Package ‘paramap’

September 20, 2017

Type Package
Title paramap
Version 1.4
Date 2017-09-20
Author Brian P. O'Connor
Maintainer Brian P. O'Connor <brian.oconnor@ubc.ca>
Depends R(>= 1.9.0), psych, polycor
Description Factor analysis functions for assessing dimensionality.
License GPL (>= 2)
LazyLoad yes
LazyData yes
NeedsCompilation no

R topics documented:

paramap-package .............................................. 2
Harman ...................................................... 2
imagefa ....................................................... 3
localdep ..................................................... 3
map .......................................................... 4
maxlike ....................................................... 6
paf .......................................................... 7
parallel ...................................................... 8
pca ........................................................... 9
polychoricR .................................................. 10
promax ....................................................... 11
rawpar ....................................................... 12
rootfit ...................................................... 14
RSE ......................................................... 15
varimax ...................................................... 15
wiscsem ..................................................... 16

Index 17
This package provides three factor analysis-based functions for assessing dimensionality: parallel (for parallel analysis), map (for Velicer’s Minimum Average Partial test), and rootfit (for the fit coefficients for the possible N-factor solutions). 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. The localdep function provides the residual correlations after partialling the first component out of a correlation matrix.

There are also functions for conducting principal components analysis, principal axis factor analysis, maximum likelihood factor analysis, and image 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 and promax rotation can also be performed.

**Description**

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

**Usage**

```r
data(Harman)
```

**Examples**

```r
head(Harman)

# MAP test on the Harman correlation matrix
map(Harman, corkind='pearson', display = 'yes')

# parallel analysis of the Harman correlation matrix
rawpar(Harman, extract='PCA', Ndatasets=100, percentile=95, corkind='pearson', Ncases=305, display='yes')
```
Description

image factor analysis

Usage

imagefa( 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 'yes' or 'no'.

Value

returns a list with eigenvalues & image factor loadings for the specified number of factors

Author(s)

Brian P. O'Connor

Examples

imagefa( RSE, corkind='pearson', nfactors=2, rotate='varimax', ppower=3, display='yes' )

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.

**Usage**

```r
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 'yes' or 'no'.

**Value**

Returns a list with the residual correlation matrix and eigenvalues.

**Author(s)**

Brian P. O'Connor

**Examples**

```r
# Residual correlations for the Rosenberg Self-Esteem Scale (RSE)
localdep(RSE, corkind = 'pearson', display = 'yes')
```

---

**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

```r
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 'yes' or 'no'.

Value

A list with the following values:

- `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


**Examples**

```r
# MAP test on the Harman correlation matrix
map( Harman, corkind='pearson', display = 'yes' )

# MAP test on the Rosenberg Self-Esteem Scale (RSE)
# using Pearson correlations
map( RSE, corkind='pearson', display = 'yes' )

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

---

**maxlike**

*maximum likelihood factor analysis*

**Description**

maximum likelihood factor analysis

**Usage**

```r
maxlike( 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 'yes' or 'no'.

**Value**

returns a list with eigenvalues, communalities, & maximum likelihood factor loadings for the specified number of factors
**paf**

**Author(s)**

Brian P. O'Connor

**Examples**

```r
maxlike( RSE, corkind="pearson", nfactors = 2,
    tolerml = .001, iterml = 50, rotate='promax', ppower=3, display = 'yes' )
```

---

**paf**

*principal axis (common) factor analysis*

**Description**

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

**Usage**

```r
paf( 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 'yes' or 'no'.

**Value**

returns a list with eigenvalues & with the paf loadings on the specified number of components

**Author(s)**

Brian P. O'Connor

**Examples**

```r
paf( RSE, corkind="pearson", nfactors = 2, iterpaf = 50, rotate='promax', ppower=3, display = "yes" )
```
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='yes' )

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 'yes' or 'no'.

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


Examples

parallel( Nvars=15, Ncases=250, Ndatasets=100, extract='PCA', percentile=95, corkind='pearson', 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 'yes' or 'no'.
Value
returns a list with eigenvalues & principal component loadings on the specified number of components

Author(s)
Brian P. O’Connor

Examples
pca( RSE, corkind='pearson', nfactors=2, rotate='promax', ppower=3, display='yes' )

Description
This function produces a polychoric correlation matrix

Usage
polychoricR(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 often produces 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 heterogeneous 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**

returns a polychoric correlation matrix

**Author(s)**

Brian P. O’Connor

**Examples**

```r
# polychoric correlation matrix for the Rosenberg Self-Esteem Scale (RSE)
Rpoly <- polychoricR(RSE)
Rpoly

# polychoric correlation matrix for the RSE using the Fox polycor package
RSE2 = as.data.frame(RSE) # convert RSE to a dataframe
# convert 1st 5 variables to ordered factors, for illustration
for (j in 1:5) {RSE2[, j] <- ordered(RSE2[, j])}
Rpoly <- polychoricR(RSE2, method = 'Fox')
Rpoly
```

**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 'yes' or 'no'.

Value

returns a list with the promax-rotated structure loadings, pattern loadings, and factor correlations

Author(s)

Brian P. O’Connor

Examples

loadings <- pca( RSE, corkind=’pearson’, nfactors = 2, rotate=’none’, display = ’yes’ )
promax( loadings, ppower = 3, display = ’yes’ )

Description

This 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; ’permutated’ for permutatations 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
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.

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'.

The number of cases upon which a correlation matrix is based. Required only if data is a correlation matrix.

Display the results? The options are 'yes' or 'no'.

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.

A list with:

- eigenvalues: the eigenvalues for the real and random data
- nfPA: the number of factors based on the parallel analysis

Brian P. O'Connor


Examples

# parallel analysis of the wiscsem data
rawpar( wiscsem, randtype='generated', extract='PCA', Ndatasets=100, percentile=95, corkind='pearson', display='yes' )

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

# parallel analysis of the Rosenberg Self-Esteem Scale (RSE)
# using Pearson correlations
rawpar( RSE, randtype='permuted', extract='PCA', Ndatasets=100, percentile=95, corkind='pearson', corkindRAND='pearson', display='yes' )

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

rootfit

factor fit coefficients

Description

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

Usage

rootfit( data, corkind, 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.

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'.
**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(RSE)

**Examples**  
head(RSE)

# MAP test on the Rosenberg Self-Esteem Scale (RSE) data  
map( RSE, corkind='pearson', display = 'yes' )

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

**varimax**  

*varimax rotation*

**Description**  
varimax rotation

**Usage**  
varimax( loadings, display )

**Arguments**  
loadings  
A loading matrix.

display  
Display the results? The options are 'yes' or 'no'.

**Value**  
returns a list with eigenvalues & fit coefficients

**Author(s)**  
Brian P. O’Connor

**Examples**  
rootfit( RSE, corkind='pearson', extract='ML' )

rootfit( RSE, corkind='pearson', extract='PCA', verbose = 'FALSE' )
Value
returns the varimax-rotated loadings

Author(s)
Brian P. O’Connor

Examples
loadings <- pca( RSE, corkind='pearson', nfactors = 2, rotate='none', display = 'no' )
varimax( loadings$loadings, display = 'yes' )

Description
A data frame with scores for 175 cases on 10 WISC-R subscales, from Tabacknick & Fidell, 3rd ed., used for their section on confirmatory factor analysis.

Usage
data(wiscsem)

Examples
head(wiscsem)

# MAP test on the wiscsem data
map( wiscsem, corkind='pearson', display = 'yes' )

# parallel analysis of the wiscsem data
rawpar( wiscsem, extract='PCA', Ndatasets=100, percentile=95,
corkind='pearson', display='yes' )