Package ‘psy’

October 14, 2022

Title Various Procedures Used in Psychometrics
Version 1.2
Description Kappa, ICC, reliability coefficient, parallel analysis, multi-trait multi-methods, spherical representation of a correlation matrix.
License GPL (>= 2)
Encoding UTF-8
RoxygenNote 7.1.2
NeedsCompilation no
Author Bruno Falissard [aut, cre]
Maintainer Bruno Falissard <falissard_b@wanadoo.fr>
Repository CRAN
Date/Publication 2022-04-21 22:00:02 UTC

R topics documented:

ckappa ................................................................. 2
cronbach ............................................................ 3
ehd ................................................................. 4
expsy ............................................................... 5
f pca .............................................................. 6
icc ................................................................. 8
lkappa ............................................................. 9
mdspca ........................................................... 11
mtmm ............................................................ 12
psy ............................................................... 13
scree.plot ..................................................... 14
sleep ......................................................... 15
sphcpca .......................................................... 16
wkappa .......................................................... 18

Index 20
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


Examples

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

##to obtain a 95%confidence interval:
#library(boot)
#ckappa.boot <- function(data,x) {ckappa(data[x,]][[2]])
#res <- boot(expsy[,c(12,14)],ckappa.boot,1000)
## two-sided bootstrapped confidence interval of kappa
#quantile(res$t,c(0.025,0.975))
## cronbach

### Adjusted Bootstrap Percentile (BCa) Confidence Interval (Better)

```r
#boot.ci(res,type="bca")
```

## Cohen's Kappa for Non-Binary Diagnosis

```r
#kappa(expsy[,c(11,13)])
```

---

**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

```r
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

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

# to obtain a 95% confidence interval:
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 alpha
boot.ci(res,type="bca")  ## adjusted bootstrap percentile (BCa) confidence interval (better)

---

<table>
<thead>
<tr>
<th>ehd</th>
<th>Depressive Mood Scale</th>
</tr>
</thead>
</table>

Description


Usage

data(ehd)

Format

This data frame contains the following columns:

c1  Observed painful sadness
c2  Emotional hyperexpressiveness
c3  Emotional instability
c4  Observed monotony
c5  Lack spontaneous expressivity
c6  Lack affective reactivity
c7  Emotional incontinence
c8  Affective hyperesthesia
c9  Observed explosive mood
c10 Worried gesture
c11 Observed anhedonia
c12 Felt sadness
c13 Situational anhedonia
c14 Felt affective indifference
<table>
<thead>
<tr>
<th>e15</th>
<th>Hypersensibility unpleasant events</th>
</tr>
</thead>
<tbody>
<tr>
<td>e16</td>
<td>Sensory anhedonia</td>
</tr>
<tr>
<td>e17</td>
<td>Felt affective monotony</td>
</tr>
<tr>
<td>e18</td>
<td>Felt hyperemotionalism</td>
</tr>
<tr>
<td>e19</td>
<td>Felt irritability</td>
</tr>
<tr>
<td>e20</td>
<td>Felt explosive mood</td>
</tr>
</tbody>
</table>

Source


Examples

```r
data(ehd)
str(ehd)
```

<table>
<thead>
<tr>
<th>expsy</th>
<th>Data set related to psychometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>it1</td>
<td>a numeric vector corresponding to item 1 (rated from 1:low to 4:high)</td>
</tr>
<tr>
<td>it2</td>
<td>a numeric vector corresponding to item 2 (rated from 1:high to 4:low)</td>
</tr>
<tr>
<td>it3</td>
<td>a numeric vector corresponding to item 3 (rated from 1:low to 4:high)</td>
</tr>
<tr>
<td>it4</td>
<td>a numeric vector corresponding to item 4 (rated from 1:low to 4:high)</td>
</tr>
<tr>
<td>it5</td>
<td>a numeric vector corresponding to item 5 (rated from 1:low to 4:high)</td>
</tr>
<tr>
<td>it6</td>
<td>a numeric vector corresponding to item 6 (rated from 1:low to 4:high)</td>
</tr>
<tr>
<td>it7</td>
<td>a numeric vector corresponding to item 7 (rated from 1:low to 4:high)</td>
</tr>
<tr>
<td>it8</td>
<td>a numeric vector corresponding to item 8 (rated from 1:low to 4:high)</td>
</tr>
<tr>
<td>it9</td>
<td>a numeric vector corresponding to item 9 (rated from 1:low to 4:high)</td>
</tr>
</tbody>
</table>
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(formula=NULL, y=NULL, x=NULL, data, cx=0.75, pvalues="No", partial="Yes", input="data", contraction="No", sample.size=1)

Arguments

- formula: "model" formula, of the form y ~ x
- y: column number of the dependent variable
- x: column numbers of the independent (explanatory) variables
- data: name of datafile
- cx: size of the lettering (0.75 by default, 1 for bigger letters, 0.5 for smaller)
- 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 Morphey, Adeline Abbe

References


Examples

data(sleep)
fpca(Paradoxical.sleep~Body.weight+Brain.weight+Slow.wave.sleep+Maximum.life.span+
   Gestation.time+Predation+Sleep.exposure+Danger,data=sleep)
fpca(y="Paradoxical.sleep",x=c("Body.weight","Brain.weight","Slow.wave.sleep",
   "Maximum.life.span","Gestation.time","Predation","Sleep.exposure","Danger"),data=sleep)
## 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(Paradoxical.sleep~Body.weight+Brain.weight+Slow.wave.sleep+Maximum.life.span+
   Gestation.time+Predation+Sleep.exposure+Danger, 
data=corsleep,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)
icc <- length(numer)
resu <- vector(length=icc)
for(i in 1:icc)
{
  int <- sleep[,numer[i]]
  mod <- lm(sleep$Paradoxical.sleep~int)
  resu[i] <- summary(mod)[[4]][2,4]*sign(summary(mod)[[4]][2,1])
}

fpca(Paradoxical.sleep~Body.weight+Brain.weight+Slow.wave.sleep+Maximum.life.span+
    Gestation.time+Predation+Sleep.exposure+Danger,
data=sleep,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(Paradoxical.sleep~Body.weight+Brain.weight+Slow.wave.sleep+Maximum.life.span+
    Gestation.time+Predation+Sleep.exposure+Danger,
data=sleep,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**

```r
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


Examples

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

#to obtain a 95% confidence interval:
library(boot)
#icc.boot <- function(data,x) {icc(data[,])[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 (better)

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

\( r \) \n\( m \times n \) matrix, \( m \) subjects and \( n \) raters

\( \text{type} \) \n\text{type=}"Cohen"\ for a categorical diagnosis. If not, the diagnosis is supposed to be ordered

\( \text{weights} \) \n\text{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 \text{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 \text{weights}.

Value

\text{kappa (mean of the } n(n-1)/2 \text{ kappas obtained from each pair of raters)}

Author(s)

Bruno Falissard

References


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 diagnosis

# to obtain a 95\% confidence interval:
# library(boot)
# lkappa.boot <- function(data,x) {lkappa(data[,], 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**

Graphical representation of a correlation matrix using a Principal Component Analysis

**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**

```r
mdspca(datafile, supvar="no", supsubj="no", namesupvar=colnames(supvar, do.NULL=FALSE), namesupsubj=colnames(supsubj, do.NULL=FALSE), dimx=1, dimy=2, cx=0.75)
```

**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**

```r
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], namesupsubj="",cx=0.5)
```
## 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

---

**mtmm**

*Multitrait Multimethod approach of scale validation*

### Description

This function is destined to assess the convergent and discriminant validity of subscales of a given scale. Items belonging to the same subscale should correlate highly amongst themselves. Items belonging to different subscales should not correlate highly. This approach is simpler and more robust than confirmatory factor analysis (CFA). It can be interesting to verify (at least approximately) the proposed structure of an existing instrument in a new population. Most psychometricians will however prefer CFA.

### Usage

```r
mtmm(datafile,x,color=FALSE,itemTot=FALSE,graphItem=FALSE,stripChart=FALSE,namesDim=NULL)
```

### Arguments

- `datafile` name of datafile
- `x` a list of variable names (as many elements as there are subscales)
- `color` boxplot are in colour: `FALSE` = colourless just in grey and white (by default), `TRUE` = with colours
- `itemTot` if `TRUE`, for subscale i (i=1,...,n), boxplot of Pearson’s correlations between total score of subscale i and the items of subscale j (j=1,...,n). If j=i, the item is omitted in the computation of the total score. If `FALSE`, for subscale i (i=1,...,n), boxplot of Pearson’s correlations between the items of subscale i and the items of subscale j (j=1,...,n)
- `graphItem` if `TRUE` represents graphically each correlation
- `stripChart` if `TRUE`, dot charts are preferred to boxplots. Used with small number of items
- `namesDim` Labels foreach boxplots

### Value

For subscale i (i=1,...,n), displays the n boxplots of the distributions of the Pearson’s correlations between items of subscale i and items of subscale j (j=1,...,n). If j=i, the correlation of a given item with itself is omitted. Boxplot for i=j (grey by default) should be above boxplots for i≠j. Likewise, the correlation of an item with the global score of its subscale should be above its correlations with the global score of the other subscales.
Examples

data(ehd)

par(mfrow=c(1,5))
mtmm(ehd, list(c("e15", "e18", "e19", "e20"), c("e4", "e5", "e6", "e14", "e17"), c("e11", "e13", "e16"), c("e1", "e10", "e12"), c("e2", "e3", "e7", "e8", "e9")))

# Boxplots of the distributions of the Pearson's correlations between total score of
# subscale i and the items of subscale j
par(mfrow=c(1,5))
mtmm(ehd, list(c("e15", "e18", "e19", "e20"), c("e4", "e5", "e6", "e14", "e17"), c("e11", "e13", "e16"), c("e1", "e10", "e12"), c("e2", "e3", "e7", "e8", "e9")))

# Pearson's correlations between total score of subscale i and all items
par(mfrow=c(3,2))
mtmm(ehd, list(c("e15", "e18", "e19", "e20"), c("e4", "e5", "e6", "e14", "e17"), c("e11", "e13", "e16"), c("e1", "e10", "e12"), c("e2", "e3", "e7", "e8", "e9")), graphItem=TRUE)

 psy Various procedures used in psychometry

Description

Kappa, Intra class correlation coefficient, Cronbach alpha, Scree plot, Multitraits multimethods, Spherical representation of a correlation matrix

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>psy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2009-12-23</td>
</tr>
<tr>
<td>License:</td>
<td>free</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>

Author(s)

Bruno Falissard <falissard_b@wanadoo.fr>
References

Horn, JL (1965) A Rationale and Test for the Number of Factors in Factor Analysis, Psychometrika, 30, 179-185.

Examples

data(sleep)
sphpca(sleep[,c(2:5,7:11)])
data(expsy)
scree.plot(expsy[,1:10],simu=20,use="P")
data(ehd)
par(mfrow=c(1,5))
mtmm(ehd,list(c("e15","e18","e19","e20"),c("e4","e5","e6","e14","e17"),c("e11","e13","e16"),c("e1","e10","e12"),c("e2","e3","e7","e8","e9")))

---

scree.plot

Screeplot of eigenvalues, simulated data are available

---

Description

Graphical representation of the eigenvalues of a correlation/covariance matrix. Useful 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 = "F")

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


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**


**Examples**

```r
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**

```r
sphpca(datafile, h=0, v=0, f=0, cx=0.75, nbsphere=2, back=FALSE, input="data", method="approx", maxiter=500, output=FALSE)
```

**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 :

$stress.before.optim
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.

$stress.after.optim
Stress after optimization.

$convergence
If 0, convergence is OK. If not, maxiter may be increased.

$correlations
Correlation matrix of variables (Pearson).

$residuals
Differences between observed correlations (hypersphere) and correlations estimated from points on the 3d sphere.

$mean.abs.resid
Mean of absolute values of residuals.

Author(s)

Bruno Falissard

References

Examples

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

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

corsleep <- as.data.frame(cor(sleep[,c(2:5,7:11)],use="pairwise.complete.obs"))
sphpcacorsleep,input="Cor")
sphpcacorsleep,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...)

sphpcacorsleep,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)
# sphpcacorsleeph,input="Cor",f=180)
# sphpcacorsleeph,method="rscal",input="Cor",f=180)
## --> Correlations between discrete variables may appear shoking to some statisticians (?)
## --> Representation of polychoric/polyserial correlations could be prefered in this situation

---

**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**

- `r` : n*2 matrix or dataframe, n subjects and 2 raters
- `weights` : weights="squared" to obtain squared weights. If not, absolute weights are computed (see details)
Details

Diagnoses have to be coded by 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 is supposed to deal with the case where the two raters have not exactly the same scope of rating. Missing value are omitted.

Value

A list with:

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

Author(s)

Bruno Falissard

References


Examples

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

#to obtain a 95% confidence interval:
#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 kappa
#boot.ci(res,type="bca") # adjusted bootstrap percentile (BCa) confidence interval (better)
Index

* datasets
  ehd, 4
  expsy, 5
  sleep, 15
* multivariate
  fpca, 6
  mdspca, 11
  scree.plot, 14
  sphpca, 16
* package
  psy, 13
* univar
  ckappa, 2
  cronbach, 3
  icc, 8
  lkappa, 9
  wkappa, 18

  ckappa, 2
  cronbach, 3
  ehd, 4
  expsy, 5
  fpca, 6
  icc, 8
  lkappa, 9
  mdspca, 11
  mtmm, 12
  psy, 13
  scree.plot, 14
  sleep, 15
  sphpca, 16
  wkappa, 18