Package 'concord'

February 9, 2012

Version 1.4-9

Date 2007-01-31

Title Concordance and reliability

Author Jim Lemon <jim@bitwrit.com.au>, Ian Fellows <ifellows@uscd.edu>

Maintainer Jim Lemon <jim@bitwrit.com.au>

Description Measures of concordance and reliability

License GPL (>= 2)

Repository CRAN

Date/Publication 2007-01-31 15:53:31

R topics documented:

BEZ *Z score approximation of contrast significance*

Description

calculates a Z score approximation of contrast significance

Usage

```
BEZ(rankarray,lambda)
```

Arguments

rankarray	matrix of ranks
lambda	contrast coefficient vector

Details

BEZ calculates a Z score approximation of the probability that the deviation of the mean rank scores from the contrast coefficient vector would occur given randomly allocated ranks. It is only useful in conjuction with the kendall.w function.

Value

Z	Z score

Author(s)

Jim Lemon

See Also

kendall.w

cohen.kappa *kappa reliability coefficient for nominal data*

Description

calculates the kappa coefficient of reliability for nominal data

Usage

```
cohen.kappa(classif,type=c("score","count"))
```

Arguments

classif	matrix of classification counts or scores
type	whether classif is an object by method matrix of scores or an object by category matrix of counts

Details

cohen.kappa will accept either an object by category matrix of counts in which the numbers represent how many methods have placed the object in each category, or an object by method matrix of categories in which the numbers represent each method's categorization of that object. The default is to assume scores and the operator **must** specify if counts are used. cohen.kappa reports two or three kappa values. If the classification matrix is composed of scores, the first is the original calculation from Cohen(1960) which does not assume equal classification proportions for the different methods. The next value is calculated as in Siegel & Castellan (1988) and uses pooled classification proportions. This method provides an adjustment for bias, where the different methods systematically differ in their categorization. The third value is adjusted for prevalence using the method proposed by Byrt, Bishop and Carlin (1993). An approximate distribution of this statistic does not seem to be available, so there is no approximation or probability reported.

Value

kappa.c	value of kappa (Cohen)
kappa.sc	value of kappa (Siegel & Castellan)
kappa.bbc	value of kappa (Byrt, Bishop & Carlin)
Zc	the Z-score approximation for kappa.c
Zsc	the Z-score approximation for kappa.sc
pc	the probability for Zc
psc	the probability for Zsc

Note

This is sometimes called Cohen's kappa. The name also avoids confusion with the kappa estimate of the conditioning number of a matrix. For a contingency table version of this statistic, see classAgreement in package e1071

Author(s)

Jim Lemon

References

Byrt, T., Bishop, J. & Carlin, J.B. (1993) Bias, Prevalence and Kappa. Journal of Clinical Epidemiology, 46(5): 423-429.

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

Siegel, S. & Castellan, N.J.Jr. (1988) Nonparametric statistics for the behavioral sciences. Boston, MA: McGraw-Hill.

See Also

scores.to.counts, wtpc

Examples

```
# the "C" data from Krippendorff
nmm<-matrix(c(1,1,NA,1,2,2,3,2,3,3,3,3,3,3,3,2,2,2,2,1,2,3,4,4,4,4,4,
1,1,2,1,2,2,2,2,NA,5,5,5,NA,NA,1,1,NA,NA,3,NA),nrow=4)
# first show the score to count transformation, remembering that
# Krippendorff's data is classifier by object and must be transposed
scores.to.counts(t(nmm))
# now calculate kappa - note that Cohen's method does not work with NAs
cohen.kappa(t(nmm),"score")
```

`coincidence.matrix` *calculate the concordance/discordance matrix for Krippendorff's alpha*

Description

calculates the coincidence matrix for Krippendorff's alpha.

Usage

```
coincidence.matrix(x)
```

Arguments

x	classifier by object matrix of classifications

Details

probably of no use by itself.

Value

`statistic`	always NA - value is calculated by `kripp.alpha`
`coincidence.matrix`	the concordance/discordance matrix used in the calculation of alpha
`data.values`	a character vector of the original data values
`nmatchval`	the count of matches, used in calculation
`data.level`	the data level is specified by `kripp.alpha`

Note

This is the initial step in calculating Krippendorff's alpha coefficient.

Author(s)

Jim Lemon

References

Krippendorff, K. (1980) Content analysis: An introduction to its methodology. Beverly Hills, CA: Sage.

See Also

`kripp.alpha`

Examples

```
# the "C" data from Krippendorff
nmm<-matrix(c(1,1,NA,1,2,2,3,2,3,3,3,3,3,3,3,3,2,2,2,2,1,2,3,4,4,4,4,4,
1,1,2,1,2,2,2,2,NA,5,5,5,NA,NA,1,1,NA,NA,3,NA),nrow=4)
coincidence.matrix(nmm)
```

`kendall.w` *Kendall's W coefficient of concordance*

Description

calculates Kendall's W coefficient of concordance

Usage

```
kendall.w(x,lambda,descending=TRUE,ranks=FALSE)
```

Arguments

x	matrix of scores or ranks
lambda	optional contrast coefficient vector or matrix
descending	whether high (default) or low scores represent top ranks
ranks	whether the values in x are scores or ranks

Details

kendall.w will accept either a matrix or data frame of scores or ranks where the rows represent methods (usually raters) and the columns represent data objects. It will handle ties, but not missing values. By default it assumes that numerically higher scores represent numerically lower ranks. A vector or matrix of contrast coefficients (which each must sum to zero) may be supplied. A z-score approximation of the significance of each contrast will be displayed if lambda is present. The ranks argument allows the user to pass ranks directly to the function. If ranks are passed without setting ranks to TRUE and with descending TRUE, the order of the ranks will be reversed. For small values of k (methods), kendall.w will try to lookup the tabled values for significance. For k greater than 7, a chi-squared approximation is returned. Only one of these values will be returned.

Value

W	value of W
p.table	whether the obtained W exceeded the table value for small N
p.chisq	the probability of the obtained chi-squared value for larger N

Note

Kendall's W may not be appropriate for nominal class data.

Author(s)

Jim Lemon

References

Siegel, S. & Castellan, N.J.Jr. (1988) Nonparametric statistics for the behavioral sciences. Boston, MA: McGraw-Hill.

See Also

kripp.alpha,cohen.kappa

Examples

```
# fictional rankings of job applicants
app.mat<-matrix(c(1,3,4,2,6,5,2,4,3,1,5,6,3,2,5,1,5,4),nrow=3,byrow=TRUE)
# Test the hypothesis that the first three applicants are ranked higher
# than the last three.
lambda<-c(1,1,1,-1,-1,-1)
print(kendall.w(app.mat,lambda))
```

`kripp.alpha` *calculate Krippendorff's alpha reliability coefficient*

Description

calculates the alpha coefficient of reliability proposed by Krippendorff

Usage

```
kripp.alpha(x, method="nominal")
```

Arguments

`x`	classifier by object matrix of classifications
`method`	data level of x

Value

`statistic`	value of alpha
`coincidence.matrix`	the concordance/discordance matrix used in the calculation of alpha
`data.values`	a character vector of the unique data values
`nmatchval`	the count of matches, used in calculation

Note

Krippendorff's alpha coefficient is particularly useful where the level of measurement of classification data is higher than nominal or ordinal.

Author(s)

Jim Lemon

References

Krippendorff, K. (1980) Content analysis: An introduction to its methodology. Beverly Hills, CA: Sage.

See Also

`coincidence.matrix`

Examples

```
# the "C" data from Krippendorff
nmm<-matrix(c(1,1,NA,1,2,2,3,2,3,3,3,3,3,3,3,2,2,2,2,1,2,3,4,4,4,4,4,
1,1,2,1,2,2,2,2,NA,5,5,5,NA,NA,1,1,NA,NA,3,NA),nrow=4)
# first assume the default nominal classification
kripp.alpha(nmm)
# now use the same data with the other three methods
kripp.alpha(nmm,"ordinal")
kripp.alpha(nmm,"interval")
kripp.alpha(nmm,"ratio")
```

`mcnemar.mh` *2 way reliability coefficient for nominal data*

Description

calculates a coefficient of reliability for nominal data

Usage

```
mcnemar.mh(x)
```

Arguments

x	2x2 classification matrix or matrix of dichotomous classification scores.

Details

`mcnemar.mh` calculates a reliability coefficient for two raters classifying n objects into two categories. It will accept either a 2x2 classification matrix of counts of objects falling into two categories or a 2xn or nx2 matrix of classification scores.

Value

kappa	value of kappa
Z	the Z-score approximation
p	the probability of Z

Author(s)

Jim Lemon

References

Siegel, S. & Castellan, N.J.Jr. (1988) Nonparametric statistics for the behavioral sciences. Boston, MA: McGraw-Hill.

See Also

stuart.maxwell.mh

Examples

```
xt<-table(sample(0:1,50,TRUE),sample(0:1,50,TRUE))
mcnemar.mh(xt)
```

N.cohen.kappa *Sample Size Calculation for Cohen's Kappa Statistic*

Description

This function is a sample size estimator for the Cohen's Kappa statistic for a binary outcome. Note that any value of "hypokappa" in the interval [0,1] is acceptable (i.e. Kappa=0 is a valid null hypothesis).

Usage

```
N.cohen.kappa(rate1,rate2,kappa,hypokappa,power=.8,alpha=.05,twosided=FALSE)
```

Arguments

rate1	The probability that the first rater will record a positive diagnosis
rate2	The probability that the second rater will record a positive diagnosis
kappa	The true Cohen's Kappa statistic
hypokappa	The value of kappa under the null hypothesis
alpha	Type I error of test
power	The desired power to detect the difference between kappa and hypokappa
twosided	TRUE if test is two-sided

Value

returns required sample size

Author(s)

Ian Fellows <ifellows@ucsd.edu>

References

Cantor, A. B. (1996) Sample-size calculation for Cohen's kappa. Psychol. Method., 1, 150-153.

Examples

```
# Testing H0: kappa = 0.7 vs. HA: kappa > 0.7 given that
# kappa = 0.85 and both raters classify 50% of subjects as positive.
N.cohen.kappa(0.5,0.5,0.7,0.85)
```

`page.trend.test` *Page test for ordered alternatives*

Description

calculates the Page test for ordered alternatives.

Usage

```
page.trend.test(x)
```

Arguments

`x`	a 2D matrix of scores

Details

`page.trend.test` will accept a matrix of scores where the rows represent methods (usually raters) and the columns represent related data objects. It apparently handles ties, but not missing values. By default it assumes that numerically higher scores represent numerically lower ranks. For small values of k (methods) or N (data objects), `page.trend.test` will try to look up the tabled values (as in Siegel & Castellan (1988) for significance. For `k,N > 3,20` or `k,N > 4-10,12`, a normal approximation is returned. Only one of these values will be returned.

Value

`ranks`	matrix of ranks
`mean.ranks`	mean ranks of data objects
`L`	value of the L statistic
`p.table`	whether the obtained L exceeded the table value for small k,N
`Z`	The normal approximation for larger k,N
`pZ`	the probability of the obtained normal value for larger k,N

Note

The Page test for ordered alternatives is slightly more powerful than the Friedman analysis of variance by ranks.

Author(s)

Jim Lemon

References

Siegel, S. & Castellan, N.J.Jr. (1988) Nonparametric statistics for the behavioral sciences. Boston, MA: McGraw-Hill.

Examples

```
# Craig's data from Siegel & Castellan, p 186
soa.mat<-matrix(c(.797,.873,.888,.923,.942,.956,
 .794,.772,.908,.982,.946,.913,
 .838,.801,.853,.951,.883,.837,
 .815,.801,.747,.859,.887,.902),nrow=4,byrow=TRUE)
page.trend.test(soa.mat)
```

---

print.cohen.kappa *prints kappa reliability coefficient for nominal data*

Description

prints the kappa coefficient and associated probability

Usage

```
 ## S3 method for class 'cohen.kappa'
print(x,...)
```

Arguments

x	an object of class "cohen.kappa" returned from cohen.kappa
...	arguments to be passed to print

Value

nil

Author(s)

Jim Lemon

---

print.kendall.w *displays Kendall's W information*

Description

prints the W coefficient, associated probability and optionally a table of ranks

Usage

```
 ## S3 method for class 'kendall.w'
print(x,...)
```

Arguments

x	an object of class "kendall.w" returned from kendall.w
...	arguments to be passed to print

Value

nil

Author(s)

Jim Lemon

---

print.kripp.alpha *displays Krippendorff's alpha information*

Description

prints the Krippendorff's alpha coefficient

Usage

```
 ## S3 method for class 'kripp.alpha'
print(x,...)
```

Arguments

x	an object of class "kripp.alpha" returned from kripp.alpha
...	arguments to be passed to print

Value

nil

Author(s)

Jim Lemon

print.page.trend.test *prints the L statistic for Page's trend test*

Description

prints the obtained L statistic and the associated probability for the normal approximation if the sample sized is sufficiently large

Usage

```
 ## S3 method for class 'page.trend.test'
print(x,...)
```

Arguments

x	an object returned from page.trend.test
...	arguments to be passed to print

Value

nil

Author(s)

Jim Lemon

print.quest.reliability *Print questionnaire reliability measures*

Description

Prints the values returned from quest.reliability

Usage

```
 ## S3 method for class 'quest.reliability'
print(x,...)
```

Arguments

x	a list of results from quest.reliability
...	dummy argument to match print(x,...)

Author(s)

Jim Lemon

See Also

print

---

print.stuart.maxwell *displays Stuart-Maxwell marginal homogeneity test information*

Description

prints the information from stuart.maxwell.mh

Usage

```
 ## S3 method for class 'stuart.maxwell'
print(x,...)
```

Arguments

x	an object of class "stuart.maxwell" returned from stuart.maxwell
...	arguments to be passed to print

Value

nil

Author(s)

Jim Lemon

---

quest.reliability *Questionnaire reliability measures*

Description

Calculates a few common measures of questionnaire reliability

Usage

```
quest.reliability(x,nsplits=10)
```

Arguments

x	a matrix of numeric scores with subjects as rows and items as columns
nsplits	the number of splits to make in calculating split-half reliability

Details

quest.reliability will only handle numeric scores. Character scores can be transformed into numeric using as.numeric(as.factor(... but take care that the transformation preserves the order of the scores. Any rows with NA values in the score matrix will be removed before the calculations are made.

Value

A list containing five components:

cronbach.alpha	Cronbach's alpha for the entire scale.
split.half	An approximate split-half reliability is calculated by dividing the items into halves on a random basis and calculating the correlation between the summed scores for the halves. This is repeated nsplits times.
item.whole	The correlation of each item with the sum of the remaining items is calculated.
item.alpha	Cronbach's alpha is calculated for each item on all other items, i.e. the scale alpha if that item were dropped.
item.var	The variance of scores for each item is calculated. Items with relatively low variance are likely to be poor discriminators.

Author(s)

Jim Lemon

See Also

cor,var

Examples

```
# fake a data set with one bad item
fakedata<-matrix(c(rep(c(2,2:9,9),9)+sample(c(-1,0,1),90,TRUE),
 rep(5,10)+sample(c(-1,0,1),10,TRUE)),nrow=10)
quest.reliability(fakedata)
```

`rater.bias` *coefficient of rater bias*

Description

calculates a coefficient of systematic bias between two raters

Usage

```
rater.bias(x)
```

Arguments

`x`	CxC classification matrix or matrix of classification scores into C categories.

Details

`rater.bias` calculates a reliability coefficient for two raters classifying n objects into any number of categories. It will accept either a cxc classification matrix of counts of objects falling into c categories or a 2xn or nx2 matrix of classification scores.

Value

`statistic`	value of coefficient
`p`	the probability of the coefficient as a df 1 Chi-square variable

Author(s)

Jim Lemon

References

Bishop Y.M.M., Fienberg S.E. & Holland P.W. (1975) Discrete multivariate analysis: theory and practice. Cambridge, Massachusetts: MIT Press.

See Also

`mcnemar.mh`

Examples

```
# fake a 2xn matrix of three way classification scores
ratings<-matrix(sample(1:3,60,TRUE),nrow=2)
rater.bias(ratings)
```

`scores.to.counts` *transform a score matrix to a count matrix*

Description

transforms an object by classifier matrix of classification scores to an object by category matrix of classification counts

Usage

```
scores.to.counts(scores)
```

Arguments

`scores`	object by classifier matrix of classification scores

Details

usually called by `cohen.kappa`

Value

`counts`	an object by category matrix of counts
`scores.to.counts`	the concordance/discordance matrix used in the calculation of alpha
`data.values`	a character vector of the original data values
`nmatchval`	the count of matches, used in calculation

Note

There seems to be no established convention for whether objects(cases) should be rows (as they are here) or columns. Make sure to transpose as below if necessary.

Author(s)

Jim Lemon

See Also

`cohen.kappa`

Examples

```
# the "C" data from Krippendorff
nmm<-matrix(c(1,1,NA,1,2,2,3,2,3,3,3,3,3,3,3,2,2,2,2,1,2,3,4,4,4,4,4,
1,1,2,1,2,2,2,2,NA,5,5,5,NA,NA,1,1,NA,NA,3,NA),nrow=4)
# remember to transpose the data to the expected format
scores.to.counts(t(nmm))
```

stickleback *Estimates of nuptial coloration of male sticklebacks*

Description

Agreement of classification of 4 raters of 29 male sticklebacks into 5 color categories.

Usage

```
data(stickleback)
```

Source

Rowland, W. J. (1984) The relationships among nuptial coloration, aggression and courtship of male three-spined sticklebacks, *Gasterosetus aculeatus*.*Canadian Journal of Zoology*, **62**, 999-1004.

stuart.maxwell.mh *between rater concordance coefficient for classification*

Description

calculates the coefficient of concordance for two raters

Usage

```
stuart.maxwell.mh(x)
```

Arguments

x	cxc classification matrix or matrix of classification scores into c categories.

Details

`stuart.maxwell.mh` calculates a reliability coefficient for two raters classifying n objects into any number of categories. It will accept either a cxc classification matrix of counts of objects falling into c categories or a cxn or nxc matrix of classification scores.

Value

statistic	value of coefficient
p	the probability of the coefficient as a Chi-square variable

Author(s)

Jim Lemon

References

Stuart A.A. (1955) A test for homogeneity of the marginal distributions in a two-way classification. Biometrika, 42, 412-416.

Maxwell A.E. (1970) Comparing the classification of subjects by two independent judges. British Journal of Psychiatry, 116, 651-655.

See Also

rater.bias

Examples

```
# fake a 2xn matrix of three way classification scores
ratings<-matrix(sample(1:3,60,TRUE),nrow=2)
stuart.maxwell.mh(ratings)
```

---

tiecorr *correction for tied ranks*

Description

calculates a correction for tied ranks

Usage

```
tiecorr(rankarray)
```

Arguments

rankarray	matrix of ranks

Details

tiecorr calculates a correction for tied ranks that is appropriate for Kendall's W and the Spearman rank-order correlation coefficient.

Value

tie3margsum	correction factor

Author(s)

Jim Lemon

References

Siegel, S. & Castellan, N.J.Jr. (1988) Nonparametric statistics for the behavioral sciences. Boston, MA: McGraw-Hill.

See Also

kendall.w

---

wtpc *calculate weighted percentages for nominal classification data*

Description

calculates weighted percentages for nominal classification data

Usage

```
wtpc(x, n.methods, n.objects, type=c("count","score"))
```

Arguments

x	vector, data frame or matrix of ratings
n.methods	the number of methods (e.g. raters) used
n.objects	the number of objects rated or classified
type	whether the data represent counts of objects in categories or scores for objects by methods

Details

wtpc will accept a vector (if only one method is used) matrix or data frame in which values represent counts of objects (rows) classified as categories (columns), or the same objects in which values represent categorizations of those objects. The default is to assume counts and the operator **must** specify if scores are used. The formula used is: `wtpc = 100/n.methods * sum(category counts)/n.objects`.

Value

wtpc	vector of weighted percentages of occurrence of categories

Note

This is a convenience function for calculating the frequencies of category occurrences in ratings where all methods may not recognize the same objects as belonging to the available categories.

Author(s)

Jim Lemon

Examples

```
# the "C" data from Krippendorff
nmm<-matrix(c(1,1,NA,1,2,2,3,2,3,3,3,3,3,3,3,2,2,2,2,1,2,3,4,4,4,4,4,
1,1,2,1,2,2,2,2,NA,5,5,5,NA,NA,1,1,NA,NA,3,NA),nrow=4)
# Krippendorff's data is classifier by object and must be transposed
wtpc(t(nmm),4,12,"score")
```

Index