

Package ‘psy’

April 19, 2009

Title Various procedures used in psychometry

Version 0.7

Author Bruno Falissard

Description Kappa, ICC, Cronbach alpha, screeplot, PCA, biplots, non linear mapping and related methods

Maintainer Bruno Falissard <falissard_b@wanadoo.fr>

License GPL

Repository CRAN

Date/Publication 2005-06-27 11:13:34

R topics documented:

ckappa	2
cronbach	3
expsy	4
fpca	5
icc	7
lkappa	8
mdspca	9
scree.plot	10
sleep	11
sphpca	12
wkappa	14
Index	16

 ckappa

Cohen's Kappa for 2 raters

Description

Computes Cohen's Kappa for agreement in the case of 2 raters. The diagnosis (the object of the rating) may have k possible values.

Usage

```
ckappa(r)
```

Arguments

`r` `n*2` matrix or dataframe, n subjects and 2 raters

Details

The function deals with the case where the two raters have not exactly the same scope of rating (some software associate an error with this situation). Missing value are omitted.

Value

A list with :

`$table` the 2*k table of raw data (first rater in rows, second rater in columns)
`$kappa` Cohen's Kappa

Author(s)

Bruno Falissard

References

Cohen, J. (1960), A coefficient of agreement for nominal scales, Educational and Psychological measurements, 20, 37-46.

Examples

```
data(expsy)
ckappa(expsy[,c(12,14)]) # Cohen's kappa for binary diagnosis

library(boot)
ckappa.boot <- function(data,x) {ckappa(data[x,])[[2]]}
res <- boot(expsy[,c(12,14)],ckappa.boot,1000)
quantile(res$t,c(0.025,0.975)) # two-sided bootstrapped confidence interval of kappa
boot.ci(res,type="bca") # adjusted bootstrap percentile (BCa) confidence interval
ckappa(expsy[,c(11,13)]) # Cohen's kappa for non binary diagnosis
```

`cronbach`*Cronbach's coefficient alpha*

Description

Computes the Cronbach's reliability coefficient alpha. This coefficient may be applied to a series of items destined to be aggregated in a single score. It estimates reliability in the framework of the domain sampling model.

Usage

```
cronbach(v1)
```

Arguments

```
v1          n*p matrix or dataframe, n subjects and p items
```

Details

Missing value are omitted in a "listwise" way (all items are removed even if only one of them is missing).

Value

A list with :

```
$sample.size  Number of subjects under study
$number.of.items  Number of items of the scale or questionnaire
$alpha        alpha
```

Author(s)

Bruno Falissard

References

Nunnally, J.C., Bernstein, I.H. (1994), Psychometric Theory, 3rd edition, McGraw-Hill Series in Psychology.

Examples

```
data(expsy)
cronbach(expsy[,1:10]) ## not good because item 2 is reversed (1 is high and 4 is low)
cronbach(cbind(expsy[,c(1,3:10)],-1*expsy[,2])) ## better

datafile <- cbind(expsy[,c(1,3:10)],-1*expsy[,2])
library(boot)
```

```

cronbach.boot <- function(data,x) {cronbach(data[x,])[[3]]}
res <- boot(datafile,cronbach.boot,1000)
quantile(res$t,c(0.025,0.975)) ## two-sided bootstrapped confidence interval of Cronbach's
boot.ci(res,type="bca") ## adjusted bootstrap percentile (BCa) confidence interval (

```

expsy

Data set related to psychometry

Description

The `expsy` data frame has 30 rows and 16 columns with missing data. `it1-it10` correspond to the rating of 30 patients with a 10 items scale. `r1, r2, r3` to the rating of item 1 by 3 different clinicians of the same 30 patients. `rb1, rb2, rb3` to the binary transformation of `r1, r2, r3` (1 or 2 -> 0; and 3 or 4 -> 1).

Usage

```
data(expsy)
```

Format

This data frame contains the following columns:

it1 a numeric vector corresponding to item 1 (rated from 1:low to 4:high)
it2 a numeric vector corresponding to item 2 (rated from 1:high to 4:low)
it3 a numeric vector corresponding to item 3 (rated from 1:low to 4:high)
it4 a numeric vector corresponding to item 4 (rated from 1:low to 4:high)
it5 a numeric vector corresponding to item 5 (rated from 1:low to 4:high)
it6 a numeric vector corresponding to item 6 (rated from 1:low to 4:high)
it7 a numeric vector corresponding to item 7 (rated from 1:low to 4:high)
it8 a numeric vector corresponding to item 8 (rated from 1:low to 4:high)
it9 a numeric vector corresponding to item 9 (rated from 1:low to 4:high)
it10 a numeric vector corresponding to item 10 (rated from 1:low to 4:high)
r1 a numeric vector corresponding to item 1 rated by rater 1
rb1 binary transformation of `r1`
r2 a numeric vector corresponding to item 1 rated by rater 2
rb2 binary transformation of `r2`
r3 a numeric vector corresponding to item 1 rated by rater 3
rb3 binary transformation of `r3`

Source

artificial data

Examples

```
data(expsy)
expsy[1:4,]
```

fpca

Focused Principal Components Analysis

Description

Graphical representation similar to a principal components analysis but adapted to data structured with dependent/independent variables

Usage

```
fpca(datafile, y, x, cx=0.75, namesvar=attributes(datafile)$names, pvalues="No", partial="Yes", input="Cor", contraction="Yes", sample.size)
```

Arguments

datafile	name of datafile
y	column number of the dependent variable
x	column numbers of the independent (explanatory) variables
cx	size of the lettering (0.75 by default, 1 for bigger letters, 0.5 for smaller)
namesvar	label of variables (names of columns by default)
pvalues	vector of prespecified pvalues (pvalues="No" by default) (see below)
partial	partial="Yes" by default, corresponds to the original method (see below)
input	input="Cor" for a correlation matrix (input="data" by default)
contraction	change the aspect of the diagram, contraction="Yes" is convenient for large data set (contraction="No" by default)
sample.size	to be specified if input="Cor"

Details

This representation is close to a Principal Components Analysis (PCA). Contrary to PCA, correlations between the dependent variable and the other variables are represented faithfully. The relationships between non dependent variables are interpreted like in a PCA: correlated variables are close or diametrically opposite (for negative correlations), independent variables make a right angle with the origin. The focus on the dependent variable leads formally to a partialisation of the correlations between the non dependent variables by the dependent variable (see reference). To avoid this partialisation, the option partial="No" can be used. It may be interesting to represent graphically the strength of association between the dependent variable and the other variables using p values coming from a model. A vector of pvalue may be specified in this case.

Value

A plot (q plots in fact).

Author(s)

Bruno Falissard, Bill Morpheu

References

Falissard B, Focused Principal Components Analysis: looking at a correlation matrix with a particular interest in a given variable. *Journal of Computational and Graphical Statistics* (1999), 8(4): 906-912.

Examples

```

data(sleep)
fpca(sleep,5,c(2:4,7:11))
## focused PCA of the duration of paradoxical sleep (dreams, 5th column)
## against constitutional variables in mammals (columns 2, 3, 4, 7, 8, 9, 10, 11).
## Variables inside the red cercle are significantly correlated
## to the dependent variable with p<0.05.
## Green variables are positively correlated to the dependent variable,
## yellow variables are negatively correlated.
## There are three clear clusters of independent variables.

corsleep <- as.data.frame(cor(sleep[,2:11],use="pairwise.complete.obs"))
fpca(corsleep,4,c(1:3,6:10),input="Cor",sample.size=60)
## when missing data are numerous, the representation of a pairwise correlation
## matrix may be preferred (even if mathematical properties are not so good...)

numer <- c(2:4,7:11)
l <- length(numer)
resu <- vector(length=l)
for(i in 1:l)
{
  int <- sleep[,numer[i]]
  mod <- lm(sleep$Paradoxical.sleep~int)
  resu[i] <- summary(mod)[[4]][2,4]*sign(summary(mod)[[4]][2,1])
}
fpca(sleep,5,c(2:4,7:11),pvalues=resu)
## A representation with p values
## When input="Cor" or pvalues="Yes" partial is turned to "No"

mod <- lm(sleep$Paradoxical.sleep~sleep$Body.weight+sleep$Brain.weight+
sleep$Slow.wave.sleep+sleep$Maximum.life.span+sleep$Gestation.time+
sleep$Predation+sleep$Sleep.exposure+sleep$Danger)
resu <- summary(mod)[[4]][2:9,4]*sign(summary(mod)[[4]][2:9,1])
fpca(sleep,5,c(2:4,7:11),pvalues=resu)
## A representation with p values which come from a multiple linear model
## (here results are difficult to interpret)

```

icc *Intraclass correlation coefficient (ICC)*

Description

Computes the ICC of several series of measurements, for example in an interrater agreement study. Two types of ICC are proposed: consistency and agreement.

Usage

```
icc(data)
```

Arguments

data n*p matrix or dataframe, n subjects p raters

Details

Missing data are omitted in a listwise way. The "agreement" ICC is the ratio of the subject variance by the sum of the subject variance, the rater variance and the residual; it is generally preferred. The "consistency" version is the ratio of the subject variance by the sum of the subject variance and the residual; it may be of interest when estimating the reliability of pre/post variations in measurements.

Value

A list with :

```
$nb.subjects  number of subjects under study
$nb.raters    number of raters
$subject.variance
               subject variance
$rater.variance
               rater variance
$residual     residual variance
$icc.consistency
               Intra class correlation coefficient, "consistency" version
$icc.agreement
               Intra class correlation coefficient, "agreement" version
```

Author(s)

Bruno Falissard

References

Shrout, P.E., Fleiss, J.L. (1979), Intraclass correlation: uses in assessing rater reliability, Psychological Bulletin, 86, 420-428.

Examples

```

data(expsy)
icc(expsy[,c(12,14,16)])

library(boot)
icc.boot <- function(data,x) {icc(data[x,])[[7]]}
res <- boot(expsy[,c(12,14,16)],icc.boot,1000)
quantile(res$t,c(0.025,0.975)) # two-sided bootstrapped confidence interval of icc (agreement)
boot.ci(res,type="bca") # adjusted bootstrap percentile (BCa) confidence interval (bca)

```

lkappa

Light's kappa for n raters

Description

Computes Light's Kappa for agreement in the case of n raters. The diagnosis (the object of the rating) may have k possible values (ordered or not).

Usage

```
lkappa(r, type="Cohen", weights="squared")
```

Arguments

r	m*n matrix, m subjects and n raters
type	type="Cohen" for a categorical diagnosis. If not, the diagnosis is supposed to be ordered
weights	weights="squared" for a weighted kappa with squared weights. If not, weights are computed with absolute values.

Details

Light's Kappa is equal to the mean of the $n(n-1)/2$ kappas obtained from each pair of raters. Missing values are omitted locally when considering each pair of raters. If type="Cohen" the diagnosis is considered as a categorical variable. If not, the diagnosis is considered as an ordered variable and weighted kappa's are computed. In this last situation, the type of weights that is used (squared or absolute values) is given by the variable weights.

Value

kappa (mean of the $n(n-1)/2$ kappas obtained from each pair of raters)

Author(s)

Bruno Falissard

References

Conger, A.J. (1980), Integration and generalisation of Kappas for multiple raters, *Psychological Bulletin*, 88, 322-328.

Examples

```
data(expsy)
lkappa(expsy[,c(11,13,15)])      # Light's kappa for non binary diagnosis
lkappa(expsy[,c(12,14,16)])      # Light's kappa for binary diagnosis
lkappa(expsy[,c(11,13,15)], type="weighted")  # Light's kappa for non binary ordered dia

library(boot)
lkappa.boot <- function(data,x) {lkappa(data[x,], type="weighted")}
res <- boot(expsy[,c(11,13,15)],lkappa.boot,1000)
quantile(res$t,c(0.025,0.975))    # Bootstrapped confidence interval of Light's kappa
boot.ci(res,type="bca")           # adjusted bootstrap percentile (BCa) confidence interval
```

mdspca	<i>Graphical representation of a correlation matrix using a Principal Component Analysis</i>
--------	--

Description

Similar to many routines, the interest is in the possible representation of both variables and subjects (and by the way categorical variables) with active and supplementary points. Missing data are omitted.

Usage

```
mdspca(datafile, supvar="no", supsubj="no", namesupvar=colnames(supvar),do.NULL=FALSE)
```

Arguments

datafile	name of datafile
supvar	matrix corresponding to supplementary variables (if any), supvar="no" by default
supsubj	matrix corresponding to supplementary subjects (if any), supsubj="no" by default
namesupvar	names of the points corresponding to the supplementary variables
namesupsubj	names of the points corresponding to the supplementary subjects
dimx	rank of the component displayed on the x axis (1 by default)
dimy	rank of the component displayed on the y axis (2 by default)
cx	size of the lettering (0.75 by default, 1 for bigger letters, 0.5 for smaller)

Value

A diagram (two diagrams if supplementary subjects are used)

Author(s)

Bruno Falissard

Examples

```
data(sleep)

mdspca(sleep[,c(2:5,7:11)])
## three consistent groups of variables, paradoxical sleep (in other words: dream)
## is negatively correlated with danger

mdspca(sleep[,c(2:5,7:11)], supvar=sleep[,6], namesupvar="Total.sleep", supsubj=sleep[,1], names
## Total.sleep is here a supplementary variable since it is deduced
## from Paradoxical.sleep and Slow.wave.sleep
## The variable Species is displayed in the subject plane,
## Rabbit and Cow have a high level of danger
```

scree.plot

Screeplot of eigenvalues, simulated data are available

Description

Graphical representation of the eigenvalues of a correlation/covariance matrix. Usefull to determine the dimensional structure of a set of variables. Simulation are proposed to help the interpretation.

Usage

```
scree.plot(namefile, title = "Scree Plot", type = "R", use = "complete.obs", simu =
```

Arguments

namefile	dataset
title	Title
type	type="R" to obtain the eigenvalues of the correlation matrix of dataset, type="V" for the covariance matrix, type="M" if the input data is directly the matrix, type="E" if the input data are directly the eigenvalues
use	omit missing values by default, use="P" to analyse the pairwise correlation/covariance matrix
simu	simu=p to add p screeplots of simulated random normal data (same number of patients and variables as in the original data set, same pattern of missing data if use="P")

Details

Simulations lead sometimes to underestimate the actual number of dimensions (as opposed to Kayser rule: eigen values superior to 1). Basically, simu=20 is enough.

Value

a plot

Author(s)

Bruno Falissard

References

Horn, JL (1965) A Rationale and Test for the Number of Factors in Factor Analysis, *Psychometrika*, 30, 179-185. Cattell, RB (1966) The scree test for the number of factors. *Multivariate Behavioral Research*, 1, 245-276.

Examples

```
data(expsy)
scree.plot(expsy[,1:10],simu=20,use="P") #no obvious structure with such a small sample
```

sleep

Ecological and Constitutional Data in Mammals

Description

Data from which conclusions were drawn in the article *Mammals: Ecological and Constitutional Correlates*, by Allison, T. and Cicchetti, D. (1976) *Science*, November 12, vol. 194, pp.732-734

Usage

```
data(sleep)
```

Format

This data frame contains the following columns:

Species a factor with levels

Body.weight a numeric vector, body weight in kg

Brain.weight a numeric vector, Brain weight in g

Slow.wave.sleep a numeric vector, nondreaming sleep (hrs/day)

Paradoxical.sleep a numeric vector, dreaming sleep (hrs/day)

Total.sleep a numeric vector, nondreaming + "dreaming" (hrs/day)

Maximum.life.span a numeric vector (in years)

Gestation.time a numeric vector (in days)

Predation a numeric vector, Predation index (1 min - 5 max)

Sleep.exposure a numeric vector, Sleep exposure index (1 min - 5 max)

Danger a numeric vector, Overall danger index (1 min - 5 max)

Source

<http://lib.stat.cmu.edu/datasets/sleep>

References

Mammals: Ecological and Constitutional Correlates, by Allison, T. and Cicchetti, D. (1976) Science, November 12, vol. 194, pp.732-734

Examples

```
data(sleep)
str(sleep)
```

sphpca

Spherical Representation of a Correlation Matrix

Description

Graphical representation of a correlation matrix, similar to principal component analysis (PCA) but the mapping is on a sphere. The information is close to a 3d PCA, the picture is however easier to interpret since the variables are in fact on a 2d map.

Usage

```
sphpca(datafile, h=0, v=0, f=0, cx=0.75, nbsphere=2, back=FALSE, input="data", method="PCA")
```

Arguments

datafile	name of datafile
h	rotation of the sphere on a horizontal plane (in degrees)
v	rotation of the sphere on a vertical plane (in degrees)
f	rotation of the sphere on a frontal plane (in degrees)
cx	size of the lettering (0.75 by default, 1 for bigger letters, 0.5 for smaller)
nbsphere	two by default: front and back
back	"FALSE" by default: the back sphere is not seen through
input	"data" by default: raw data are analysed, if not "data": correlation matrix is expected

method	"approx" by default: the estimation is based on a principal component analysis approximation. If "exact" the "approx" estimation is optimized (may be computationally consuming). if "rscal" a multidimensional scaling approach is used: distances between points on the sphere are optimized so that they represent at best the original correlations. The scaling that is used leads to angles on the sphere proportional to correlation between variables
maxiter	maximum number of iterations in the optim process
output	FALSE by default: if TRUE and method="rscal" numerical results are proposed

Details

There is an isophormism between a correlation matrix and points on the unit hypersphere of R_n . It can be shown that a 3d spherical representation of a correlation matrix is statistically and cognitively interesting (see reference).

The default option method="approx" is based on a principal components approximation (see reference). It is fast and gives rather good results.

If method="exact" the representation is slightly improved in terms of fit (the sphere minimizes the sum of squared distances between the original variables on the hypersphere and their projections on the sphere).

The option method="rscal" optimizes the representation of correlations between variables with distances between points (in a least squares sense). For convenience, the scaling of points on the sphere is chosen so that angles between points are linearly related to correlations between variables (this is not the case on the hypersphere were $d = [2 * (1 - r)]^{0.5}$).

For method="exact" or method="rscal" computations may be rather lengthy (and not sensible for more than 20-40 variables).

The sphere may be rotated to help in visualising most of variables on a same side (front for example).

By default, the back of the sphere (right plot) is not seen showing through.

Value

A plot. If method="rscal" and output=TRUE, a list with :

<code>\$stress.before.optim</code>	Stress before optimization. The stress is equal to the sum of squares of differences between distances on the 3d sphere and distances on the hypersphere.
<code>\$stress.after.optim</code>	Stress after optimization.
<code>\$convergence</code>	If 0, convergence is OK. If not, maxiter may be increased.
<code>\$correlations</code>	Correlation matrix of variables (Pearson).
<code>\$residuals</code>	Differences between observed correlations (hypersphere) and correlations estimated from points on the 3d sphere.
<code>\$mean.abs.resid</code>	Mean of absolute values of residuals.

Author(s)

Bruno Falissard

References

Falissard B, A spherical representation of a correlation matrix, *Journal of Classification* (1996), 13:2, 267-280.

Examples

```
data(sleep)
sphpca(sleep[,c(2:5,7:11)])
## spherical representation of ecological and constitutional correlates in mammals

sphpca(sleep[,c(2:5,7:11)],method="rscal",output=TRUE)
## idem, but optimizes the representation of correlations between variables with distances b

corsleep <- as.data.frame(cor(sleep[,c(2:5,7:11)],use="pairwise.complete.obs"))
sphpca(corsleep,input="Cor")
sphpca(corsleep,method="rscal",input="Cor")
## when missing data are numerous, the representation of a pairwise correlation
## matrix may be preferred (even if mathematical properties are not so good...)

sphpca(corsleep,method="rscal",input="Cor",h=180,f=180,nbsphere=1,back=TRUE)
## other option of presentation

##
# library(polycor)
# sleep$Predation <- as.ordered(sleep$Predation)
# sleep$Sleep.exposure <- as.ordered(sleep$Sleep.exposure)
# sleep$Danger <- as.ordered(sleep$Danger)
# corsleeph <- as.data.frame(hetcor(sleep[,c(2:5,7:11)])$correlations)
# sphpca(corsleeph,input="Cor",f=180)
# sphpca(corsleeph,method="rscal",input="Cor",f=180)
## --> Correlations between discrete variables may appear shoking to some statisticians (?)
## --> Representation of polychoric/polyserial correlations could be preferred in this situat
```

 wkappa

weighted Kappa for 2 raters

Description

Computes a weighted Kappa for agreement in the case of 2 raters. The diagnosis (the object of the rating) may have k possible ordered values.

Usage

```
wkappa(r,weights="squared")
```

Arguments

<code>r</code>	<code>n*2</code> matrix or dataframe, <code>n</code> subjects and 2 raters
<code>weights</code>	<code>weights="squared"</code> to obtain squared weights. If not, absolute weights are computed (see details)

Details

The diagnoses are ordered as follow: numbers < letters, letters and numbers ordered naturally. For `weights="squared"`, weights are related to squared differences between rows and columns indices (in this situation `wkappa` is close to an `icc`). For `weights!="squared"`, weights are related to absolute values of differences between rows and columns indices. The function deals with the case where the two raters have not exactly the same scope of rating (some software associate an error with this situation). Missing value are omitted.

Value

A list with :

<code>\$table</code>	the <code>2*k</code> table of raw data (first rater in rows, second rater in columns)
<code>\$weights</code>	"squared" or "absolute"
<code>\$kappa</code>	Weighted Kappa

Author(s)

Bruno Falissard

References

Cohen, J. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin* 70 (1968): 213-220.

Examples

```
data(expsy)
wkappa(expsy[,c(11,13)])          # weighted kappa (squared weights)

library(boot)
wkappa.boot <- function(data,x) {wkappa(data[x,])[[3]]}
res <- boot(expsy[,c(11,13)],wkappa.boot,1000)
quantile(res$t,c(0.025,0.975))    # two-sided bootstrapped confidence interval of weighted kap
boot.ci(res,type="bca")           # adjusted bootstrap percentile (BCa) confidence interval (b
```

Index

*Topic **datasets**

expsy, 3
sleep, 11

*Topic **multivariate**

fpca, 4
mdspca, 9
scree.plot, 10
sphpca, 12

*Topic **univar**

ckappa, 1
cronbach, 2
icc, 6
lkappa, 8
wkappa, 14

ckappa, 1
cronbach, 2

expsy, 3

fpca, 4

icc, 6

lkappa, 8

mdspca, 9

scree.plot, 10
sleep, 11
sphpca, 12

wkappa, 14