Package 'nFactors'

December 18, 2011

Type Package

Title Parallel Analysis and Non Graphical Solutions to the Cattell Scree Test

Version 2.3.3

Date 2010-04-10

Encoding latin1

Author Gilles Raiche (Universite du Quebec a Montreal) and David Magis (Universite de Liege)

Maintainer Gilles Raiche <raiche.gilles@uqam.ca>

Depends R (>= 2.9.2), MASS, psych, boot, lattice

Suggests xtable

Description Indices, heuristics and strategies to help determine the number of factors/components to retain: 1. Acceleration factor (af with or without Parallel Analysis); 2. Optimal Coordinates (noc with or without Parallel Analysis); 3. Parallel analysis (components, factors and bootstrap); 4. lambda > mean(lambda) (Kaiser, CFA and related); 5. Cattell-Nelson-Gorsuch (CNG); 6. Zoski and Jurs multiple regression (b, t and p); 7. Zoski and Jurs standard error of the regression coeffcient (sescree); 8. Nelson R2; 9. Bartlett khi-2; 10. Anderson khi-2; 11. Lawley khi-2 and 12. Bentler-Yuan khi-2.

License GPL (version 2 or later)

R topics documented:

`nFactors-package` *Parallel Analysis and Non Graphical Solutions to the Cattell Scree Test*

Description

Indices, heuristics and strategies to help determine the number of factors/components to retain:

1. - Acceleration factor (*noc* with or without Parallel Analysis)
2. - Optimal Coordinates (*noc* with or without Parallel Analysis)
3. - Parallel analysis (components, factors and bootstrap)
4. - $\lambda >= \bar{\lambda}$ (Kaiser, CFA and related rule)
5. - Cattell-Nelson-Gorsuch (*CNG*)
6. - Zoski and Jurs Multiple regression (β, *t* and *p*)
7. - Zoski and Jurs standard error of the regression coefficient (sescree, $S_{Y \bullet X}$)
8. - Nelson R^2
9. - Bartlett χ^2
10. - Anderson χ^2
11. - Lawley χ^2 and
12. - Bentler-Yuan χ^2.

Details

Package:	nFactors
Type:	Package
Version:	2.3.2
Date:	2010-04-10
Depends:	R (>= 2.9.2), MASS, psych, boot
License:	GPL

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

David Magis
Departement de mathematiques
Universite de Liege
`<David.Magis@ulg.ac.be>`

References

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

See Also

Other packages are also very useful for principal component and factor analysis. The *R* psychometric view is instructive at this point. See http://cran.stat.sfu.ca/web/views/Psychometrics.html for further details.

`bentlerParameters` *Bentler and Yuan's Computation of the LRT Index and the Linear Trend Coefficients*

Description

This function computes the Bentler and Yuan's (1996, 1998) *LRT* index for the linear trend in eigenvalues of a covariance matrix. The related χ^2 and *p*-value are also computed. This function is generally called from the `nBentler` function. But it could be of use for graphing the linear trend function and to study it's behavior.

Usage

```
bentlerParameters(x, N, nFactors, log=TRUE, cor=TRUE,
                  minPar=c(min(lambda) - abs(min(lambda)) + .001,
                           0.001),
                  maxPar=c(max(lambda),
                           lm(lambda~I(length(lambda):1))$coef[2]),
                  resParx=c(0.01, 2), resPary=c(0.01, 2),
                  graphic=TRUE, resolution=30,
                  typePlot="wireframe", ...)
```

Arguments

`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data
`N`	numeric: number of subjects.
`nFactors`	numeric: number of components to test.
`log`	logical: if `TRUE` the minimization is applied on the log values.
`cor`	logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix
`minPar`	numeric: minimums for the coefficient of the linear trend.
`maxPar`	numeric: maximums for the coefficient of the linear trend.
`resParx`	numeric: restriction on the α coefficient (x) to graph the function to minimize.
`resPary`	numeric: restriction on the β coefficient (y) to graph the function to minimize.
`graphic`	logical: if `TRUE` plots the minimized function `"wireframe"`, `"contourplot"` or `"levelplot"`.
`resolution`	numeric: resolution of the 3D graph (number of points from α and from β).
`typePlot`	character: plots the minimized function according to a 3D plot: `"wireframe"`, `"contourplot"` or `"levelplot"`.
`...`	variable: additionnal parameters from the `"wireframe"`, `"contourplot"` or `"levelplot"` `lattice` functions. Also additionnal parameters for the `eigenFrom` function.

Details

The implemented Bentler and Yuan's procedure must be used with care because the minimized function is not always stable. In many cases, constraints must applied to obtain a solution. The actual implementation did, but the user can modify these constraints.

The hypothesis tested (Bentler and Yuan, 1996, equation 10) is:

(1) $$H_k : \lambda_{k+i} = \alpha + \beta x_i, (i = 1, \ldots, q)$$

The solution of the following simultaneous equations is needed to find $(\alpha, \beta) \in$

(2) $$f(x) = \sum_{i=1}^{q} \frac{[\lambda_{k+j} - N\alpha + \beta x_j]x_j}{(\alpha + \beta x_j)^2} = 0$$

and $$g(x) = \sum_{i=1}^{q} \frac{\lambda_{k+j} - N\alpha + \beta x_j x_j}{(\alpha + \beta x_j)^2} = 0$$

The solution to this system of equations was implemented by minimizing the following equation:

(3) $$(\alpha, \beta) \in \inf\left[h(x)\right] = \inf \log\left[f(x)^2 + g(x)^2\right]$$

The likelihood ratio test LRT proposed by Bentler and Yuan (1996, equation 7) follows a χ^2 probability distribution with $q - 2$ degrees of freedom and is equal to:

(4) $$LRT = N(k-p)\left\{\ln\left(\frac{n}{N}\right) + 1\right\} - N \sum_{j=k+1}^{p} \ln\left\{\frac{\lambda_j}{\alpha + \beta x_j}\right\} + n \sum_{j=k+1}^{p} \left\{\frac{\lambda_j}{\alpha + \beta x_j}\right\}$$

With p beeing the number of eigenvalues, k the number of eigenvalues to test, q the $p - k$ remaining eigenvalues, N the sample size, and $n = N - 1$. Note that there is an error in the Bentler and Yuan equation, the variables N and n beeing inverted in the preceeding equation 4.

A better strategy proposed by Bentler an Yuan (1998) is to use a minimized χ^2 solution. This strategy will be implemented in a future version of the **nFactors** package.

Value

`nFactors`	numeric: vector of the number of factors retained by the Bentler and Yuan's procedure.
`details`	numeric: matrix of the details of the computation.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

David Magis
Departement de mathematiques
Universite de Liege
<David.Magis@ulg.ac.be>

References

Bentler, P. M. and Yuan, K.-H. (1996). Test of linear trend in eigenvalues of a covariance matrix with application to data analysis. *British Journal of Mathematical and Statistical Psychology, 49*, 299-312.

Bentler, P. M. and Yuan, K.-H. (1998). Test of linear trend in the smallest eigenvalues of the correlation matrix. *Psychometrika, 63*(2), 131-144.

See Also

`nBartlett`, `nBentler`

Examples

```
## ................................................
## SIMPLE EXAMPLE OF THE BENTLER AND YUAN PROCEDURE

# Bentler (1996, p. 309) Table 2 - Example 2 .............
n=649
bentler2<-c(5.785, 3.088, 1.505, 0.582, 0.424, 0.386, 0.360, 0.337, 0.303,
            0.281, 0.246, 0.238, 0.200, 0.160, 0.130)

results  <- nBentler(x=bentler2, N=n,  details=TRUE)
results

# Two different figures to verify the convergence problem identified with
# the 2th component
bentlerParameters(x=bentler2, N=n, nFactors= 2, graphic=TRUE,
                  typePlot="contourplot",
                  resParx=c(0,9), resPary=c(0,9), cor=FALSE)

bentlerParameters(x=bentler2, N=n, nFactors= 4, graphic=TRUE, drape=TRUE,
                  resParx=c(0,9), resPary=c(0,9),
                  scales = list(arrows = FALSE) )

plotuScree(x=bentler2, model="components",
  main=paste(results$nFactors,
  " factors retained by the Bentler and Yuan's procedure (1996, p. 309)",
  sep=""))
# ........................................................

# Bentler (1998, p. 140) Table 3 - Example 1 .............
n        <- 145
example1 <- c(8.135, 2.096, 1.693, 1.502, 1.025, 0.943, 0.901, 0.816,
              0.790,0.707, 0.639, 0.543,0.533, 0.509, 0.478, 0.390,
              0.382, 0.340, 0.334, 0.316, 0.297,0.268, 0.190, 0.173)

results  <- nBentler(x=example1, N=n,  details=TRUE)
results

# Two different figures to verify the convergence problem identified with
# the 10th component
bentlerParameters(x=example1, N=n, nFactors= 10, graphic=TRUE,
                  typePlot="contourplot",
                  resParx=c(0,0.4), resPary=c(0,0.4))

bentlerParameters(x=example1, N=n, nFactors= 10, graphic=TRUE, drape=TRUE,
                  resParx=c(0,0.4), resPary=c(0,0.4),
                  scales = list(arrows = FALSE) )

plotuScree(x=example1, model="components",
   main=paste(results$nFactors,
   " factors retained by the Bentler and Yuan's procedure (1998, p. 140)",
   sep=""))
# ........................................................
```

`componentAxis` *Principal Component Analysis With Only n First Components Retained*

Description

The `componentAxis` function returns a principal component analysis with the first *n* components retained.

Usage

```
componentAxis(R, nFactors=2)
```

Arguments

`R`	numeric: correlation or covariance matrix
`nFactors`	numeric: number of components/factors to retain

Value

`values`	numeric: variance of each component/factor retained
`varExplained`	numeric: variance explained by each component/factor retained
`varExplained`	numeric: cumulative variance explained by each component/factor retained
`loadings`	numeric: loadings of each variable on each component/factor retained

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, `http://www.er.uqam.ca/nobel/r17165/`

References

Kim, J.-O. and Mueller, C. W. (1978). *Introduction to factor analysis. What it is and how to do it*. Beverly Hills, CA: Sage.

Kim, J.-O. and Mueller, C. W. (1987). *Factor analysis. Statistical methods and practical issues*. Beverly Hills, CA: Sage.

See Also

`principalComponents`, `iterativePrincipalAxis`, `rRecovery`

Examples

```
# ......................................................
# Example from Kim and Mueller (1978, p. 10)
# Simulated sample: lower diagnonal
 R <- matrix(c( 1.000, 0.560, 0.480, 0.224, 0.192, 0.16,
                0.560, 1.000, 0.420, 0.196, 0.168, 0.14,
                0.480, 0.420, 1.000, 0.168, 0.144, 0.12,
                0.224, 0.196, 0.168, 1.000, 0.420, 0.35,
                0.192, 0.168, 0.144, 0.420, 1.000, 0.30,
                0.160, 0.140, 0.120, 0.350, 0.300, 1.00),
                nrow=6, byrow=TRUE)

# Factor analysis: Selected principal components - Kim and Mueller
# (1978, p. 20)
 componentAxis(R, nFactors=2)

# ......................................................
```

---

corFA — *Insert Communalities in the Diagonal of a Correlation or a Covariance Matrix*

Description

This function inserts communalities in the diagonal of a correlation/covariance matrix.

Usage

```
corFA(R, method="ginv")
```

Arguments

`R`	numeric: correlation matrix.
`method`	character: actually only `"giv"` is supplied to compute the approximation of the communalities by maximum correlation.

Value

`values`	numeric: matrix of correlation/covariance with communalities in the diagonal.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

See Also

`plotuScree`, `nScree`, `plotnScree`, `plotParallel`

Examples

```
## LOWER CORRELATION MATRIX WITH ZEROS ON UPPER PART
## From Gorsuch (table 1.3.1)
 gorsuch <- c(
 1,0,0,0,0,0,0,0,0,0,
 .6283, 1,0,0,0,0,0,0,0,0,
 .5631, .7353, 1,0,0,0,0,0,0,0,
 .8689, .7055, .8444, 1,0,0,0,0,0,0,
 .9030, .8626, .6890, .8874, 1,0,0,0,0,0,
 .6908, .9028, .9155, .8841, .8816, 1,0,0,0,0,
 .8633, .7495, .7378, .9164, .9109, .8572, 1,0,0,0,
 .7694, .7902, .7872, .8857, .8835, .8884, .7872, 1,0,0,
 .8945, .7929, .7656, .9494, .9546, .8942, .9434, .9000, 1,0,
 .5615, .6850, .8153, .7004, .6583, .7720, .6201, .6141, .6378, 1)

## UPPER CORRELATION MATRIX FILLED WITH UPPER CORRELATION MATRIX
 gorsuch <- makeCor(gorsuch)

## REPLACE DIAGONAL WITH COMMUNALITIES
 gorsuchCfa <- corFA(gorsuch)
 gorsuchCfa
```

`dFactors` *Eigenvalues Vectors From the Litterature*

Description

Classical examples of eigenvalues vectors used to study the number of factors to retain in the litterature. These examples generally give the number of subjects use to obtain these eigenvalues. The number of subjects is used with the parallel analysis.

Usage

```
data(dFactors)
```

Format

A list of examples. For each example, a list is also used to give the eigenvalues vector and the number of subjects.

Details

Other datasets will be added in future versions of the package.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

Source

Lawley and Hand dataset: Bartholomew *et al.* (2002, p. 123, 126)

Bentler dataset: Bentler and Yuan (1998, p. 139-140)

Buja datasets: Buja and Eyuboglu (1992, p. 516, 519) < Number of subjects not specified by Buja and Eyuboglu >

Cliff datasets: Cliff (1970, p. 165)

Raiche dataset: Raiche, Langevin, Riopel and Mauffette (2006)

Raiche dataset: Raiche, Riopel and Blais (2006, p. 9)

Tucker datasets: Tucker *et al.* (1969, p. 442)

References

Bartholomew, D. J., Steele, F., Moustaki, I. and Galbraith, J. I. (2002). *The analysis and interpretation of multivariate data for social scientists*. Boca Raton, FL: Chapman and Hall.

Bentler, P. M. and Yuan, K.-H. (1998). Tests for linear trend in the smallest eigenvalues of the correlation matrix. *Psychometrika, 63*(2), 131-144.

Buja, A. and Eyuboglu, N. (1992). Remarks on parallel analysis. *Multivariate Behavioral Research, 27*(4), 509-540.

Cliff, N. (1970). The relation between sample and population characteristic vectors. *Psychometrika, 35*(2), 163-178.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J. and Ostrowski, E. (1994). *A handbook of small data sets*. Boca Raton, FL: Chapman and Hall.

Jaejon Song, B. A., Walls, T. A. and Raiche, G. (2008). *Numerical solutions for Cattel's scree test: application to the adolescent smoking consequences questionnaire (ASCQ)*. Paper presented at the International Annual meeting of the Psychometric Society, Durham, New Hamphire. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

Lawley, D. N. and Maxwell, A. E. (1971). *Factor analysis as a statistical method* (2nd edition). London: Butterworth.

Raiche, G., Langevin, L., Riopel, M. and Mauffette, Y. (2006). Etude exploratoire de la dimensionnalite et des facteurs expliques par une traduction francaise de l'Inventaire des approches d'enseignement de Trigwell et Prosser dans trois universite quebecoises. *Mesure et Evaluation en Education, 29*(2), 41-61.

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

Tucker, L. D., Koopman, R. F. and Linn, R. L. (1969). Evaluation of factor analytic research procedures by mean of simulated correlation matrices. *Psychometrika, 34*(4), 421-459.

Zoski, K. and Jurs, S. (1993). Using multiple regression to determine the number of factors to retain in factor analysis. *Multiple Linear Regression Viewpoint, 20*(1), 5-9.

Examples

```
# EXAMPLES FROM DATASET
 data(dFactors)

# COMMAND TO VISUALIZE THE CONTENT AND ATTRIBUTES OF THE DATASETS
 names(dFactors)
 attributes(dFactors)
```

```
 dFactors$Cliff1$eigenvalues
 dFactors$Cliff1$nsubjects

# SCREE PLOT
 plotuScree(dFactors$Cliff1$eigenvalues)
```

`diagReplace` *Replacing Upper or Lower Diagonal of a Correlation or Covariance Matrix*

Description

The `diagReplace` function returns a modified correlation or covariance matrix by replacing upper diagonal with lower diagonal, or lower diagonal with upper diagonal.

Usage

```
diagReplace(R, upper=TRUE)
```

Arguments

`R`	numeric: correlation or covariance matrix
`upper`	logical: if `TRUE` upper diagonal is replaced with lower diagonal. If `FALSE`, lower diagonal is replaced with upper diagonal.

Value

`R`	numeric: correlation or covariance matrix

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

Examples

```
# .......................................................
# Example from Kim and Mueller (1978, p. 10)
# Population: upper diagonal
# Simulated sample: lower diagnonal
 R <- matrix(c( 1.000, .6008, .4984, .1920, .1959, .3466,
                .5600, 1.000, .4749, .2196, .1912, .2979,
                .4800, .4200, 1.000, .2079, .2010, .2445,
                .2240, .1960, .1680, 1.000, .4334, .3197,
                .1920, .1680, .1440, .4200, 1.000, .4207,
                .1600, .1400, .1200, .3500, .3000, 1.000),
                nrow=6, byrow=TRUE)

# Replace upper diagonal with lower diagonal
```

```
 RU <- diagReplace(R, upper=TRUE)

# Replace lower diagonal with upper diagonal
 RL <- diagReplace(R, upper=FALSE)
# ......................................................
```

---

eigenBootParallel *Bootstrapping of the Eigenvalues From a Data Frame*

Description

The `eigenBootParallel` function samples observations from a `data.frame` to produce correlation or covariance matrices from which eigenvalues are computed. The function returns statistics about these bootstrapped eigenvalues. Their means or their quantile could be used later to replace the eigenvalues inputted to a parallel analysis. The `eigenBootParallel` can also compute random eigenvalues from empirical data by column permutation (Buja and Eyuboglu, 1992).

Usage

```
eigenBootParallel(x, quantile=0.95, nboot=30, option="permutation",
                    cor=TRUE, model="components", ...)
```

Arguments

`x`	data.frame: data from which a correlation matrix will be obtained
`quantile`	numeric: eigenvalues quantile to be reported
`nboot`	numeric: number of bootstrap samples
`option`	character: `"permutation"` or `"bootstrap"`
`cor`	logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix (`eigenComputes`)
`model`	character: bootstraps from a principal component analysis (`"components"`) or from a factor analysis (`"factors"`)
`...`	variable: additionnal parameters to give to the `cor` or `cov` functions

Value

`values`	data.frame: mean, median, quantile, standard deviation, minimum and maximum of bootstrapped eigenvalues

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

References

Buja, A. and Eyuboglu, N. (1992). Remarks on parallel analysis. *Multivariate Behavioral Research, 27*(4), 509-540.

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

See Also

principalComponents, iterativePrincipalAxis, rRecovery

Examples

```
# ....................................................
# Example from the iris data
 eigenvalues <- eigenComputes(x=iris[,-5])

# Permutation parallel analysis distribution
 aparallel   <- eigenBootParallel(x=iris[,-5], quantile=0.95)$quantile

# Number of components to retain
 results     <- nScree(x = eigenvalues, aparallel = aparallel)
 results$Components
 plotnScree(results)
# ...................................................

# ...................................................
# Bootstrap distributions study of the eigenvalues from iris data
# with different correlation methods
 eigenBootParallel(x=iris[,-5],quantile=0.05,
                   option="bootstrap",method="pearson")
 eigenBootParallel(x=iris[,-5],quantile=0.05,
                   option="bootstrap",method="spearman")
 eigenBootParallel(x=iris[,-5],quantile=0.05,
                   option="bootstrap",method="kendall")
```

---

eigenComputes — *Computes Eigenvalues According to the Data Type*

Description

The `eigenComputes` function computes eigenvalues from the identified data type. It is used internally in many fonctions of the **nFactors** package in order to apply these to a vector of eigenvalues, a matrix of correlations or covariance or a data frame.

Usage

```
eigenComputes(x, cor=TRUE, model="components", ...)
```

Arguments

x	numeric: a vector of eigenvalues, a matrix of correlations or of covariances or a data.frame of data
cor	logical: if TRUE computes eigenvalues from a correlation matrix, else from a covariance matrix
model	character: "components" or "factors"
...	variable: additionnal parameters to give to the cor or cov functions

Value

value	numeric: return a vector of eigenvalues

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

Examples

```
# .......................................................
# Different data types
# Vector of eigenvalues
  data(dFactors)
  x1 <- dFactors$Cliff1$eigenvalues
  eigenComputes(x1)

# Data from a data.frame
  x2 <- data.frame(matrix(20*rnorm(100), ncol=5))
  eigenComputes(x2, cor=TRUE,  use="everything")
  eigenComputes(x2, cor=FALSE, use="everything")
  eigenComputes(x2, cor=TRUE,  use="everything", method="spearman")
  eigenComputes(x2, cor=TRUE,  use="everything", method="kendall")

# From a covariance matrix
  x3 <- cov(x2)
  eigenComputes(x3, cor=TRUE,  use="everything")
  eigenComputes(x3, cor=FALSE, use="everything")

# From a correlation matrix
  x4 <- cor(x2)
  eigenComputes(x4, use="everything")
# .......................................................
```

---

eigenFrom *Identify the Data Type to Obtain the Eigenvalues*

Description

The `eigenFrom` function identifies the data type from which to obtain the eigenvalues. The function is used internally in many functions of the **nFactors** package to be able to apply these to a vector of eigenvalues, a matrix of correlations or covariance or a `data.frame`.

Usage

```
eigenFrom(x)
```

Arguments

`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data

Value

`value`	character: return the data type to obtain the eigenvalues: `"eigenvalues"`, `"correlation"` or `"data"`

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

Examples

```
# ......................................................
# Different data types
# Examples of adequate data sources
# Vector of eigenvalues
  data(dFactors)
  x1 <- dFactors$Cliff1$eigenvalues
  eigenFrom(x1)

# Data from a data.frame
  x2 <- data.frame(matrix(20*rnorm(100), ncol=5))
  eigenFrom(x2)

# From a covariance matrix
  x3 <- cov(x2)
  eigenFrom(x3)

# From a correlation matrix
  x4 <- cor(x2)
  eigenFrom(x4)

# Examples of inadequate data sources: not run because of errors generated
  # x0 <- c(2,1)              # Error: not enough eigenvalues
  # eigenFrom(x0)
  # x2 <- matrix(x1, ncol=5) # Error: non a symetric covariance matrix
  # eigenFrom(x2)
  # eigenFrom(x3[,(1:2)])     # Error: not enough variables
```

```
  # x6 <- table(x5)          # Error: not a valid data class
  # eigenFrom(x6)
# .....................................................
```

generateStructure *Generate a Factor Structure Matrix.*

Description

The generateStructure function returns a *mjc* factor structure matrix. The number of variables per major factor *pmjc* is equal for each factor. The argument *pmjc* must be divisible by *nVar*. The arguments are strongly inspired from Zick and Velicer (1986, p. 435-436) methodology.

Usage

```
generateStructure(var, mjc, pmjc, loadings, unique)
```

Arguments

var	numeric: number of variables
mjc	numeric: number of major factors (factors with practical significance)
pmjc	numeric: number of variables that load significantly on each major factor
loadings	numeric: loadings on the significant variables on each major factor
unique	numeric: loadings on the non significant variables on each major factor

Value

values	numeric matrix: factor structure

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

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

See Also

principalComponents, iterativePrincipalAxis, rRecovery

Examples

```
# ......................................................
# Example inspired from Zwick and Velicer (1986, table 2, p. 437)
## ...................................................................
 unique=0.2; loadings=0.5
 zwick1 <- generateStructure(var=36, mjc=6, pmjc= 6, loadings=loadings,
                             unique=unique)
 zwick2 <- generateStructure(var=36, mjc=3, pmjc=12, loadings=loadings,
                             unique=unique)
 zwick3 <- generateStructure(var=72, mjc=9, pmjc= 8, loadings=loadings,
                             unique=unique)
 zwick4 <- generateStructure(var=72, mjc=6, pmjc=12, loadings=loadings,
                             unique=unique)
 sat=0.8
## ...................................................................
 zwick5 <- generateStructure(var=36, mjc=6, pmjc= 6, loadings=loadings,
                             unique=unique)
 zwick6 <- generateStructure(var=36, mjc=3, pmjc=12, loadings=loadings,
                             unique=unique)
 zwick7 <- generateStructure(var=72, mjc=9, pmjc= 8, loadings=loadings,
                             unique=unique)
 zwick8 <- generateStructure(var=72, mjc=6, pmjc=12, loadings=loadings,
                             unique=unique)
## ...................................................................

# nsubjects <- c(72, 144, 180, 360)
# require(psych)
# Produce an usual correlation matrix from a congeneric model
 nsubjects <- 72
 mzwick5   <- sim.structure(fx=as.matrix(zwick5), n=nsubjects)
 mzwick5$r

# Factor analysis: recovery of the factor structure
 iterativePrincipalAxis(mzwick5$model, nFactors=6,
                        communalities="ginv")$loadings
 iterativePrincipalAxis(mzwick5$r    , nFactors=6,
                        communalities="ginv")$loadings
 factanal(covmat=mzwick5$model,         factors=6)
 factanal(covmat=mzwick5$r    ,         factors=6)

# Number of components to retain
 eigenvalues  <- eigen(mzwick5$r)$values
 aparallel    <- parallel(var      = length(eigenvalues),
                          subject  = nsubjects,
                          rep      = 30,
                          quantile = 0.95,
                          model="components")$eigen$qevpea
 results <- nScree(x         = eigenvalues,
                   aparallel = aparallel)
 results$Components
 plotnScree(results)

# Number of factors to retain
 eigenvalues.fa  <- eigen(corFA(mzwick5$r))$values
 aparallel.fa    <- parallel(var      = length(eigenvalues.fa),
                             subject  = nsubjects,
```

```
                              rep      = 30,
                              quantile = 0.95,
                              model="factors")$eigen$qevpea
 results.fa <- nScree(x       = eigenvalues.fa,
                    aparallel = aparallel.fa,
                    model     ="factors")
 results.fa$Components
 plotnScree(results.fa)
# ......................................................
```

`iterativePrincipalAxis` *Iterative Principal Axis Analysis*

Description

The `iterativePrincipalAxis` function returns a principal axis analysis with iterated communality estimates. Four different choices of initial communality estimates are given: maximum correlation, multiple correlation (usual and generalized inverse) or estimates based on the sum of the squared principal component analysis loadings. Generally, statistical packages initialize the communalities at the multiple correlation value. Unfortunately, this strategy cannot always deal with singular correlation or covariance matrices. If a generalized inverse, the maximum correlation or the estimated communalities based on the sum of loadings are used instead, then a solution can be computed.

Usage

```
iterativePrincipalAxis(R,
                       nFactors=2,
                       communalities="component",
                       iterations=20,
                       tolerance=0.001)
```

Arguments

`R`	numeric: correlation or covariance matrix
`nFactors`	numeric: number of factors to retain
`communalities`	character: initial values for communalities (`"component"`, `"maxr"`, `"ginv"` or `"multiple"`)
`iterations`	numeric: maximum number of iterations to obtain a solution
`tolerance`	numeric: minimal difference in the estimated communalities after a given iteration

Value

`values`	numeric: variance of each component
`varExplained`	numeric: variance explained by each component

`varExplained`	numeric: cumulative variance explained by each component
`loadings`	numeric: loadings of each variable on each component
`iterations`	numeric: maximum number of iterations to obtain a solution
`tolerance`	numeric: minimal difference in the estimated communalities after a given iteration

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

Kim, J.-O. and Mueller, C. W. (1978). *Introduction to factor analysis. What it is and how to do it*. Beverly Hills, CA: Sage.

Kim, J.-O. and Mueller, C. W. (1987). *Factor analysis. Statistical methods and practical issues*. Beverly Hills, CA: Sage.

See Also

componentAxis, principalAxis, rRecovery

Examples

```
# .......................................................
# Example from Kim and Mueller (1978, p. 10)
# Population: upper diagonal
# Simulated sample: lower diagnonal
 R <- matrix(c( 1.000, .6008, .4984, .1920, .1959, .3466,
                .5600, 1.000, .4749, .2196, .1912, .2979,
                .4800, .4200, 1.000, .2079, .2010, .2445,
                .2240, .1960, .1680, 1.000, .4334, .3197,
                .1920, .1680, .1440, .4200, 1.000, .4207,
                .1600, .1400, .1200, .3500, .3000, 1.000),
                nrow=6, byrow=TRUE)

# Factor analysis: Principal axis factoring with iterated communalities
# Kim and Mueller (1978, p. 23)
# Replace upper diagonal with lower diagonal
 RU          <- diagReplace(R, upper=TRUE)
 nFactors    <- 2
 fComponent <- iterativePrincipalAxis(RU, nFactors=nFactors,
                                       communalities="component")
 fComponent
 rRecovery(RU,fComponent$loadings, diagCommunalities=FALSE)

 fMaxr      <- iterativePrincipalAxis(RU, nFactors=nFactors,
                                       communalities="maxr")
 fMaxr
 rRecovery(RU,fMaxr$loadings, diagCommunalities=FALSE)

 fMultiple  <- iterativePrincipalAxis(RU, nFactors=nFactors,
                                       communalities="multiple")
```

```
 fMultiple
 rRecovery(RU,fMultiple$loadings, diagCommunalities=FALSE)
# .....................................................
```

makeCor *Create a Full Correlation/Covariance Matrix from a Matrix With Lower Part Filled and Upper Part With Zeros*

Description

This function creates a full correlation/covariance matrix from a matrix with lower part filled and upper part with zeros.

Usage

```
makeCor(x)
```

Arguments

`x`	numeric: matrix.

Value

`values`	numeric: full correlation matrix.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

See Also

plotuScree, nScree, plotnScree, plotParallel

Examples

```
## LOWER CORRELATION MATRIX WITH ZEROS ON UPPER PART
## From Gorsuch (table 1.3.1)
 gorsuch <- c(
 1,0,0,0,0,0,0,0,0,0,
 .6283, 1,0,0,0,0,0,0,0,0,
 .5631, .7353, 1,0,0,0,0,0,0,0,
 .8689, .7055, .8444, 1,0,0,0,0,0,0,
 .9030, .8626, .6890, .8874, 1,0,0,0,0,0,
 .6908, .9028, .9155, .8841, .8816, 1,0,0,0,0,
 .8633, .7495, .7378, .9164, .9109, .8572, 1,0,0,0,
 .7694, .7902, .7872, .8857, .8835, .8884, .7872, 1,0,0,
 .8945, .7929, .7656, .9494, .9546, .8942, .9434, .9000, 1,0,
 .5615, .6850, .8153, .7004, .6583, .7720, .6201, .6141, .6378, 1)
```

```
## UPPER CORRELATION MATRIX FILLED WITH UPPER CORRELATION MATRIX
 gorsuch <- makeCor(gorsuch)
 gorsuch
```

---

moreStats *Statistical Summary of a Data Frame*

Description

This function produces another summary of a `data.frame`. This function was proposed in order to apply some functions globally on a `data.frame`: `quantile`, `median`, `min` and `max`. The usual *R* version cannot do so.

Usage

```
moreStats(x, quantile=0.95, show=FALSE)
```

Arguments

`x`	numeric: matrix or `data.frame`.
`quantile`	numeric: quantile of the distribution.
`show`	logical: if `TRUE` prints the quantile choosen.

Value

`values`	numeric: `data.frame` of statistics: mean, median, quantile, standard deviation, minimum and maximum.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

See Also

`plotuScree`, `nScree`, `plotnScree`, `plotParallel`

Examples

```
## GENERATION OF A MATRIX OF 100 OBSERVATIONS AND 10 VARIABLES
 x   <- matrix(rnorm(1000),ncol=10)

## STATISTICS
 res <- moreStats(x, quantile=0.05, show=TRUE)
 res
```

nBartlett *Bartlett, Anderson and Lawley Procedures to Determine the Number of Components/Factors*

Description

This function computes the Bartlett, Anderson and Lawley indices for determining the number of components/factors to retain.

Usage

```
nBartlett(x, N, alpha=0.05, cor=TRUE, details=TRUE, correction=TRUE, ...)
```

Arguments

`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data (eigenFrom)
`N`	numeric: number of subjects
`alpha`	numeric: statistical significance level
`cor`	logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix
`details`	logical: if `TRUE` also returns detains about the computation for each eigenvalue
`correction`	logical: if `TRUE` uses a correction for the degree of freedom after the first eigenvalue
`...`	variable: additionnal parameters to give to the `cor` or `cov` functions

Details

The hypothesis tested is:

$$H_k : \lambda_{k+1} = \ldots = \lambda_p \tag{1}$$

This hypothesis is verified by the application of different version of a χ^2 test with different values for the degrees of freedom. Each of these tests shares the compution of a V_k value:

$$V_k = \prod_{i=k+1}^{p} \left\{ \frac{\lambda_i}{{}^1\!/_q \sum_{i=k+1}^{p} \lambda_i} \right\} \tag{2}$$

p is the number of eigenvalues, k the number of eigenvalues to test, and q the $p-k$ remaining eigenvalues. n is equal to the sample size minus 1 ($n = N - 1$).

The Anderson statistic is distributed as a χ^2 with $(q+2)(q-1)/2$ degrees of freedom and is equal to:

$$-n \log(V_k) \sim \chi^2_{(q+2)(q-1)/2} \tag{3}$$

An improvement of this statistic from Bartlett (Bentler, and Yuan, 1996, p. 300; Horn and Engstrom, 1979, equation 8) is distributed as a χ^2 with $(q)(q-1)/2$ degrees of freedom and is equal to:

(4) $$-\left[n-k-\frac{2q^2q+2}{6q}\right]\log(V_k)\sim\chi^2_{(q+2)(q-1)/2}$$

Finally, Anderson (1956) and James (1969) proposed another statistic.

(5) $$-\left[n-k-\frac{2q^2q+2}{6q}+\sum_{i=1}^{k}\frac{\bar{\lambda}_q^2}{\left(\lambda_i-\bar{\lambda}_q\right)^2}\right]\log(V_k)\sim\chi^2_{(q+2)(q-1)/2}$$

Bartlett (1950, 1951) proposed a correction to the degrees of freedom of these χ^2 after the first significant test: $(q+2)(q-1)/2$.

Value

`nFactors`	numeric: vector of the number of factors retained by the Bartlett, Anderson and Lawley procedures.
`details`	numeric: matrix of the details for each index.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, `http://www.er.uqam.ca/nobel/r17165/`

References

Anderson, T. W. (1963). Asymptotic theory for principal component analysis. *Annals of Mathematical Statistics, 34*, 122-148.

Bartlett, M. S. (1950). Tests of significance in factor analysis. *British Journal of Psychology, 3*, 77-85.

Bartlett, M. S. (1951). A further note on tests of significance. *British Journal of Psychology, 4*, 1-2.

Bentler, P. M. and Yuan, K.-H. (1996). Test of linear trend in eigenvalues of a covariance matrix with application to data analysis. *British Journal of Mathematical and Statistical Psychology, 49*, 299-312.

Bentler, P. M. and Yuan, K.-H. (1998). Test of linear trend in the smallest eigenvalues of the correlation matrix. *Psychometrika, 63*(2), 131-144.

Horn, J. L. and Engstrom, R. (1979). Cattell's scree test in relation to Bartlett's chi-square test and other observations on the number of factors problem. *Multivariate Behavioral Reasearch, 14*(3), 283-300.

James, A. T. (1969). Test of equality of the latent roots of the covariance matrix. *In* P. K. Krishna (Eds): *Multivariate analysis, volume 2*. New-York, NJ: Academic Press.

Lawley, D. N. (1956). Tests of significance for the latent roots of covariance and correlation matrix. *Biometrika, 43*(1/2), 128-136.

See Also

`plotuScree`, `nScree`, `plotnScree`, `plotParallel`

Examples

```
## SIMPLE EXAMPLE OF A BARTLETT PROCEDURE

 data(dFactors)
 eig      <- dFactors$Raiche$eigenvalues

 results  <- nBartlett(x=eig, N= 100, alpha=0.05, details=TRUE)
 results

 plotuScree(eig, main=paste(results$nFactors[1], ", ",
                            results$nFactors[2], " or ",
                            results$nFactors[3],
                            " factors retained by the LRT procedures",
                            sep=""))
```

---

nBentler *Bentler and Yuan's Procedure to Determine the Number of Components/Factors*

Description

This function computes the Bentler and Yuan's indices for determining the number of components/factors to retain.

Usage

```
nBentler(x, N, log=TRUE, alpha=0.05, cor=TRUE, details=TRUE,
        minPar=c(min(lambda) - abs(min(lambda)) +.001, 0.001),
        maxPar=c(max(lambda),
                 lm(lambda ~ I(length(lambda):1))$coef[2]), ...)
```

Arguments

`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data
`N`	numeric: number of subjects.
`log`	logical: if `TRUE` does the maximization on the log values.
`alpha`	numeric: statistical significance level.
`cor`	logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix
`details`	logical: if `TRUE` also returns detains about the computation for each eigenvalue.
`minPar`	numeric: minimums for the coefficient of the linear trend to maximize.
`maxPar`	numeric: maximums for the coefficient of the linear trend to maximize.
`...`	variable: additionnal parameters to give to the `cor` or `cov` functions

Details

The implemented Bentler and Yuan's procedure must be used with care because the minimized function is not always stable, as Bentler and Yan (1996, 1998) already noted. In many cases, constraints must applied to obtain a solution, as the actual implementation did, but the user can modify these constraints.

The hypothesis tested (Bentler and Yuan, 1996, equation 10) is:

(1) $$H_k : \lambda_{k+i} = \alpha + \beta x_i, (i = 1, \ldots, q)$$

The solution of the following simultaneous equations is needed to find $(\alpha, \beta) \in$

(2) $$f(x) = \sum_{i=1}^{q} \frac{[\lambda_{k+j} - N\alpha + \beta x_j]x_j}{(\alpha + \beta x_j)^2} = 0$$

and $$g(x) = \sum_{i=1}^{q} \frac{\lambda_{k+j} - N\alpha + \beta x_j x_j}{(\alpha + \beta x_j)^2} = 0$$

The solution to this system of equations was implemented by minimizing the following equation:

(3) $$(\alpha, \beta) \in \inf\left[h(x)\right] = \inf \log\left[f(x)^2 + g(x)^2\right]$$

The likelihood ratio test LRT proposed by Bentler and Yuan (1996, equation 7) follows a χ^2 probability distribution with $q - 2$ degrees of freedom and is equal to:

(4) $$LRT = N(k-p)\left\{\ln\left(\frac{n}{N}\right) + 1\right\} - N\sum_{j=k+1}^{p} \ln\left\{\frac{\lambda_j}{\alpha + \beta x_j}\right\} + n\sum_{j=k+1}^{p}\left\{\frac{\lambda_j}{\alpha + \beta x_j}\right\}$$

With p beeing the number of eigenvalues, k the number of eigenvalues to test, q the $p-k$ remaining eigenvalues, N the sample size, and $n = N - 1$. Note that there is an error in the Bentler and Yuan equation, the variables N and n beeing inverted in the preceeding equation 4.

A better strategy proposed by Bentler an Yuan (1998) is to used a minimized χ^2 solution. This strategy will be implemented in a future version of the **nFactors** package.

Value

`nFactors`	numeric: vector of the number of factors retained by the Bentler and Yuan's procedure.
`details`	numeric: matrix of the details of the computation.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

David Magis
Departement de mathematiques
Universite de Liege
`<David.Magis@ulg.ac.be>`

References

Bentler, P. M. and Yuan, K.-H. (1996). Test of linear trend in eigenvalues of a covariance matrix with application to data analysis. *British Journal of Mathematical and Statistical Psychology, 49*, 299-312.

Bentler, P. M. and Yuan, K.-H. (1998). Test of linear trend in the smallest eigenvalues of the correlation matrix. *Psychometrika, 63*(2), 131-144.

See Also

nBartlett, bentlerParameters

Examples

```
## ................................................
## SIMPLE EXAMPLE OF THE BENTLER AND YUAN PROCEDURE

# Bentler (1996, p. 309) Table 2 - Example 2 ............
n=649
bentler2<-c(5.785, 3.088, 1.505, 0.582, 0.424, 0.386, 0.360, 0.337, 0.303,
            0.281, 0.246, 0.238, 0.200, 0.160, 0.130)

results  <- nBentler(x=bentler2, N=n)
results

plotuScree(x=bentler2, model="components",
    main=paste(results$nFactors,
    " factors retained by the Bentler and Yuan's procedure (1996, p. 309)",
    sep=""))
# ........................................................

# Bentler (1998, p. 140) Table 3 - Example 1 ............
n        <- 145
example1 <- c(8.135, 2.096, 1.693, 1.502, 1.025, 0.943, 0.901, 0.816, 0.790,
              0.707, 0.639, 0.543,
              0.533, 0.509, 0.478, 0.390, 0.382, 0.340, 0.334, 0.316, 0.297,
              0.268, 0.190, 0.173)

results  <- nBentler(x=example1, N=n)
results

plotuScree(x=example1, model="components",
   main=paste(results$nFactors,
   " factors retained by the Bentler and Yuan's procedure (1998, p. 140)",
   sep=""))
# ........................................................
```

---

nCng *Cattell-Nelson-Gorsuch CNG Indices*

Description

This function computes the *CNG* indices for the eigenvalues of a correlation/covariance matrix (Gorsuch and Nelson, 1981; Nasser, 2002, p. 400; Zoski and Jurs, 1993, p. 6).

Usage

```
nCng(x, cor=TRUE, model="components", details=TRUE, ...)
```

Arguments

`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data
`cor`	logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix
`model`	character: `"components"` or `"factors"`
`details`	logical: if `TRUE` also returns detains about the computation for each eigenvalue.
`...`	variable: additionnal parameters to give to the `eigenComputes` function

Details

Note that the `nCng` function is only valid when more than six eigenvalues are used and that these are obtained in the context of a principal component analysis. For a factor analysis, some eigenvalues could be negative and the function will stop and give an error message.

The slope of all possible sets of three adjacent eigenvalues are compared, so *CNG* indices can be applied only when more than six eigenvalues are used. The eigenvalue at which the greatest difference between two successive slopes occurs is the indicator of the number of components/factors to retain.

Value

`nFactors`	numeric: number of factors retained by the CNG procedure.
`details`	numeric: matrix of the details for each index.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

References

Gorsuch, R. L. and Nelson, J. (1981). *CNG scree test: an objective procedure for determining the number of factors*. Presented at the annual meeting of the Society for multivariate experimental psychology.

Nasser, F. (2002). The performance of regression-based variations of the visual scree for determining the number of common factors. *Educational and Psychological Measurement, 62(3)*, 397-419.

Zoski, K. and Jurs, S. (1993). Using multiple regression to determine the number of factors to retain in factor analysis. *Multiple Linear Regression Viewpoints, 20*(1), 5-9.

See Also

`plotuScree`, `nScree`, `plotnScree`, `plotParallel`

Examples

```
## SIMPLE EXAMPLE OF A CNG ANALYSIS

 data(dFactors)
 eig      <- dFactors$Raiche$eigenvalues

 results  <- nCng(eig, details=TRUE)
 results

 plotuScree(eig, main=paste(results$nFactors,
                            " factors retained by the CNG procedure",
                            sep=""))
```

`nFactors-parameters` *Argument and Value Parameters Common to the Different Functions Available in Package nFactors*

Description

This help file describes the argument and value parameters used in the different functions available in package **nFactors**.

Arguments:

1. *adequacy*: logical: if `TRUE` prints the recovered population matrix from the factor structure (`structureSim`)
2. *all*: logical: if `TRUE` computes he Bentler and Yuan index (very long computing time to consider) (`structureSim, studySim`)
3. *alpha*: numeric: statistical significance level (`nBartlett, nBentler`)
4. *aparallel*: numeric: results of a parallel analysis (`nScree`)
5. *cent*: depreciated numeric (use quantile instead): quantile of the distribution (`moreStats, parallel`)
6. *communalities*: character: initial values for communalities (`"component"`, `"ginv"`, `"maxr"`, or `"multiple"`) (`iterativePrincipalAxis, principalAxis`)
7. *cor*: logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix (`eigenComputes, nBartlett, nBentler, nCng, nMreg, nScree, nSeScree`)
8. *correction*: logical: if `TRUE` uses a correction for the degree of freedom after the first eigenvalue (`nBartlett`)
9. *criteria*: numeric: by default fixed at $\hat{\lambda}$. When the λs are computed from a principal components analysis on a correlation matrix, it corresponds to the usual Kaiser $\lambda >= 1$ rule. On a covariance matrix or from a factor analysis, it is simply the mean. To apply the $\lambda >= 0$ sometimes used with factor analysis, fixed the criteria to 0 (`nScree`)
10. *details*: logical: if `TRUE` also returns details about the computation for each eigenvalues (`nBartlett, nBentler, nCng, nMreg, structureSim`)
11. *diagCommunalities*: logical: if `TRUE`, the correlation between the initial solution and the estimated one will use a correlation of one in the diagonal. If `FALSE` (default) the diagonal is not used in the computation of this correlation or covariance matrix (`rRecovery`)

12. *dir*: character: directory where to save output (`studySim`)
13. *eig*: depreciated parameter (use x instead): eigenvalues to analyse (`nScree, plotParallel`)
14. *Eigenvalue*: depreciated parameter (use x instead): eigenvalues to analyse (`plotuScree`)
15. *fload*: matrix: loadings of the factor structure (`structureSim`)
16. *graphic*: logical: specific plot (`bentlerParameters, structureSim`)
17. *index*: numeric: vector of the index of the selected indices (`plot.structureSim, print.structureSim, summary.structureSim`
18. *iterations*: numeric: maximum number of iterations to obtain a solution (`iterativePrincipalAxis`)
19. *legend*: logical indicator of the presence of a legend (`plotnScree, plotParallel`)
20. *loadings*: numeric: loadings from a factor analysis solution (`rRecovery, generateStructure, studySim`)
21. *log*: logical: if `TRUE` does the minimization on the log values (`bentlerParameters, nBentler`)
22. *main*: character: main title (`plotnScree, plotParallel, plotuScree, boxplot.structureSim, plot.structureSim`)
23. *maxPar*: numeric: maximums for the coefficient of the linear trend (`bentlerParameters, nBentler`)
24. *minPar*: numeric: minimums for the coefficient of the linear trend (`bentlerParameters, nBentler`)
25. *method*: character: actually only `"giv"` is supplied to compute the approximation of the communalities by maximum correlation (`corFA, nCng, nMreg, nScree, nSeScree`)
26. *mjc*: numeric: number of major factors (factors with practical significance) (`generateStructure`)
27. *pmjc*: numeric: number of variables that load significantly on each major factor (`generateStructure`)
28. *model*: character: `"components"` or `"factors"` (`nScree, parallel, plotParallel, plotuScree, structureSim, eigenBootParallel, eigenBootParallel, studySim`)
29. *N*: numeric: number of subjects (`nBartlett, bentlerParameters, nBentler, studySim`)
30. *nboot*: numeric: number of bootstrap samples (`eigenBootParallel`)
31. *nFactors*: numeric: number of components/factors to retained (`componentAxis, iterativePrincipalAxis, principalAxis, bentlerParameters, boxplot.structureSim, studySim`)
32. *nScree*: results of a previous nScree analysis (`plotnScree`)
33. *option*: character: `"permutation"` or `"bootstrap"` (`eigenBootParallel`)
34. *object*: nScree: an object of the class nScree `is.nScree, summary.nScree`
35. *object*: structureSim: an object of the class structureSim (`is.structureSim, summary.structureSim`)
36. *parallel*: numeric: vector of the result of a previous parallel analysis (`plotParallel`)
37. *pmjc*: numeric: number of major loadings on each factor factors (`generateStructure, studySim`)
38. *quantile*: numeric: quantile that will be reported (`parallel, moreStats, eigenBootParallel, structureSim, studySim`)
39. *R*: numeric: correlation or covariance matrix (`componentAxis, iterativePrincipalAxis, principalAxis, principalComponents, rRecovery, corFA`)
40. *r2limen*: numeric: R2 limen value for the R2 Nelson index (`structureSim, nSeScree, studySim`)

41. *rep*: numeric: number of replications of the correlation or the covariance matrix (default is 100) (`parallel`)
42. *reppar*: numeric: number of replications for the parallel analysis (`structureSim, studySim`)
43. *repsim*: numeric: number of replications of the matrix correlation simulation (`structureSim, studySim`)
44. *resParx*: numeric: restriction on the α coefficient (x) to graph the function to minimize (`bentlerParameters`)
45. *resolution*: numeric: resolution of the 3D graph (number of points from α and from β).
46. *resPary*: numeric: restriction on the β coefficient (y) to graph the function to minimize (`bentlerParameters`)
47. *sd*: numeric: vector of standard deviations of the simulated variables (for a parallel analysis on a covariance matrix) `parallel`)
48. *show*: logical: if `TRUE` prints the quantile chosen (`moreStats`)
49. *stats*: numeric: vector of the statistics to return: mean(1), median(2), sd(3), quantile(4), min(5), max(6) (`studySim`)
50. *subject*: numeric: number of subjects (default is 100) (`parallel`)
51. *tolerance*: numeric: minimal difference in the estimated communalities after a given iteration (`iterativePrincipalAxis`)
52. *trace*: logical: if `TRUE` gives details of the status of the simulations (`studySim`)
53. *typePlot*: character: plots the minimized function according to a 3D plot: `"wireframe"`, `"contourplot"` or `"levelplot"` (`bentlerParameters`)
54. *unique*: numeric: loadings on the non significant variables on each major factor (`generateStructure, studySim`)
55. *upper*: logical: if `TRUE` upper diagonal is replaced with lower diagonal. If `FALSE`, lower diagonal is replaced with upper diagonal (`diagReplace`)
56. *use*: character: how to deal with missing values, same as the parameter from the `corr` function (`eigenBootParallel`)
57. *var*: numeric: number of variables (default is 10) (`parallel, generateStructure, studySim`)
58. *vLine*: character: color of the vertical indicator line in the eigen boxplot (`boxplot.structureSim`)
59. *x*: numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data (`eigenFrom, nBartlett, nCng, nMreg`)
60. *xlab*: character: label of the x axis (`plotnScree, plotParallel, plotuScree, boxplot.structureSim`)
61. *x*: data.frame: data from which a correlation or covariance matrix will be obtained (`eigenBootParallel`)
62. *x*: depreciated: (`plotParallel`)
63. *x*: nScree: an object of the class nScree (`plot.nScree, print.nScree`)
64. *x*: numeric: matrix (`makeCor`)
65. *x*: numeric: matrix or data.frame (`moreStats`)
66. *x*: structureSim: an object of the class structureSim (`boxplot.structureSim, plot.structureSim, print.structureSim`)
67. *ylab*: character: label of the y axis (`plotnScree, plotParallel, plotuScree, boxplot.structureSim`)

Values:

1. *cor*: numeric: Pearson correlation between initial and recovered estimated correlation or covariance matrix. Compution depend on the logical value of the `communalities` argument (`rRecovery`)
2. *details*: numeric: matrix of the details for each index (`nBartlett, bentlerParameters, nCng, nMreg`)
3. *difference*: numeric: difference between initial and recovered estimated correlation or covariance matrix (`rRecovery`)
4. *iterations*: numeric: maximum number of iterations to obtain a solution (`iterativePrincipalAxis`)
5. *loadings*: numeric: loadings of each variable on each component or factor retained (`componentAxis, iterativePrincipalAxis, principalAxis, principalComponents`)
6. *nFactors*: numeric: vector of the number of components or factors retained by the Bartlett, Anderson and Lawley procedures (`nBartlett, bentlerParameters, nCng, nMreg`)
7. *R*: numeric: correlation or covariance matrix (`diagReplace, rRecovery`)
8. *recoveredR*: numeric: recovered estimated correlation or covariance matrix (`rRecovery`)
9. *tolerance*: numeric: minimal difference in the estimated communalities after a given iteration (`iterativePrincipalAxis`)
10. *values*: numeric: data.frame of information (`nScree, parallel, plotnScree, plotParallel, plotuScree, structureSim`)
11. *values*: numeric: data.frame of statistics (`moreStats`)
12. *values*: numeric: full matrix of correlation or covariance (`makeCor`)
13. *values*: numeric: variance of each component or factor (`iterativePrincipalAxis, principalComponents`)
14. *values*: data.frame: mean, median, quantile, standard deviation, minimum and maximum of bootstrapped eigenvalues (`eigenBootParallel`)
15. *values*: numeric: matrix of correlation or covariance with communalities in the diagonal (`corFA`)
16. *values*: numeric: variance of each component or factor retained (`componentAxis, principalAxis`)
17. *values*: numeric: matrix factor structure (`generateStructure`)
18. *varExplained*: numeric: variance explained by each component or factor retained (`componentAxis, iterativePrincipalAxis, principalAxis, principalComponents`)
19. *varExplained*: numeric: cumulative variance explained by each component or factor retained (`componentAxis, iterativePrincipalAxis, principalAxis, principalComponents`)

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

David Magis
Departement de mathematiques
Universite de Liege
`<David.Magis@ulg.ac.be>`

References

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

See Also

Other packages are also very useful for principal component and factor analysis. The *R* psychometric view is instructive at this point. See http://cran.stat.sfu.ca/web/views/Psychometrics.html for further details.

`nFactorsObjectMethods` *Utility Functions for nFactors Class Objects*

Description

Utility functions for `nFactors` class objects.

Usage

```
 ## S3 method for class 'nFactors':
is      (object)
 ## S3 method for class 'nFactors':
print   (x, ...)
 ## S3 method for class 'nFactors':
summary(object, ...)
```

Arguments

`x`	nFactors: an object of the class nFactors
`object`	nFactors: an object of the class nFactors
`...`	variable: additionnal parameters to give to the `print` function with `print.nFactors` or to the `summary` function with `summary.nFactors`

Value

Generic functions for the nFactors class:

`is.nFactors`	logical: is the object of the class nFactors?
`print.nFactors`	numeric: vector of the number of components/factors to retain: same as the `nFactors` vector from the `nFactors` object
`summary.nFactors`	data.frame: details of the results from a nFactors object: same as the `details` data.frame from the `nFactors` object, but with easier control of the number of decimals with the `digits` parameter

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

See Also

nBentler, nBartlett, nCng, nMreg, nSeScree

Examples

```
## SIMPLE EXAMPLE
 data(dFactors)
 eig      <- dFactors$Raiche$eigenvalues
 N        <- dFactors$Raiche$nsubjects

 res <- nBartlett(eig,N); res; is.nFactors(res); summary(res, digits=2)
 res <- nBentler(eig,N);  res; is.nFactors(res); summary(res, digits=2)
 res <- nCng(eig);        res; is.nFactors(res); summary(res, digits=2)
 res <- nMreg(eig);       res; is.nFactors(res); summary(res, digits=2)
 res <- nSeScree(eig);    res; is.nFactors(res); summary(res, digits=2)

## SIMILAR RESULTS, BUT NOT A nFactors OBJECT
 res <- nScree(eig);      res; is.nFactors(res); summary(res, digits=2)
```

---

nMreg *Multiple Regression Procedure to Determine the Number of Components/Factors*

Description

This function computes the β indices, like their associated Student t and probability (Zoski and Jurs, 1993, 1996, p. 445). These three values can be used as three different indices for determining the number of components/factors to retain.

Usage

```
nMreg(x, cor=TRUE, model="components", details=TRUE, ...)
```

Arguments

`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data (eigenFrom)
`cor`	logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix
`model`	character: `"components"` or `"factors"`
`details`	logical: if `TRUE` also returns details about the computation for each eigenvalue.
`...`	variable: additionnal parameters to give to the `eigenComputes` and `cor` or `cov` functions

Details

When the associated Student *t* test is applied, the following hypothesis is considered:

$$(1) \qquad H_k : \beta(\lambda_1 \ldots \lambda_k) - \beta(\lambda_{k+1} \ldots \lambda_p), (k = 3, \ldots, p-3) = 0$$

Value

`nFactors`	numeric: number of components/factors retained by the *MREG* procedures.
`details`	numeric: matrix of the details for each indices.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

References

Zoski, K. and Jurs, S. (1993). Using multiple regression to determine the number of factors to retain in factor analysis. *Multiple Linear Regression Viewpoints, 20*(1), 5-9.

Zoski, K. and 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.

See Also

`plotuScree`, `nScree`, `plotnScree`, `plotParallel`

Examples

```
## SIMPLE EXAMPLE OF A MREG ANALYSIS

 data(dFactors)
 eig      <- dFactors$Raiche$eigenvalues

 results  <- nMreg(eig)
 results

 plotuScree(eig, main=paste(results$nFactors[1], ", ",
                            results$nFactors[2], " or ",
```

```
                        results$nFactors[3],
                        " factors retained by the MREG procedures",
                        sep=""))
```

---

nScree *Non Graphical Cattel's Scree Test*

Description

The `nScree` function returns an analysis of the number of component or factors to retain in an exploratory principal component or factor analysis. The function also returns information about the number of components/factors to retain with the Kaiser rule and the parallel analysis.

Usage

```
nScree(eig=NULL, x=eig, aparallel=NULL, cor=TRUE, model="components",
       criteria=NULL, ...)
```

Arguments

`eig`	depreciated parameter (use x instead): eigenvalues to analyse
`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data
`aparallel`	numeric: results of a parallel analysis. Defaults eigenvalues fixed at $\lambda >= \bar{\lambda}$ (Kaiser and related rule) or $\lambda >= 0$ (CFA analysis)
`cor`	logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix
`model`	character: `"components"` or `"factors"`
`criteria`	numeric: by default fixed at $\bar{\lambda}$. When the λs are computed from a principal component analysis on a correlation matrix, it corresponds to the usual Kaiser $\lambda >= 1$ rule. On a covariance matrix or from a factor analysis, it is simply the mean. To apply $\lambda >= 0$, sometimes used with factor analysis, fix the criteria to 0.
`...`	variabe: additionnal parameters to give to the `cor` or `cov` functions

Details

The `nScree` function returns an analysis of the number of components/factors to retain in an exploratory principal component or factor analysis. Different solutions are given. The classical ones are the Kaiser rule, the parallel analysis, and the usual scree test (`plotuScree`). Non graphical solutions to the Cattell subjective scree test are also proposed: an acceleration factor (*af*) and the optimal coordinates index *oc*. The acceleration factor indicates where the elbow of the scree plot appears. It corresponds to the acceleration of the curve, i.e. the second derivative. The optimal coordinates are the extrapolated coordinates of the previous eigenvalue that allow the observed eigenvalue to go beyond this extrapolation. The extrapolation is made by a linear regression using the last eigenvalue coordinates and the $k+1$ eigenvalue coordinates. There are $k-2$ regression lines like this. The Kaiser rule or a parallel analysis criterion (`parallel`) must also be simultaneously satisfied to retain the components/factors, whether for the acceleration factor, or for the optimal coordinates.

If λ_i is the i^{th} eigenvalue, and LS_i is a location statistics like the mean or a centile (generally the followings: $1^{st},\ 5^{th},\ 95^{th},\ or\ 99^{th}$).

The Kaiser rule is computed as:

$$n_{Kaiser} = \sum_i (\lambda_i \geq \bar{\lambda}).$$

Note that $\bar{\lambda}$ is equal to 1 when a correlation matrix is used.

The parallel analysis is computed as:

$$n_{parallel} = \sum_i (\lambda_i \geq LS_i).$$

The acceleration factor (AF) corresponds to a numerical solution to the elbow of the scree plot:

$$n_{AF} \equiv\ If\ \left[(\lambda_i \geq LS_i)\ and\ max(AF_i)\right].$$

The optimal coordinates (OC) corresponds to an extrapolation of the preceeding eigenvalue by a regression line between the eigenvalue coordinates and the last eigenvalue coordinates:

$$n_{OC} = \sum_i \left[(\lambda_i \geq LS_i) \cap (\lambda_i \geq (\lambda_{i\ predicted})\right].$$

Value

`Components` Data frame for the number of components/factors according to different rules

`Components$noc`
Number of components/factors to retain according to optimal coordinates *oc*

`Components$naf`
Number of components/factors to retain according to the acceleration factor *af*

`Components$npar.analysis`
Number of components/factors to retain according to parallel analysis

`Components$nkaiser`
Number of components/factors to retain according to the Kaiser rule

`Analysis` Data frame of vectors linked to the different rules

`Analysis$Eigenvalues`
Eigenvalues

`Analysis$Prop`
Proportion of variance accounted by eigenvalues

`Analysis$Cumu`
Cumulative proportion of variance accounted by eigenvalues

`Analysis$Par.Analysis`
Centiles of the random eigenvalues generated by the parallel analysis.

`Analysis$Pred.eig`
Predicted eigenvalues by each optimal coordinate regression line

`Analysis$OC` Critical optimal coordinates *oc*

`Analysis$Acc.factor`
Acceleration factor *af*

`Analysis$AF` Critical acceleration factor *af*

Otherwise, returns a summary of the analysis.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research, 1*, 245-276.

Dinno, A. (2009). *Gently clarifying the application of Horn's parallel analysis to principal component analysis versus factor analysis*. Portland, Oregon: Portland Sate University [http://doyenne.com/Software/files/PA_for_PCA_vs_FA.pdf]

Guttman, L. (1954). Some necessary conditions for common factor analysis. *Psychometrika, 19, 149-162*.

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

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

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

See Also

plotuScree, plotnScree, parallel, plotParallel,

Examples

```
## INITIALISATION
 data(dFactors)                      # Load the nFactors dataset
 attach(dFactors)
 vect         <- Raiche              # Uses the example from Raiche
 eigenvalues  <- vect$eigenvalues    # Extracts the observed eigenvalues
 nsubjects    <- vect$nsubjects      # Extracts the number of subjects
 variables    <- length(eigenvalues) # Computes the number of variables
 rep          <- 100                 # Number of replications for PA analysis
 cent         <- 0.95                # Centile value of PA analysis

## PARALLEL ANALYSIS (qevpea for the centile criterion, mevpea for the
## mean criterion)
 aparallel    <- parallel(var     = variables,
                          subject = nsubjects,
                          rep     = rep,
                          cent    = cent
                          )$eigen$qevpea  # The 95 centile

## NUMBER OF FACTORS RETAINED ACCORDING TO DIFFERENT RULES
 results      <- nScree(x=eigenvalues, aparallel=aparallel)
 results
 summary(results)

## PLOT ACCORDING TO THE nScree CLASS
```

```
plotnScree(results)
```

---

nScreeObjectMethods *Utility Functions for nScree Class Objects*

Description

Utility functions for `nScree` class objects. Some of these functions are already implemented in the `nFactors` package, but are easier to use with generic functions like these.

Usage

```
## S3 method for class 'nScree':
is      (object)
## S3 method for class 'nScree':
plot    (x, ...)
## S3 method for class 'nScree':
print   (x, ...)
## S3 method for class 'nScree':
summary(object, ...)
```

Arguments

`x`	nScree: an object of the class `nScree`
`object`	nScree: an object of the class `nScree`
`...`	variable: additionnal parameters to give to the `print` function with `print.nScree`, the `plotnScree` with `plot.nScree` or to the `summary` function with `summary.nScree`

Value

Generic functions for the nScree class:

`is.nScree`	logical: is the object of the class `nScree`?
`plot.nScree`	graphic: plots a figure according to the `plotnScree` function
`print.nScree`	numeric: vector of the number of components/factors to retain: same as the `Components` vector from the `nScree` object
`summary.nScree`	data.frame: details of the results from a nScree analysis: same as the `Analysis` data.frame from the `nScree` object, but with easier control of the number of decimals with the `digits` parameter

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

See Also

plotuScree, plotnScree, parallel, plotParallel,

Examples

```
## INITIALISATION
 data(dFactors)                      # Load the nFactors dataset
 attach(dFactors)
 vect          <- Raiche             # Use the example from Raiche
 eigenvalues   <- vect$eigenvalues   # Extract the observed eigenvalues
 nsubjects     <- vect$nsubjects     # Extract the number of subjects
 variables     <- length(eigenvalues) # Compute the number of variables
 rep           <- 100                # Number of replications for the parallel analysis
 cent          <- 0.95               # Centile value of the parallel analysis

## PARALLEL ANALYSIS (qevpea for the centile criterion, mevpea for the mean criterion)
 aparallel     <- parallel(var     = variables,
                          subject = nsubjects,
                          rep     = rep,
                          cent    = cent
                          )$eigen$qevpea  # The 95 centile

## NOMBER OF FACTORS RETAINED ACCORDING TO DIFFERENT RULES
 results       <- nScree(x=eigenvalues, aparallel=aparallel)

 is.nScree(results)
 results
 summary(results)

## PLOT ACCORDING TO THE nScree CLASS
 plot(results)
```

---

nSeScree — *Standard Error Scree and Coefficient of Determination Procedures to Determine the Number of Components/Factors*

Description

This function computes the *seScree* ($S_{Y \bullet X}$) indices (Zoski and Jurs, 1996) and the coefficient of determination indices of Nelson (2005) R^2 for determining the number of components/factors to retain.

Usage

```
nSeScree(x, cor=TRUE, model="components", details=TRUE, r2limen=0.75, ...)
```

Arguments

`x`	numeric: eigenvalues.
`cor`	logical: if `TRUE` computes eigenvalues from a correlation matrix, else from a covariance matrix
`model`	character: `"components"` or `"factors"`
`details`	logical: if `TRUE` also returns details about the computation for each eigenvalue.
`r2limen`	numeric: criterion value retained for the coefficient of determination indices.
`...`	variable: additionnal parameters to give to the `eigenComputes` and `cor` or `cov` functions

Details

The Zoski and Jurs $S_{Y \bullet X}$ index is the standard error of the estimate (predicted) eigenvalues by the regression from the $(k+1, \ldots, p)$ subsequent ranks of the eigenvalues. The standard error is computed as:

(1) $$S_{Y \bullet X} = \sqrt{\frac{(\lambda_k - \hat{\lambda}_k)^2}{p-2}}$$

A value of $1/p$ is choosen as the criteria to determine the number of components or factors to retain, *p* corresponding to the number of variables.

The Nelson R^2 index is simply the multiple regresion coefficient of determination for the $k + 1, \ldots, p$ eigenvalues. Note that Nelson didn't give formal prescriptions for the criteria for this index. He only suggested that a value of 0.75 or more must be considered. More is to be done to explore adequate values.

Value

`nFactors`	numeric: number of components/factors retained by the seScree procedure.
`details`	numeric: matrix of the details for each index.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

References

Nasser, F. (2002). The performance of regression-based variations of the visual scree for determining the number of common factors. *Educational and Psychological Measurement, 62(3)*, 397-419.

Nelson, L. R. (2005). Some observations on the scree test, and on coefficient alpha. *Thai Journal of Educational Research and Measurement, 3(1)*, 1-17.

Zoski, K. and Jurs, S. (1993). Using multiple regression to determine the number of factors to retain in factor analysis. *Multiple Linear Regression Viewpoints, 20*(1), 5-9.

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

See Also

plotuScree, nScree, plotnScree, plotParallel

Examples

```
## SIMPLE EXAMPLE OF SESCREE AND R2 ANALYSIS

 data(dFactors)
 eig      <- dFactors$Raiche$eigenvalues

 results  <- nSeScree(eig)
 results

 plotuScree(eig, main=paste(results$nFactors[1], " or ", results$nFactors[2],
                            " factors retained by the sescree and R2 procedures",
                            sep=""))
```

`parallel` *Parallel Analysis of a Correlation or Covariance Matrix*

Description

This function gives the distribution of the eigenvalues of correlation or a covariance matrices of random uncorrelated standardized normal variables. The mean and a selected quantile of this distribution are returned.

Usage

```
parallel(subject  = 100,
         var      = 10,
         rep      = 100,
         cent     = 0.05,
         quantile = cent,
         model    = "components",
         sd       = diag(1,var),
         ...)
```

Arguments

`subject`	numeric: nmber of subjects (default is 100)
`var`	numeric: number of variables (default is 10)
`rep`	numeric: number of replications of the correlation matrix (default is 100)
`cent`	depreciated numeric (use quantile instead): quantile of the distribution on which the decision is made (default is 0.05)
`quantile`	numeric: quantile of the distribution on which the decision is made (default is 0.05)
`model`	character: `"components"` or `"factors"`
`sd`	numeric: vector of standard deviations of the simulated variables (for a parallel analysis on a covariance matrix)
`...`	variable: other parameters for the `"mvrnorm"`, `corr` or `cov` functions

Details

Note that if the decision is based on a quantile value rather than on the mean, care must be taken with the number of replications (`rep`). In fact, the smaller the quantile (`cent`), the bigger the number of necessary replications.

Value

`eigen`	Data frame consisting of the mean and the quantile of the eigenvalues distribution
`eigen$mevpea`	Mean of the eigenvalues distribution
`eigen$sevpea`	Standard deviation of the eigenvalues distribution
`eigen$qevpea`	quantile of the eigenvalues distribution
`eigen$sqevpea`	Standard error of the quantile of the eigenvalues distribution
`subject`	Number of subjects
`variables`	Number of variables
`centile`	Selected quantile

Otherwise, returns a summary of the parallel analysis.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

References

Drasgow, F. and Lissak, R. (1983) Modified parallel analysis: a procedure for examining the latent dimensionality of dichotomously scored item responses. *Journal of Applied Psychology, 68*(3), 363-373.

Hoyle, R. H. and Duvall, J. L. (2004). Determining the number of factors in exploratory and confirmatory factor analysis. In D. Kaplan (Ed.): *The Sage handbook of quantitative methodology for the social sciences*. Thousand Oaks, CA: Sage.

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

See Also

`plotuScree`, `nScree`, `plotnScree`, `plotParallel`

Examples

```
## SIMPLE EXAMPLE OF A PARALLEL ANALYSIS
## OF A CORRELATION MATRIX WITH ITS PLOT
 data(dFactors)
 eig      <- dFactors$Raiche$eigenvalues
 subject  <- dFactors$Raiche$nsubjects
 var      <- length(eig)
 rep      <- 100
 quantile <- 0.95
```

```
 results  <- parallel(subject, var, rep, quantile)

 results

## IF THE DECISION IS BASED ON THE CENTILE USE qevpea INSTEAD
## OF mevpea ON THE FIRST LINE OF THE FOLLOWING CALL
 plotuScree(x    = eig,
            main = "Parallel Analysis"
            )

 lines(1:var,
       results$eigen$qevpea,
       type="b",
       col="green"
       )


## ANOTHER SOLUTION IS SIMPLY TO
 plotParallel(results)
```

`plotnScree` *Scree Plot According to a nScree Object Class*

Description

Plot a scree plot adding information about a non graphical `nScree` analysis.

Usage

```
plotnScree(nScree,
           legend = TRUE,
           ylab   = "Eigenvalues",
           xlab   = "Components",
           main   = "Non Graphical Solutions to Scree Test"
           )
```

Arguments

`nScree`	Results of a previous `nScree` analysis
`legend`	Logical indicator of the presence or not of a legend
`xlab`	Label of the x axis (default to "`Component`")
`ylab`	Label of the y axis (default to "`Eigenvalue`")
`main`	Main title (default to "`Non Graphical Solutions to the Scree Test`")

Value

Nothing returned.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

See Also

plotuScree, nScree, plotParallel, parallel

Examples

```
## INITIALISATION
 data(dFactors)                        # Load the nFactors dataset
 attach(dFactors)
 vect         <- Raiche                # Use the second example from Buja and Eyuboglu
                                       # (1992, p. 519, nsubjects not specified by them)
 eigenvalues  <- vect$eigenvalues      # Extract the observed eigenvalues
 nsubjects    <- vect$nsubjects        # Extract the number of subjects
 variables    <- length(eigenvalues) # Compute the number of variables
 rep          <- 100                   # Number of replications for the parallel analysis
 cent         <- 0.95                  # Centile value of the parallel analysis

## PARALLEL ANALYSIS (qevpea for the centile criterion, mevpea for the mean criterion)
 aparallel    <- parallel(var     = variables,
                          subject = nsubjects,
                          rep     = rep,
                          cent    = cent)$eigen$qevpea  # The 95 centile

## NOMBER OF FACTORS RETAINED ACCORDING TO DIFFERENT RULES
 results <- nScree(eig       = eigenvalues,
                   aparallel = aparallel
                   )

 results

## PLOT ACCORDING TO THE nScree CLASS
 plotnScree(results)
```

---

plotParallel   *Plot a Parallel Analysis Class Object*

Description

Plot a scree plot adding information about a parallel analysis.

Usage

```
plotParallel(parallel,
             eig    = NA,
             x      = eig,
             model  = "components",
             legend = TRUE,
             ylab   = "Eigenvalues",
             xlab   = "Components",
             main   = "Parallel Analysis",
             ...
             )
```

Arguments

`parallel`	numeric: vector of the results of a previous parallel analysis
`eig`	depreciated parameter: eigenvalues to analyse (not used if x is used, recommended)
`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data
`model`	character: `"components"` or `"factors"`
`main`	character: title of the plot
`xlab`	character: label of the x axis
`ylab`	character: label of the y axis
`legend`	logical: indicator of the presence or not of a legend
`...`	variable: additionnal parameters to give to the `cor` or `cov` functions

Details

If `eig` is `FALSE` the plot shows only the parallel analysis without eigenvalues.

Value

Nothing returned.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

References

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/]

See Also

`plotuScree`, `nScree`, `plotnScree`, `parallel`

Examples

```
## SIMPLE EXAMPLE OF A PARALLEL ANALYSIS
## OF A CORRELATION MATRIX WITH ITS PLOT
 data(dFactors)
 eig      <- dFactors$Raiche$eigenvalues
 subject  <- dFactors$Raiche$nsubjects
 var      <- length(eig)
 rep      <- 100
 cent     <- 0.95
 results  <- parallel(subject,var,rep,cent)

 results


## PARALLEL ANALYSIS SCREE PLOT
 plotParallel(results, x=eig)
 plotParallel(results)
```

`plotuScree` *Plot of the Usual Cattell's Scree Test*

Description

`uScree` plot a usual scree test of the eigenvalues of a correlation matrix.

Usage

```
plotuScree(Eigenvalue,
            x       = Eigenvalue,
            model   = "components",
            ylab    = "Eigenvalues",
            xlab    = "Components",
            main    = "Scree Plot",
            ...
            )
```

Arguments

`Eigenvalue`	depreciated parameter: eigenvalues to analyse (not used if x is used, recommended)
`x`	numeric: a `vector` of eigenvalues, a `matrix` of correlations or of covariances or a `data.frame` of data
`model`	character: `"components"` or `"factors"`
`main`	character: title of the plot (default is `Scree Plot`)
`xlab`	character: label of the x axis (default is `Component`)
`ylab`	character: label of the y axis (default is `Eigenvalue`)
`...`	variable: additionnal parameters to give to the `eigenComputes` function

Value

Nothing returned with this function.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research, 1*, 245-276.

See Also

nScree, parallel

Examples

```
## SCREE PLOT
 data(dFactors)
 attach(dFactors)
 eig = Cliff1$eigenvalues
 plotuScree(x=eig)
```

---

principalAxis *Principal Axis Analysis*

Description

The PrincipalAxis function returns a principal axis analysis without iterated communalities estimates. Three different choices of communalities estimates are given: maximum corelation, multiple correlation or estimates based on the sum of the squared principal component analysis loadings. Generally statistical packages initialize the the communalities at the multiple correlation value (usual inverse or generalized inverse). Unfortunately, this strategy cannot deal with singular correlation or covariance matrices. If a generalized inverse, the maximum correlation or the estimated communalities based on the sum of loading are used instead, then a solution can be computed.

Usage

```
principalAxis(R,
                nFactors=2,
                communalities="component")
```

Arguments

`R`	numeric: correlation or covariance matrix
`nFactors`	numeric: number of factors to retain
`communalities`	character: initial values for communalities (`"component"`, `"maxr"`, `"ginv"` or `"multiple"`)

Value

`values`	numeric: variance of each component/factor
`varExplained`	numeric: variance explained by each component/factor
`varExplained`	numeric: cumulative variance explained by each component/factor
`loadings`	numeric: loadings of each variable on each component/factor

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

Kim, J.-O. and Mueller, C. W. (1978). *Introduction to factor analysis. What it is and how to do it*. Beverly Hills, CA: Sage.

Kim, J.-O. and Mueller, C. W. (1987). *Factor analysis. Statistical methods and practical issues*. Beverly Hills, CA: Sage.

See Also

componentAxis, iterativePrincipalAxis, rRecovery

Examples

```
# .......................................................
# Example from Kim and Mueller (1978, p. 10)
# Population: upper diagonal
# Simulated sample: lower diagnonal
 R <- matrix(c( 1.000, .6008, .4984, .1920, .1959, .3466,
                .5600, 1.000, .4749, .2196, .1912, .2979,
                .4800, .4200, 1.000, .2079, .2010, .2445,
                .2240, .1960, .1680, 1.000, .4334, .3197,
                .1920, .1680, .1440, .4200, 1.000, .4207,
                .1600, .1400, .1200, .3500, .3000, 1.000),
                nrow=6, byrow=TRUE)

# Factor analysis: Principal axis factoring
# without iterated communalities -
# Kim and Mueller (1978, p. 21)
# Replace upper diagonal with lower diagonal
 RU <- diagReplace(R, upper=TRUE)
 principalAxis(RU, nFactors=2, communalities="component")
 principalAxis(RU, nFactors=2, communalities="maxr")
```

```
 principalAxis(RU, nFactors=2, communalities="multiple")
# Replace lower diagonal with upper diagonal
 RL <- diagReplace(R, upper=FALSE)
 principalAxis(RL, nFactors=2, communalities="component")
 principalAxis(RL, nFactors=2, communalities="maxr")
 principalAxis(RL, nFactors=2, communalities="multiple")
# ...............................................
```

`principalComponents` *Principal Component Analysis*

Description

The `principalComponents` function returns a principal component analysis. Other R functions give the same results, but `principalComponents` is customized mainly for the other factor analysis functions available in the **nfactors** package. In order to retain only a small number of components the `componentAxis` function has to be used.

Usage

```
principalComponents(R)
```

Arguments

`R`	numeric: correlation or covariance matrix

Value

`values`	numeric: variance of each component
`varExplained`	numeric: variance explained by each component
`varExplained`	numeric: cumulative variance explained by each component
`loadings`	numeric: loadings of each variable on each component

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, http://www.er.uqam.ca/nobel/r17165/

References

Joliffe, I. T. (2002). *Principal components analysis* (2th Edition). New York, NJ: Springer-Verlag.

Kim, J.-O. and Mueller, C. W. (1978). *Introduction to factor analysis. What it is and how to do it*. Beverly Hills, CA: Sage.

Kim, J.-O. and Mueller, C. W. (1987). *Factor analysis. Statistical methods and practical issues*. Beverly Hills, CA: Sage.

See Also

componentAxis, iterativePrincipalAxis, rRecovery

Examples

```
# ..................................................
# Example from Kim and Mueller (1978, p. 10)
# Population: upper diagonal
# Simulated sample: lower diagnonal
 R <- matrix(c( 1.000, .6008, .4984, .1920, .1959, .3466,
               .5600, 1.000, .4749, .2196, .1912, .2979,
               .4800, .4200, 1.000, .2079, .2010, .2445,
               .2240, .1960, .1680, 1.000, .4334, .3197,
               .1920, .1680, .1440, .4200, 1.000, .4207,
               .1600, .1400, .1200, .3500, .3000, 1.000),
               nrow=6, byrow=TRUE)

# Factor analysis: Principal component -
# Kim et Mueller (1978, p. 21)
# Replace upper diagonal with lower diagonal
 RU <- diagReplace(R, upper=TRUE)
 principalComponents(RU)

# Replace lower diagonal with upper diagonal
 RL <- diagReplace(R, upper=FALSE)
 principalComponents(RL)
# ..................................................
```

---

rRecovery — *Test of Recovery of a Correlation or a Covariance matrix from a Factor Analysis Solution*

Description

The rRecovery function returns a verification of the quality of the recovery of the initial correlation or covariance matrix by the factor solution.

Usage

```
rRecovery(R, loadings, diagCommunalities=FALSE)
```

Arguments

R — numeric: initial correlation or covariance matrix

loadings — numeric: loadings from a factor analysis solution

diagCommunalities — logical: if TRUE, the correlation between the initial solution and the estimated one will use a correlation of one in the diagonal. If FALSE (default) the diagonal is not used in the computation of this correlation.

Value

`R` numeric: initial correlation or covariance matrix

`recoveredR` numeric: recovered estimated correlation or covariance matrix

`difference` numeric: difference between initial and recovered estimated correlation or covariance matrix

`cor` numeric: Pearson correlation between initial and recovered estimated correlation or covariance matrix. Computations depend on the logical value of the `communalities` argument.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

See Also

`componentAxis`, `iterativePrincipalAxis`, `principalAxis`

Examples

```
# ......................................................
# Example from Kim and Mueller (1978, p. 10)
# Population: upper diagonal
# Simulated sample: lower diagnonal
 R <- matrix(c( 1.000, .6008, .4984, .1920, .1959, .3466,
               .5600, 1.000, .4749, .2196, .1912, .2979,
               .4800, .4200, 1.000, .2079, .2010, .2445,
               .2240, .1960, .1680, 1.000, .4334, .3197,
               .1920, .1680, .1440, .4200, 1.000, .4207,
               .1600, .1400, .1200, .3500, .3000, 1.000),
               nrow=6, byrow=TRUE)


# Replace upper diagonal with lower diagonal
 RU         <- diagReplace(R, upper=TRUE)
 nFactors   <- 2
 loadings   <- principalAxis(RU, nFactors=nFactors,
                             communalities="component")$loadings
 rComponent <- rRecovery(RU,loadings, diagCommunalities=FALSE)$cor

 loadings   <- principalAxis(RU, nFactors=nFactors,
                             communalities="maxr")$loadings
 rMaxr      <- rRecovery(RU,loadings, diagCommunalities=FALSE)$cor

 loadings   <- principalAxis(RU, nFactors=nFactors,
                             communalities="multiple")$loadings
 rMultiple  <- rRecovery(RU,loadings, diagCommunalities=FALSE)$cor

 round(c(rComponent = rComponent,
         rmaxr      = rMaxr,
         rMultiple  = rMultiple), 3)
# ......................................................
```

structureSim *Population or Simulated Sample Correlation Matrix from a Given Factor Structure Matrix*

Description

The structureSim function returns a population and a sample correlation matrices from a predefined congeneric factor structure.

Usage

```
structureSim(fload, reppar=30, repsim=100, N, quantile=0.95,
             model="components", adequacy=FALSE, details=TRUE,
             r2limen=0.75, all=FALSE)
```

Arguments

fload	matrix: loadings of the factor structure
reppar	numeric: number of replications for the parallel analysis
repsim	numeric: number of replications of the matrix correlation simulation
N	numeric: number of subjects
quantile	numeric: quantile for the parallel analysis
model	character: "components" or "factors"
adequacy	logical: if TRUE prints the recovered population matrix from the factor structure
details	logical: if TRUE outputs details of the repsim simulations
r2limen	numeric: R2 limen value for the R2 Nelson index
all	logical: if TRUE computes the Bentler and Yuan index (very long computing time to consider)

Value

values	the output depends of the logical value of details. If FALSE, returns only statistics about the eigenvalues: mean, median, quantile, standard deviation, minimum and maximum. If TRUE, returns also details about the repsim simulations. If adequacy = TRUE returns the recovered factor structure

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

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

See Also

principalComponents, iterativePrincipalAxis, rRecovery

Examples

```
## Not run:
# .......................................................
# Example inspired from Zwick and Velicer (1986, table 2, p. 437)
## ...................................................................
 nFactors  <- 3
 unique    <- 0.2
 loadings  <- 0.5
 nsubjects <- 180
 repsim    <- 30
 zwick     <- generateStructure(var=36, mjc=nFactors, pmjc=12,
                                 loadings=loadings,
                                 unique=unique)
## ...................................................................

# Produce statistics about a replication of a parallel analysis on
# 30 sampled correlation matrices

 mzwick.fa <-  structureSim(fload=as.matrix(zwick), reppar=30,
                             repsim=repsim, N=nsubjects, quantile=0.5,
                             model="factors")

 mzwick    <-  structureSim(fload=as.matrix(zwick), reppar=30,
                             repsim=repsim, N=nsubjects, quantile=0.5, all=TRUE)

# Very long execution time that could be used only with model="components"
# mzwick    <-  structureSim(fload=as.matrix(zwick), reppar=30,
#                             repsim=repsim, N=nsubjects, quantile=0.5, all=TRUE)

 par(mfrow=c(2,1))
 plot(x=mzwick,    nFactors=nFactors, index=c(1:14), cex.axis=0.7, col="red")
 plot(x=mzwick.fa, nFactors=nFactors, index=c(1:11), cex.axis=0.7, col="red")
 par(mfrow=c(1,1))

 par(mfrow=c(2,1))
 boxplot(x=mzwick,    nFactors=3, cex.axis=0.8, vLine="blue", col="red")
 boxplot(x=mzwick.fa, nFactors=3, cex.axis=0.8, vLine="blue", col="red",
         xlab="Components")
 par(mfrow=c(1,1))
# ......................................................

## End(Not run)
```

structureSimObjectMethods

Utility Functions for nScree Class Objects

Description

Utility functions for `structureSim` class objects. Note that with the `plot.structureSim` a dotted black vertical line shows the median number of factors retained by all the different indices.

Usage

```
 ## S3 method for class 'structureSim':
boxplot (x, nFactors=NULL, eigenSelect=NULL,
                                 vLine="green", xlab="Factors",
                                 ylab="Eigenvalues", main="Eigen Box Plot", ...)
 ## S3 method for class 'structureSim':
is      (object)
 ## S3 method for class 'structureSim':
plot    (x, nFactors=NULL, index=NULL,
                                 main="Index Acuracy Plot", ...)
 ## S3 method for class 'structureSim':
print   (x, index=NULL, ...)
 ## S3 method for class 'structureSim':
summary (object, index=c(1:15), eigenSelect=NULL, ...)
```

Arguments

`eigenSelect`	numeric: vector of the index of the selected eigenvalues
`index`	numeric: vector of the index of the selected indices
`main`	character: main title
`nFactors`	numeric: if known, number of factors
`object`	structureSim: an object of the class `structureSim`
`vLine`	character: color of the vertical indicator line of the initial number of factors in the eigen boxplot
`x`	structureSim: an object of the class `structureSim`
`xlab`	character: x axis label
`ylab`	character: y axis label
`...`	variable: additionnal parameters to give to the `boxplot`, `plot`, `print` and `summary functions.`

Value

Generic functions for the `structureSim` class:

`boxplot.structureSim`
: graphic: plots an eigen boxplot

`is.structureSim`
: logical: is the object of the class `structureSim`?

`plot.structureSim`
: graphic: plots an index acuracy plot

`print.structureSim`
: numeric: data.frame of statistics about the number of components/factors to retain according to different indices following a `structureSim` simulation

`summary.structureSim`
: list: two data.frame, the first with the details of the simulated eigenvalues, the second with the details of the simulated indices

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
`<raiche.gilles@uqam.ca>`, `http://www.er.uqam.ca/nobel/r17165/`

References

Raiche, G., Riopel, M. and Blais, J.-G. (2006). *Non graphical solutions for the Cattell's scree test*. Paper presented at the International Annual meeting of the Psychometric Society, Montreal. [`http://www.er.uqam.ca/nobel/r17165/RECHERCHE/COMMUNICATIONS/`]

See Also

`nFactors-package`

Examples

```
## Not run:
## INITIALISATION
 library(xtable)
 library(nFactors)
 nFactors  <- 3
 unique    <- 0.2
 loadings  <- 0.5
 nsubjects <- 180
 repsim    <- 10
 var       <- 36
 pmjc      <- 12
 reppar    <- 10
 zwick     <- generateStructure(var=var, mjc=nFactors, pmjc=pmjc,
                                 loadings=loadings,
                                 unique=unique)

## SIMULATIONS
mzwick    <-  structureSim(fload=as.matrix(zwick), reppar=reppar,
                           repsim=repsim, details=TRUE,
                           N=nsubjects, quantile=0.5)

## TEST OF structureSim METHODS
 is(mzwick)
 summary(mzwick, index=1:5, eigenSelect=1:10, digits=3)
 print(mzwick, index=1:10)
 plot(x=mzwick, index=c(1:10), cex.axis=0.7, col="red")
 boxplot(x=mzwick, nFactors=3, vLine="blue", col="red")
```

```
## End(Not run)
```

---

studySim — *Simulation Study from Given Factor Structure Matrices and Conditions*

Description

The `structureSim` function returns statistical results from simulations from predefined congeneric factor structures. The main ideas come from the methodology applied by Zwick and Velicer (1986).

Usage

```
studySim(var, nFactors, pmjc, loadings, unique, N, repsim, reppar,
         stats=1, quantile=0.5, model="components", r2limen=0.75,
         all=FALSE, dir=NA, trace=TRUE)
```

Arguments

`var`	numeric: vector of the number of variables
`nFactors`	numeric: vector of the number of components/factors
`pmjc`	numeric: vector of the number of major loadings on each component/factor
`loadings`	numeric: vector of the major loadings on each component/factor
`unique`	numeric: vector of the unique loadings on each component/factor
`N`	numeric: vector of the number of subjects/observations
`repsim`	numeric: number of replications of the matrix correlation simulation
`reppar`	numeric: number of replications for the parallel and permutation analysis
`stats`	numeric: vector of the statistics to return: mean(1), median(2), sd(3), quantile(4), min(5), max(6)
`quantile`	numeric: quantile for the parallel and permutation analysis
`model`	character: `"components"` or `"factors"`
`r2limen`	numeric: R2 limen value for the R2 Nelson index
`all`	logical: if `TRUE` computes the Bentler and Yuan index (very long computing time to consider)
`dir`	character: directory where to save output. Default to NA
`trace`	logical: if `TRUE` outputs details of the status of the simulations

Value

`values`	Returns selected statistics about the number of components/factors to retain: mean, median, quantile, standard deviation, minimum and maximum.

Author(s)

Gilles Raiche
Centre sur les Applications des Modeles de Reponses aux Items (CAMRI)
Universite du Quebec a Montreal
<raiche.gilles@uqam.ca>, http://www.er.uqam.ca/nobel/r17165/

References

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

See Also

generateStructure, structureSim

Examples

```
## Not run:
# ......................................................................
# Example inspired from Zwick and Velicer (1986)
# Very long computimg time
# .....................................................................

# 1. Initialisation
# reppar    <- 30
# repsim    <- 5
# quantile  <- 0.50

# 2. Simulations
# X         <- studySim(var=36,nFactors=3, pmjc=c(6,12), loadings=c(0.5,0.8),
#                        unique=c(0,0.2), quantile=quantile,
#                        N=c(72,180), repsim=repsim, reppar=reppar,
#                        stats=c(1:6))

# 3. Results (first 10 results)
# print(X[1:10,1:14],2)
# names(X)

# 4. Study of the error done in the determination of the number
#    of components/factors. A positive value is associated to over
#    determination.
# results   <- X[X$stats=="mean",]
# residuals <- results[,c(11:25)] - X$nfactors
# BY        <- c("nsubjects","var","loadings")
# round(aggregate(residuals, by=results[BY], mean),0)

## End(Not run)
```

Index