Package ‘paramap’

March 17, 2019

Type Package

Title Factor Analysis Functions for Assessing Dimensionality

Version 1.9

Date 2019-03-16

Author Brian P. O'Connor

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

Description Factor analysis-related functions and datasets for assessing dimensionality.

Imports stats, utils, psych, polycor

Suggests lattice

LazyLoad yes

LazyData yes

License GPL (>= 2)

NeedsCompilation no

R topics documented:

`paramap-package` *paramap*

Description

This package provides factor analysis-related functions for assessing dimensionality. Users can request that the analyses be conducted using polychoric correlations, which are preferable to Pearson correlations for item-level data, and a polychoric correlation matrix is returned for possible further analyses. 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, or polychoric correlations. Varimax rotation, promax rotation, and Procrustes rotations can be performed.

`CONGRUENCE` *factor solution congruence*

Description

This function aligns two factor loading matrices and computes the factor solution congruence and the root mean square residual.

Usage

```
CONGRUENCE(target, loadings, display)
```

Arguments

`target`	The target loading matrix.
`loadings`	The loading matrix that will be aligned with the target.
`display`	Display the results? The options are TRUE 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).

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.

Examples

```
# RSE data
loadings <- PCA(data_RSE[1:150,],   corkind='pearson', nfactors = 3,
                rotate='varimax', display=FALSE)
target   <- PCA(data_RSE[151:300,], corkind='pearson', nfactors = 3,
                rotate='varimax', display=FALSE)
CONGRUENCE(target$loadingsROT, loadings$loadingsROT, display=TRUE)

## Not run:
# NEO-PI-R data
loadings <- PCA(data_NEOPIR[1:500,], corkind='pearson', nfactors = 3,
                rotate='varimax', display=FALSE)
target <- PCA(data_NEOPIR[501:1000,], corkind='pearson', nfactors = 3,
              rotate='varimax', display=FALSE)
CONGRUENCE(target$loadingsROT, loadings$loadingsROT, display=TRUE)

## End(Not run)
```

`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

```
## Not run:
# MAP test on the Harman correlation matrix
MAP(data_Harman, display=TRUE)

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

## End(Not run)
```

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

```
head(data_NEOPIR)

## Not run:
# MAP test on the data_NEOPIR data
MAP(data_NEOPIR, corkind='pearson', display=TRUE)

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

## End(Not run)
```

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)
```

Examples

```
head(data_RSE)

## Not run:
# MAP test on the Rosenberg Self-Esteem Scale (RSE) data
MAP(data_RSE, corkind='pearson', display=TRUE)

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

## End(Not run)
```

data_WISC *data_WISC*

Description

A data frame with scores for 175 cases on 10 WISC-R subscales, used by Tabacknick & Fidell (2013, p. 737) in their section on confirmatory factor analysis.

Usage

```
data(data_WISC)
```

References

Tabachnik, B. G., & Fidell, L. S. (2014). *Using multivariate statistics.* New York, NY: Pearson.

Examples

```
head(data_WISC)

## Not run:
# MAP test on the data_WISC data
MAP(data_WISC, corkind='pearson', display=TRUE)

# parallel analysis of the data_WISC data
RAWPAR(data_WISC, extract='PCA', Ndatasets=100, percentile=95,
        corkind='pearson', display=TRUE)

## End(Not run)
```

EXTENSION_FA *extension factor analysis*

Description

Extension factor 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.

Usage

```
EXTENSION_FA(data, Ncore, Next, higherorder, roottest,
             corkind, corkindRAND, extract, rotate, Nfacts,
             NfactsHO, Ndatasets, percentile, salvalue, numsals,
             iterpaf, iterml, tolerml, ppower)
```

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.

`corkind` The kind of correlation matrix to be used. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'.

`corkindRAND` The kind of correlation matrix to be used for the random data when roottest = 'parallel'. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'. These options are included for research purposes. In most applications, it is probably best to use Pearson correlations, which is the default.

`higherorder` Should a higher-order factor analysis be conducted? The options are TRUE or FALSE.

`roottest` The method for determing the number of factors. The options are: 'Nsalient' for number of salient loadings (see salvalue & numsals below); 'parallel' for parallel analysis (see Ndatasets & percentile below); 'MAP' for Velicer's minimum average partial test; 'SEscree' for the standard error scree test; 'nevals>1' for the number of eigenvalues > 1; and 'user' for a user-specified number of factors (see Nfacts & NfactsHO below).

`Nfacts` An integer indicating the user-determined number of factors (required only if roottest = 'user').

`NfactsHO` An integer indicating the user-determined number of higher order factors (required only if roottest = 'user' and higherorder = TRUE).

`extract` 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', 'varimax', and 'none'.

`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 extract = 'PAF'). Suggested value: 100

`iterml` The maximum # of iterations for a maximum likelihood analysis (required only if extract = 'ML'). Suggested value: 100

`tolerml` The tolerance value for a maximum likelihood analysis (required only if extract = 'ML'). Suggested value: .001

`ppower` The power value to be used in a promax rotation (required only if rotate = 'promax'). Suggested value: 3

Details

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

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

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

Value

A list with the following elements:

`fits1` eigenvalues & fit coefficients for the first set of core variables

`rff` factor intercorrelations

`corelding` core variable loadings on the factors

`extcorrel` extension variable correlations with the factors

`fits2` eigenvalues & fit coefficients for the higher order factor analysis

`rfflding` factor intercorrelations from the first factor analysis and the loadings on the higher order factor(s)

`ldingsef` variable loadings on the lower order factors and their correlations with the higher order factor(s)

`extsef` extension variable correlations with the lower order factor(s) and their correlations with the higher order factor(s)

Author(s)

Brian P. O'Connor

References

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.

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.

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

```
## Not run:
EXTENSION_FA(data_RSE, Ncore=7, Next=3, higherorder=TRUE, roottest='MAP',
             corkind='pearson', extract='PCA', rotate='promax', Nfacts=4,
             NfactsHO=1, Ndatasets=100, percentile=95, salvalue=.40, numsals=3,
             iterpaf=200, iterml=30, tolerml=.001, ppower=4)

EXTENSION_FA(data_NEOPIR, Ncore=12, Next=6, higherorder=TRUE, roottest='MAP',
             corkind='pearson', extract='PCA', rotate='promax', Nfacts=4,
             NfactsHO=1, Ndatasets=100, percentile=95, salvalue=.40, numsals=3,
             iterpaf=200, iterml=30, tolerml=.001, ppower=4)


## End(Not run)
```

FACTORABILITY — *The factorability of a correlation matrix*

Description

Three methods for assessing the factorability of a correlation matrix

Usage

```
FACTORABILITY(data, corkind='pearson', Ncases=NULL, display=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', 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.
`display`	Display the results? The options are TRUE (the 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')
```

IMAGE_FA *image factor analysis*

Description

image factor analysis

Usage

```
IMAGE_FA(data, corkind, nfactors, rotate, ppower, 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.
`corkind`	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'. Required only if the entered data is not a correlation matrix.
`nfactors`	The number of factors to extract.
`rotate`	The factor rotation method. The options are: 'promax', 'varimax', and 'none'.
`ppower`	The power value to be used in a promax rotation (required only if rotate = 'promax'). Suggested value: 3
`display`	Display the results? The options are TRUE or FALSE.

Value

A list with the following elements:

`eigenvalues`	The eigenvalues
`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)

Author(s)

Brian P. O'Connor

Examples

```
IMAGE_FA(data_NEOPIR, corkind='pearson', nfactors=5, rotate='varimax', ppower=3, display=TRUE)
```

LOCALDEP *Provides the residual correlations after partialling the first component out of a correlation matrix.*

Description

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. The LOCALDEP function partials out the first component (not the IRT latent trait) from a correlation matrix. Examining the residual correlations is a preliminary, exploratory method of determining whether local dependence exists. The function also displays the number of residual correlations that are >= a range of values.

Usage

```
LOCALDEP(data, corkind, 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.

`corkind` The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'. Required only if the entered data is not a correlation matrix.

`display` Display the results? The options are TRUE or FALSE.

Value

A list with the following elements:

`correlations` The corrrelation matrix

`residcor` The residualized corrrelation matrix

Author(s)

Brian P. O'Connor

Examples

```
# Residual correlations for the Rosenberg Self-Esteem Scale (RSE)
LOCALDEP(data_RSE, corkind = 'pearson', display=TRUE)
```

MAP *Velicer's minimum average partial (MAP) test for number of factors*

Description

Velicer's minimum average partial (MAP) test for determining the number of factors 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).

The MAP test is often more appropriate for factor analyses than it is for principal components analyses. In Velicer's words, "Component analysis has a variety of purposes. It can be used to find a parsimonious description of the total variance of the variables involved; in this case, the [MAP test] is not applicable. Principal component analysis is frequently used to express the variance shared among variables in a set; that is, it is used as kind of a factor analysis" (1976, p. 321). "... if component analysis is employed as an alternative to factor analysis or as a first-stage solution for factor analysis, the stopping rule proposed here would seem the most appropriate." (1976, p. 326)'

Usage

```
MAP(data, corkind, 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.
`corkind`	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'. Required only if the entered data is not a correlation matrix.
`display`	Display the results? The options are TRUE or FALSE.

Value

A list with the following elements:

`eigenvalues`	eigenvalues
`avgsqrs`	Velicer's average squared correlations
`nfMAP`	number of factors according to the original (1976) MAP test
`nfMAP4`	number of factors according to the revised (2000) MAP test

Author(s)

Brian P. O'Connor

References

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.

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.

Examples

```
# MAP test on the Harman correlation matrix
MAP(data_Harman, corkind='pearson', display=TRUE)

## Not run:
# MAP test on the Rosenberg Self-Esteem Scale (RSE) using Pearson correlations
MAP(data_RSE, corkind='pearson', display=TRUE)

# MAP test on the Rosenberg Self-Esteem Scale (RSE) using polychoric correlations
MAP(data_RSE, corkind='polychoric', display=TRUE)

# MAP test on the NEO-PI-R data
MAP(data_NEOPIR, display=TRUE)

## End(Not run)
```

`MAXLIKE_FA` *maximum likelihood factor analysis*

Description

maximum likelihood factor analysis

Usage

```
MAXLIKE_FA(data, corkind, nfactors, tolerml, iterml, rotate, ppower, 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.
`corkind`	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'. Required only if the entered data is not a correlation matrix.
`nfactors`	The number of factors to extract.
`tolerml`	The tolerance level.
`iterml`	The maximum number of iterations.
`rotate`	The factor rotation method. The options are: 'promax', 'varimax', and 'none'.
`ppower`	The power value to be used in a promax rotation (required only if rotate = 'promax'). Suggested value: 3
`display`	Display the results? The options are TRUE or FALSE.

Value

A list with the following elements:

`eigenvalues`	The eigenvalues
`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)

Author(s)

Brian P. O'Connor

Examples

```
MAXLIKE_FA(data_RSE, corkind='pearson', nfactors = 2,
          tolerml = .001, iterml = 50, rotate='promax', ppower=3, display=TRUE)
```

NEVALSGT1 *The number of eigenvalues greater than 1 in a correlation matrix.*

Description

This function 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.

The rationale 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.

Usage

```
NEVALSGT1(data, corkind, 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.
`corkind`	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'. Required only if the entered data is not a correlation matrix.
`display`	Display the eigenvalues and the number that are greater than one? The options are TRUE or FALSE.

Value

The number of eigenvalues greater than 1.

Author(s)

Brian P. O'Connor

References

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

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

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.

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.

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

```
## Not run:
# test on the Harman correlation matrix
NEVALSGT1(data_Harman, corkind='pearson', display=TRUE)

# test on the Rosenberg Self-Esteem Scale (RSE) using Pearson correlations
NEVALSGT1(data_RSE, corkind='pearson', display=TRUE)

# test on the Rosenberg Self-Esteem Scale (RSE) using polychoric correlations
NEVALSGT1(data_RSE, corkind='polychoric', display=TRUE)

## End(Not run)
```

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

Description

This function generates eigenvalues for random data sets with specified numbers of variables and cases. Typically, the eigenvalues derived from an actual data set are compared to the eigenvalues derived from the random data. In Horn's original description of this procedure, the mean eigenvalues from the random data served as the comparison baseline, whereas the more common current practice is to use the eigenvalues that correspond to the desired percentile (typically the 95th) of the distribution of random data eigenvalues. Factors or components are retained as long as the ith eigenvalue from the actual data is greater than the ith eigenvalue from the random data. 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.

Usage

```
PARALLEL(Nvars, Ncases, Ndatasets=100, extract='PCA', percentile='95',
         corkind='pearson', display=TRUE)
```

Arguments

`Nvars` The number of variables.

`Ncases` The number of cases.

`Ndatasets` An integer indicating the # of random data sets for parallel analyses.

`extract` 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'.

`display` Display the results? The options are TRUE or FALSE.

Details

Although the PARALLEL function permits users to specify PCA or PAF or image as the factor extraction method, users should be aware of an unresolved issue in the literature. 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. But others believe that this common practice is wrong. 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 issue remains neglected and unsettled. The PAF option in the extract argument for this function was included for research purposes. It is otherwise probably best to use PCA as the extraction method for regular data analyses. The MAP test (also in this package) is probably more suitable for determining the number of common factors.

Value

The 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.

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

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.

Examples

```
## Not run:
PARALLEL(Nvars=15, Ncases=250, Ndatasets=100, extract='PCA', percentile=95,
        corkind='pearson', display=TRUE)

## End(Not run)
```

PA_FA *principal axis (common) factor analysis*

Description

principal axis (common) factor analysis with squared multiple correlations as the initial communality estimates

Usage

```
PA_FA(data, corkind, nfactors, iterpaf, rotate, ppower, 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.
corkind	The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'. Required only if the entered data is not a correlation matrix.
nfactors	The number of factors to extract.
iterpaf	The maximum number of iterations.
rotate	The factor rotation method. The options are: 'promax', 'varimax', and 'none'.
ppower	The power value to be used in a promax rotation (required only if rotate = 'promax'). Suggested value: 3
display	Display the results? The options are TRUE or FALSE.

Value

A list with the following elements:

eigenvalues	The eigenvalues
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)

Author(s)

Brian P. O'Connor

Examples

```
PA_FA(data_RSE, corkind=”pearson”, nfactors = 2, iterpaf = 50,
      rotate='promax', ppower=3, display=”yes”)
```

PCA *principal components analysis*

Description

principal components analysis

Usage

```
PCA(data, corkind, nfactors, rotate, ppower, 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.

`corkind` The kind of correlation matrix to be used if data is not a correlation matrix. The options are ’pearson’, ’kendall’, ’spearman’, and ’polychoric’. Required only if the entered data is not a correlation matrix.

`nfactors` The number of components to extract.

`rotate` The factor rotation method. The options are: ’promax’, ’varimax’, and ’none’.

`ppower` The power value to be used in a promax rotation (required only if rotate = ’promax’). Suggested value: 3

`display` Display the results? The options are TRUE or FALSE.

Value

A list with the following elements:

`eigenvalues` The eigenvalues

`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)

Author(s)

Brian P. O’Connor

Examples

```
PCA(data_RSE, corkind='pearson', nfactors=2, rotate='promax', ppower=3, display=TRUE)
```

POLYCHORIC_R *polychoric correlation matrix*

Description

This function produces a polychoric correlation matrix

Usage

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

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.

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.

The hetcor function from the polycor package requires a dataframe as input. It also "computes a heterogenous correlation matrix, consisting of Pearson product-moment correlations between numeric variables, polyserial correlations between numeric and ordinal variables, and polychoric correlations between ordinal variables." This means that polychoric correlations will not be computed if a variable is numeric. A numeric variable must first be converted to an ordinal variable (ordered factor), by the user, for the function to produce polychoric correlations for that variable.'

Value

The polychoric correlation matrix

Author(s)

Brian P. O'Connor

Examples

```
## Not run:
# polychoric correlation matrix for the Rosenberg Self-Esteem Scale (RSE)
Rpoly <- POLYCHORIC_R(data_RSE, method = 'Fox')
Rpoly

## End(Not run)
```

PROCRUSTES *factor solution congruence*

Description

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).

Usage

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

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.
`display`	Display the results? The options are TRUE 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
loadings <- PCA(data_RSE[1:150,],   nfactors = 2, rotate='varimax', display=FALSE)
target   <- PCA(data_RSE[151:300,], nfactors = 2, rotate='varimax', display=FALSE)

PROCRUSTES(loadings$loadingsROT, target$loadingsROT, type = 'orthogonal', display=TRUE)
```

PROMAX *promax rotation*

Description

promax rotation

Usage

```
PROMAX(loadings, ppower, display)
```

Arguments

`loadings`	A loading matrix.
`ppower`	The exponent for the promax target matrix. 'ppower' must be 1 or greater. '4' is a conventional value.
`display`	Display the results? The options are TRUE or FALSE.

Value

A list with the following elements:

`structure`	The structure matrix (for promax rotation)
`pattern`	The pattern matrix (for promax rotation)
`correls`	The correlations between the factors (for promax rotation)

Author(s)

Brian P. O'Connor

Examples

```
## Not run:
loadings <- PCA(data_NEOPIR, corkind='pearson', nfactors = 5, rotate='none', display=TRUE)

PROMAX(loadings, ppower = 3, display=TRUE)

## End(Not run)
```

RAWPAR *parallel analysis of eigenvalues (with real data as input)*

Description

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

Usage

```
RAWPAR(data, randtype, extract, Ndatasets, percentile,
       corkind, corkindRAND, Ncases, 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.

`randtype` The kind of random data to be used in the parallel analysis: 'generated' for random normal data generation; 'permuted' for perumatations of the raw data matrix.

`extract` 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', 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', 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.

`display` Display the results? The options are TRUE or FALSE.

Details

Although the RAWPAR function permits users to specify PCA or PAF as the factor extraction method, users should be aware of an unresolved issue in the literature. 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. But others believe that this common practice is wrong. 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 two 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 the other extraction method. The issue remains neglected and unsettled. The PAF option in the extract argument for this function was included for research purposes. It is otherwise probably best to use PCA as the extraction method for regular data analyses. The MAP test (also in this package) is probably more suitable for determining the number of common factors.

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 should probably 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 for research purposes.

Value

A list with:

`eigenvalues`	the eigenvalues for the real and random data
`nfPA`	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.

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

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.

Examples

```
## Not run:
# parallel analysis of the WISC data
RAWPAR(data_WISC, randtype='generated', extract='PCA', Ndatasets=100,
       percentile=95, corkind='pearson', display=TRUE)

# parallel analysis of the Harman correlation matrix
RAWPAR(data_Harman, randtype='generated', extract='PCA', Ndatasets=100,
       percentile=95, corkind='pearson', Ncases=305, display=TRUE)

# parallel analysis of the Rosenberg Self-Esteem Scale (RSE)
RAWPAR(data_RSE, randtype='permuted', extract='PCA', Ndatasets=100,
       percentile=95, corkind='pearson', corkindRAND='pearson', display=TRUE)

# parallel analysis of the Rosenberg Self-Esteem Scale (RSE) using polychoric correlations
RAWPAR(data_RSE, randtype='generated', extract='PCA', Ndatasets=100,
       percentile=95, corkind='polychoric', display=TRUE)

# parallel analysis of the NEO-PI-R data
RAWPAR(data_NEOPIR, randtype='generated', extract='PCA', Ndatasets=100,
       percentile=95, corkind='pearson', Ncases=305, display=TRUE)

## End(Not run)
```

`ROOTFIT` *factor fit coefficients*

Description

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

Usage

```
ROOTFIT(data, corkind, Ncases, extract, 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', 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.

`extract` 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` Display descriptions of the fit coefficients? The options are 'TRUE' (default) or 'FALSE'.

Value

A list with eigenvalues & fit coefficients.

Author(s)

Brian P. O'Connor

Examples

```
# RSE data
ROOTFIT(data_RSE, corkind='pearson', extract='ML')
ROOTFIT(data_RSE, corkind='pearson', extract='PCA', verbose = 'FALSE')

## Not run:
# NEO-PI-R data
ROOTFIT(data_NEOPIR, corkind='pearson', extract='ML')
ROOTFIT(data_NEOPIR, corkind='pearson', extract='PCA', verbose = 'FALSE')

## End(Not run)
```

SALIENT *The salient loadings criterion for determing the number of factors.*

Description

This is a procedure for determining the number of factors 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, 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.

`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', and 'polychoric'. Required only if the entered data is not a correlation matrix.

`display` Display the loadings? The options are TRUE or FALSE.

Value

The number of factors according to the salient loadings criterion.

Author(s)

Brian P. O'Connor

References

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

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.

Examples

```
# test on the Harman correlation matrix
SALIENT(data_Harman, salvalue=.4, numsals=3, corkind='pearson', display=TRUE)

## Not run:
# test on the Rosenberg Self-Esteem Scale (RSE) using Pearson correlations
SALIENT(data_RSE, salvalue=.4, numsals=3, corkind='pearson', display=TRUE)

# test on the Rosenberg Self-Esteem Scale (RSE) using polychoric correlations
SALIENT(data_RSE, salvalue=.4, numsals=3, corkind='polychoric', display=TRUE)

## End(Not run)
```

SESCREE *Standard Error Scree test for the number of components.*

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, corkind, 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.

`corkind` The kind of correlation matrix to be used if data is not a correlation matrix. The options are 'pearson', 'kendall', 'spearman', and 'polychoric'. Required only if the entered data is not a correlation matrix.

`display` Display the eigenvalues, slopes, SE estimates, & the number of components? The options are TRUE 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

```
# test on the Harman correlation matrix
SESCREE(data_Harman, corkind='pearson', display=TRUE)

## Not run:
# test on the Rosenberg Self-Esteem Scale (RSE) using Pearson correlations
SESCREE(data_RSE, corkind='pearson', display=TRUE)

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

## End(Not run)
```

`VARIMAX` *varimax rotation*

Description

varimax rotation

Usage

```
VARIMAX(loadings, display)
```

Arguments

`loadings`	A loading matrix.
`display`	Display the results? The options are TRUE or FALSE.

Value

The varimax-rotated loadings

Author(s)

Brian P. O'Connor

Examples

```
## Not run:
loadings <- PCA(data_NEOPIR, corkind='pearson', nfactors = 5, rotate='none', display=TRUE)

VARIMAX(loadings, display=TRUE)

## End(Not run)
```

Index