_dropped</code> from the analyses
<code>item_stats</code>	The item means, standard deviations, and item-total correlations
<code>resp_opt_freqs</code>	The response option frequencies
<code>resp_opt_props</code>	The response option proportions
<code>new_data</code>	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 to : A tutorial. *Psychological Assessment*, 31(12), 13951411.

**Examples**

```
# Rosenberg Self-Esteem scale items -- without reverse-coding
INTERNAL.CONSISTENCY(data_RSE, factormodel = 'PAF',
                      reverse_these = NULL, auto_reverse = FALSE, verbose=TRUE)

# Rosenberg Self-Esteem scale items -- with auto_reverse-coding
INTERNAL.CONSISTENCY(data_RSE, factormodel = 'PAF',
                      reverse_these = NULL, auto_reverse = TRUE, verbose=TRUE)

# Rosenberg Self-Esteem scale items -- another way of reverse-coding
INTERNAL.CONSISTENCY(data_RSE, factormodel = 'PAF',
                      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:

<code>correlations</code>	The correlation matrix
<code>residcor</code>	The residualized (partial) correlation matrix
<code>eigenvalues</code>	The eigenvalues
<code>resid_Q3</code>	A matrix with the Q3 statistic values described by Yen (1984)
<code>resid_LD</code>	A matrix with the X2 statistic values described by Chen and Thissen (1997)
<code>resid_LDG2</code>	A matrix with the G2 statistic values described by Chen and Thissen (1997)
<code>resid_JSI</code>	A matrix with the jack-knife statistic values described by Edwards et al. (2018)
<code>localdep_stats</code>	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 extracted 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:

```
totvarexp|NOROT
```

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)
```

---

MAXLIKE\_FA

*Maximum likelihood factor analysis*

---

**Description**

Maximum likelihood factor analysis

**Usage**

```
MAXLIKE_FA(data, corkind, Nfactors=NULL, Ncases=NULL, rotate, ppower, verbose)
```

**Arguments**

<code>data</code>	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.
<code>corkind</code>	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.
<code>Nfactors</code>	The number of factors to extract.
<code>Ncases</code>	The number of cases. Required only if data is a correlation matrix.
<code>rotate</code>	The factor rotation method. The options are: 'PROMAX', 'VARIMAX', and 'none'.
<code>ppower</code>	The power value to be used in a promax rotation (required only if rotate = 'PROMAX'). Suggested value: 3
<code>verbose</code>	Should detailed results be displayed in console? TRUE (default) or FALSE

**Details**

This function relies heavily on the R `factanal` function, and it uses the `fa` from the `psych` package when `factanal` produces an error.

**Value**

A list with the following elements:

<code>totvarexplNOROT</code>	The eigenvalues and total variance explained
<code>totvarexplROT</code>	The rotation sums of squared loadings and total variance explained for the rotated loadings
<code>loadingsNOROT</code>	The unrotated factor loadings
<code>loadingsROT</code>	The rotated factor loadings (for varimax rotation)
<code>structure</code>	The structure matrix (for promax rotation)
<code>pattern</code>	The pattern matrix (for promax rotation)
<code>correls</code>	The correlations between the factors (for promax rotation)
<code>cormat_reproduced</code>	The reproduced correlation matrix, based on the rotated loadings
<code>chisqMODEL</code>	The model chi square statistic
<code>dfMODEL</code>	The model degrees of freedom
<code>pvalue</code>	The model p-value
<code>fit_coefficients</code>	Model fit coefficients

**Author(s)**

Brian P. O'Connor

## References

Reyment, R., Joreskog, K., & Marcus, L. F. (1996). *Applied Factor Analysis in the Natural Sciences*. Cambridge, MA: Cambridge University Press.

## Examples

```
# the Harman (1967) correlation matrix
MAXLIKE_FA(data_Harman, Nfactors = 2, Ncases = 305,
            rotate='PROMAX', ppower = 4, verbose=TRUE)

# Rosenberg Self-Esteem scale items
MAXLIKE_FA(data_RSE, corkind='gamma', Nfactors = 2,
            rotate='PROMAX', ppower = 4, verbose=TRUE)

# NEO-PI-R scales
MAXLIKE_FA(data_NEOPIR, corkind='pearson', Nfactors = 5,
            rotate='PROMAX', ppower = 4, verbose=TRUE)
```

---

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)
```

## 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.

totvarexp1NOROT

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)
```

---

 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, Ncases, Ndatasets=100, factormodel='PCA', percentile='95',
         corkind='pearson', verbose=TRUE)
```

**Arguments**

Nvars	The number of variables.
Ncases	The number of cases.
Ndatasets	An integer indicating the # of random data sets for parallel analyses.
factormodel	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



## 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 *i*th eigenvalue from the actual data is greater than the *i*th 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 extract 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, factormodel='PCA', percentile=95,  
corkind='pearson', verbose=TRUE)
```

---

PA\_FA *Principal axis (common) factor analysis*

---

### Description

Principal axis (common) factor analysis with squared multiple correlations as the initial communal-ity estimates

### Usage

```
PA_FA(data, corkind, Nfactors=NULL, Ncases=NULL, iterpaf, rotate, ppower, 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.
Nfactors	The number of factors to extract.
Ncases	The number of cases. Required only if data is a correlation matrix.
iterpaf	The maximum number of iterations.
rotate	The factor rotation method. The options are: 'PROMAX', , and 'none'.
ppower	The power value to be used in a promax rotation (required only if rotate = 'PROMAX'). Suggested value: 3
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

### Value

A list with the following elements:

totvarexp1NOROT	The eigenvalues and total variance explained
totvarexp1ROT	The rotation sums of squared loadings and total variance explained for the rotated loadings
loadingsNOROT	The unrotated factor loadings
loadingsROT	The rotated factor loadings (for varimax rotation)
structure	The structure matrix (for promax rotation)
pattern	The pattern matrix (for promax rotation)
correls	The correlations between the factors (for promax rotation)
cormat_reproduced	The reproduced correlation matrix, based on the rotated loadings
fit_coefficients	Model fit coefficients

**Author(s)**

Brian P. O'Connor

**Examples**

```
# the Harman (1967) correlation matrix
PA_FA(data_Harman, corkind='pearson', Nfactors = 2, Ncases=305, iterpaf = 50,
       rotate='PROMAX', ppower = 4, verbose=TRUE)

# Rosenberg Self-Esteem scale items
PA_FA(data_RSE, corkind='polychoric', Nfactors = 2, iterpaf = 50,
       rotate='PROMAX', ppower = 4, verbose=TRUE)

# NEO-PI-R scales
PA_FA(data_NEOPIR, corkind='pearson', Nfactors = 5, iterpaf = 50,
       rotate='PROMAX', ppower = 4, verbose=TRUE)
```

---

 PCA

*Principal components analysis*


---

**Description**

Principal components analysis

**Usage**

```
PCA(data, corkind, Nfactors=NULL, Ncases=NULL, rotate, ppower, 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.
Nfactors	The number of components to extract.
Ncases	The number of cases. Required only if data is a correlation matrix.
rotate	The factor rotation method. The options are: 'PROMAX', , and 'none'.
ppower	The power value to be used in a promax rotation (required only if rotate = 'PROMAX'). Suggested value: 3
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

**Value**

A list with the following elements:

totvarexp1NOROT	The eigenvalues and total variance explained
totvarexp1ROT	The rotation sums of squared loadings and total variance explained for the rotated loadings
loadingsNOROT	The unrotated factor loadings
loadingsROT	The rotated factor loadings (for varimax rotation)
structure	The structure matrix (for promax rotation)
pattern	The pattern matrix (for promax rotation)
correls	The correlations between the factors (for promax rotation)
cormat_reproduced	The reproduced correlation matrix, based on the rotated loadings
fit_coefficients	Model fit coefficients

**Author(s)**

Brian P. O'Connor

**Examples**

```
# the Harman (1967) correlation matrix
PCA(data_Harman, Nfactors=2, Ncases=305, rotate='PROMAX', ppower = 4, verbose=TRUE)

# Rosenberg Self-Esteem scale items
PCA(data_RSE, corkind='polychoric', Nfactors=2, rotate='PROMAX', ppower = 4, verbose=TRUE)

# NEO-PI-R scales
PCA(data_NEOPIR, corkind='pearson', Nfactors=5, rotate='PROMAX', ppower = 4, verbose=TRUE)
```

---

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')
```

---

 PROCUSTES

*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, rotate='PROMAX', verbose=FALSE)

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

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

---

PROMAX	<i>promax rotation</i>
--------	------------------------

---

## Description

promax rotation

## Usage

```
PROMAX(loadings, ppower, verbose)
```

## Arguments

loadings	A loading matrix.
ppower	The exponent for the promax target matrix. 'ppower' must be 1 or greater. '4' is a conventional value.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

**Details**

This function uses the R built-in promax function and provides additional output.

**Value**

A list with the following elements:

loadingsNOROT	The unrotated loadings
pattern	The pattern matrix (for promax rotation)
structure	The structure matrix (for promax rotation)
phi	The correlations between the factors (for promax rotation)
cormat_reproduced	The reproduced correlation matrix, based on the rotated loadings

**Author(s)**

Brian P. O'Connor

**Examples**

```
# the Harman (1967) correlation matrix
PCAoutput <- PCA(data_Harman, Nfactors = 2, Ncases=305, rotate='none', verbose=TRUE)
PROMAX(PCAoutput$loadingsNOROT, ppower = 4, verbose=TRUE)

# Rosenberg Self-Esteem scale items
PCAoutput <- PCA(data_RSE, corkind='polychoric', Nfactors = 2, rotate='none', verbose=TRUE)
PROMAX(PCAoutput$loadingsNOROT, ppower = 4, verbose=TRUE)

# NEO-PI-R scales
PCAoutput <- PCA(data_NEOPIR, corkind='pearson', Nfactors = 5, rotate='none', verbose=TRUE)
PROMAX(PCAoutput$loadingsNOROT, ppower = 4, 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, factormodel, Ndatasets, percentile,
        corkind, corkindRAND, 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.
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.
factormodel	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.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

**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 *i*th eigenvalue from the actual data is greater than the *i*th 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 extract 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.

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', factormodel='PCA', Ndatasets=100,
        percentile=95, corkind='pearson', verbose=TRUE)

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

# Rosenberg Self-Esteem scale items, using Pearson correlations
```

```

RAWPAR(data_RSE, randtype='permuted', factormodel='PCA', Ndatasets=100,
        percentile=95, corkind='pearson', corkindRAND='pearson', verbose=TRUE)

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

# NEO-PI-R scales
RAWPAR(data_NEOPIR, randtype='generated', factormodel='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[, 'Q3'], type = 'reverse')
table(data_RSE_rev); table(data_RSE[, '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[, 'Q3'], old = c(0,1,2,3,4), new = c(1,2,3,4,5))
table(data_RSE_rev); table(data_RSE[, 'Q3'])

# changing the metric/range of the third item (Q3) responses for the
# Rosenberg Self-Esteem scale data
data_RSE_rev <- RECODE(data_RSE[, 'Q3'], type = 'new_range',
                      real_min = 1, real_max = 4, new_min = 1, new_max = 5 )
table(data_RSE_rev); table(data_RSE[, '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, factormodel='PAF', 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 upon which a correlation matrix is based. Required only if data is a correlation matrix.
factormodel	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

**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  $1.0 - \text{mean-squared residual} / \text{mean-squared correlation}$ , 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 extracted (implying that more factors should be extracted). Values of CAF close to one mean that the residual matrix is free of common variance after the q factors have been extracted (i.e., no more factors should be extracted)."

### **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  $\leq .05$ . According to Browne and Cudeck (1993), RMSEA values  $\leq .05$  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 indicating 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 indication 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  $CFI \geq 0.90$  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  $CFI \geq 0.95$  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 affected 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, factormodel='ML')
ROOTFIT(data_Harman, Ncases = 305, factormodel='PAF')
ROOTFIT(data_Harman, Ncases = 305, factormodel='PCA')

# RSE data
ROOTFIT(data_RSE, corkind='pearson', factormodel='ML')
ROOTFIT(data_RSE, corkind='pearson', factormodel='PAF')
ROOTFIT(data_RSE, corkind='pearson', factormodel='PCA')

# NEO-PI-R scales
ROOTFIT(data_NEOPIR, corkind='pearson', factormodel='ML')
ROOTFIT(data_NEOPIR, corkind='pearson', factormodel='PAF')
ROOTFIT(data_NEOPIR, corkind='pearson', factormodel='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, numsals, 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.
salvalue	The loading value that is considered salient. Default = .40
numsals	The required number of salient loadings for a factor. Default = 3
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

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

**Examples**

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

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

# Rosenberg Self-Esteem scale items, using polychoric correlations
SALIENT(data_RSE, salvalue=.4, numsals=3, corkind='polychoric', verbose=TRUE)

# NEO-PI-R scales
SALIENT(data_NEOPIR, salvalue=.4, numsals=3, verbose=TRUE)
```

---

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**

totvarexp1	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)
```

**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 overextract (e.g., Linn, 1968; Ruscio & Roche, 2012; Schonemann & Wang, 1972), others have indicated that the SMT has a tendency to underextract (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)
```

---

 VARIMAX

*varimax rotation*


---

### Description

varimax rotation

### Usage

```
VARIMAX(loadings, normalize = TRUE, verbose)
```

### Arguments

loadings	A loading matrix.
normalize	Should Kaiser normalization be performed? If so the rows of x are re-scaled to unit length before rotation, and scaled back afterwards. Default = TRUE.
verbose	Should detailed results be displayed in console? TRUE (default) or FALSE

### Details

This function uses the R built-in varimax function and provides additional output.

### Value

A list with the following elements:

loadingsNOROT	The unrotated loadings
loadingsV	The varimax-rotated loadings
rotmatV	The rotation matrix
cormat_reproduced	The reproduced correlation matrix, based on the rotated loadings

### Author(s)

Brian P. O'Connor

**Examples**

```
# the Harman (1967) correlation matrix
PCAoutput <- PCA(data_Harman, Nfactors = 2, Ncases=305, rotate='none', verbose=TRUE)
VARIMAX(PCAoutput$loadingsNOROT, verbose=TRUE)

# Rosenberg Self-Esteem scale items
PCAoutput <- PCA(data_RSE, corkind='polychoric', Nfactors = 2, rotate='none', verbose=TRUE)
VARIMAX(PCAoutput$loadingsNOROT, verbose=TRUE)

# NEO-PI-R scales
PCAoutput <- PCA(data_NEOPIR, Nfactors = 5, rotate='none', verbose=TRUE)
VARIMAX(PCAoutput$loadingsNOROT, verbose=TRUE)
```

# Index

COMPLEXITY, [3](#)  
CONGRUENCE, [5](#)

[data\\_Field](#), [7](#)  
[data\\_Harman](#), [7](#)  
[data\\_NEOPIR](#), [8](#)  
[data\\_RSE](#), [9](#)  
[data\\_TabFid](#), [9](#)  
DIMTESTS, [10](#)

EFA.dimensions-package, [2](#)  
EMPKC, [13](#)  
EXTENSION\_FA, [15](#)

FACTORABILITY, [18](#)

IMAGE\_FA, [19](#)  
INTERNAL . CONSISTENCY, [21](#)

LOCALDEP, [25](#)

MAP, [27](#)  
MAXLIKE\_FA, [28](#)

NEVALSGT1, [30](#)

PA\_FA, [34](#)  
PARALLEL, [32](#)  
PCA, [35](#)  
POLYCHORIC\_R, [36](#)  
PROCRUSTES, [38](#)  
PROMAX, [39](#)

RAWPAR, [40](#)  
RECODE, [43](#)  
ROOTFIT, [44](#)

SALIENT, [49](#)  
SCREE\_PLOT, [50](#)  
SESCREE, [51](#)  
SMT, [52](#)

VARIMAX, [54](#)