

Package ‘rmac’

February 15, 2012

Type Package

Title calculate RMAC or FMAC agreement coefficients

Version 0.9

Date 2010-06-02

Author Jennifer Kirk

Maintainer Jennifer Kirk <jenn.kirk@nih.gov>

Depends boot

Description Calculate Cohen’s kappa, Scott’s Pi, or the RMAC concordance correlation coefficient

License GPL-2

Repository CRAN

Date/Publication 2010-08-31 05:57:28

R topics documented:

rmac-package	2
cac	3
rmacBoot	5
wkappa	7

Index	10
--------------	-----------

Description

This package consists of four functions for calculating fixed and random marginal agreement coefficients based on the paper Fay (2005). The fixed marginal agreement coefficient (FMAC) is the traditional agreement coefficient, which assumes that each measurement's marginal distribution is the best estimate of its true distribution for the calculation of expected agreement due to chance. Examples of the fixed marginal agreement coefficient include the weighted kappa and the concordance correlation coefficient. In certain cases, fixed marginal agreement coefficients will indicate a high degree of agreement when the marginal distributions are extremely unbalanced (Feinstein and Cicchetti 1989). As an alternative to this, the Fay (2005) paper proposes the random marginal agreement coefficient (RMAC), which assumes that a mixture distribution of the two measurement distributions is the best estimate of both measurements' true distributions. For example, the paper uses the Westlund and Kurkland (1953) multiple sclerosis data. Patients were rated for likelihood of multiple sclerosis on a scale from 1 to 4 by two different neurologists. As the code below (see example section) shows, when the original data (`msd1`) is modified so that the 10 patients neurologist 1 awarded a score of "3" and neurologist 2 awarded a score of "1" receive a score of "1" from neurologist 1 and a score of "3" from neurologist 2, the FMAC changes, while the RMAC is the same.

This package provides a general method for calculating the random marginal agreement coefficient in `rmacBoot` and a general method for calculating the fixed marginal agreement coefficient in `fmacBoot`. For categorical data, `wkappa` provides the FMAC, which is equivalent to Cohen's kappa, and the RMAC, which is equivalent to Scott's pi for the 2 x 2 case. For continuous data, `cac` provides the fixed marginal agreement coefficient, which with the squared difference cost function is equivalent to Lin's concordance correlation coefficient, and the random marginal agreement coefficient, presented in Fay (2005).

Details

Package: rmac
Type: Package
Version: 0.9
Date: 2008-05-31
License: GPL2

Author(s)

Jennifer Kirk Maintainer: Jennifer Kirk <jenn.kirk@nih.gov>

References

- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.*, 20: 37-46.
- Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychol. Bull.*, 70: 213-220.
- Fay, M.P. (2005). Random marginal agreement coefficients: Rethinking the adjustment for chance in agreement coefficients. *Biostatistics*, 6: 171-180.
- Feinstein, A. R. and Cicchetti, D. V. (1989). High agreement but low kappa: I. the problems of two paradoxes. *J. Clin. Epidemiol.* 43: 543-549.
- Lin, L. I. (1989) A concordance correlation coefficient to evaluate reproducibility. *Biometrics*. 45: 255-268. (Correction: 200, pp. 324-325)
- Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quart.*, 19: 321-325.
- Westlund, K. B. and Kurkland, L. T. (1953). Studies in multiple sclerosis in Winnipeg, Manitoba and New Orleans, Louisiana. *American Journal of Hygiene*. 57: 380-396.

See Also

[wkappa](#), [cac](#), [rmacBoot](#)

Examples

```
#The Multiple Sclerosis Diagnoses Example (from Fay 2005)
#the original data
msd1<- as.table(matrix(data = c(38,5,0,1,33,11,3,0,10,14,5,6,3,7,3,10), 4,4, byrow = TRUE))
msd1
#the data with cell counts (1,3) and (3,1) reversed
msd2<- as.table(matrix(data = c(38,5,10,1,33,11,3,0,0,14,5,6,3,7,3,10), 4,4, byrow = TRUE))
msd2

#calculate the FMAC of each data set
wkappa(msd1, method = "fmac")
wkappa(msd2, method = "fmac")

#calculate the FMAC of each data set
wkappa(msd1, method = "rmac")
wkappa(msd2, method = "rmac")
```

cac

A General Method for Calculating the RMAC and FMAC for Continuous Data

Description

Calculates the FMAC or RMAC for continuous data sets. When the squared difference function is used with the FMAC, it is equivalent to Lin's concordance correlation coefficient.

Usage

```
cac(x, type = c("absolute", "squared"), method = c("fmac", "rmac"), alternative = c("two.sided", "less"
```

Arguments

<code>x</code>	matrix or data frame of responses, where the responses are in columns 1 and 2
<code>type</code>	either "absolute" or "squared"; indicates the cost function
<code>method</code>	indicates which method of calculating the agreement coefficient to use
<code>alternative</code>	either "two.sided", "less", "greater"; indicates the t-test if using the squared difference function
<code>conf.level</code>	confidence level for interval
<code>na.rm</code>	logical, remove missing values for both if missing response for either
<code>numr</code>	integer indicating the number of bootstrapping samples to use (R)

Details

The function assumes that the input vectors are the first two columns of `x`. This function calculates the FMAC and RMAC for continuous data sets using one of two built in cost functions. If the squared difference cost function is used with the RMAC, this function uses an optimized method. Otherwise, this function uses the general methods, `fmacBoot` and `rmacBoot`.

The confidence intervals are calculated using the BCa method (as per `fmacBoot` and `rmacBoot`), unless the squared difference cost function is used, in which case the delta method is used (See the supplement for Fay (2005) for a description.).

Value

A list with class "htest" containing the following components is returned:

<code>method</code>	a character string describing the statistical method used
<code>statistic</code>	the value of the test statistic with a name describing it
<code>conf.int</code>	a confidence interval for the agreement coefficient
<code>estimate</code>	an estimate of the agreement coefficient
<code>alternative</code>	a character string describing the alternative hypothesis
<code>p.value</code>	the p-value for the test
<code>data.name</code>	a character string giving the names of the data

Author(s)

Jennifer Kirk (using functions written by M.P. Fay)

References

- Fay, M.P. (2005). Random marginal agreement coefficients: Rethinking the adjustment for chance in agreement coefficients. *Biostatistics*, 6: 171-180.
- Lin, L. I. (1989) A concordance correlation coefficient to evaluate reproducibility. *Biometrics*. 45: 255-268. (Correction: 2000, pp. 324-325)

See Also

[rmac-package](#), [rmacBoot](#), [wkappa](#)

Examples

```
# A simple example
set.seed(12321)
x<-rnorm(10,1+(1:10))
y<-rnorm(10,(1:10))

plot(x,y,xlim=c(0,12),ylim=c(0,12))
lines(c(0,12),c(0,12))

example<- cbind(x,y)

cac(example, type = "absolute", method = "fmac")
cac(example, type = "absolute", method = "rmac")
#cor.test(x,y,alternative="two.sided")
```

rmacBoot

A General Method for Calculating the RMAC and FMAC using BCa Bootstrapping

Description

Calculates the RMAC and FMAC using the ideal bootstrap estimate and calculates the confidence interval using BCa bootstrapping, given a specific cost function.

Usage

```
rmacBoot(x,...)
fmacBoot(x,...)

## Default S3 method:
rmacBoot(x,cost, conf.level= 0.95, na.rm = FALSE, numr = 999,...)
## S3 method for class 'table'
rmacBoot(x,cost, conf.level= 0.95, na.rm = FALSE, numr = 999,...)

## Default S3 method:
fmacBoot(x,cost, conf.level= 0.95, na.rm = FALSE, numr = 999, ...)
## S3 method for class 'table'
fmacBoot(x,cost, conf.level= 0.95, na.rm = FALSE, numr = 999, ...)
```

Arguments

<code>x</code>	table, matrix or data frame of responses, where the responses are in columns 1 and 2 of the matrix or data frame
<code>cost</code>	a function that defines the cost function
<code>conf.level</code>	confidence level for interval
<code>na.rm</code>	logical, remove missing values for both if missing response for either
<code>numr</code>	integer indicating the number of bootstrapping samples to use (R)
<code>...</code>	any other arguments passed to the function; not currently used

Details

For categorical data, these functions accept data in table form, ie. a table of counts. If the data is in a table, it is first converted to two vectors of scores before the agreement coefficient is calculated. The default methods assumes that the data are in the first two columns of `x`.

If the two input vectors are not the same length, an error is returned.

The cost function should have two inputs, corresponding to individual values from the two input vectors. See below for an example of a cost function.

The confidence intervals are calculated using the built-in R function `boot.out` with the `BCa` option.

Value

A list with class "htest" containing the following components is returned:

<code>method</code>	a character string describing the statistical method used
<code>statistic</code>	the value of the test statistic with a name describing it
<code>conf.int</code>	a confidence interval for the agreement coefficient
<code>estimate</code>	an estimate of the agreement coefficient
<code>alternative</code>	a character string describing the alternative hypothesis
<code>p.value</code>	the p-value for the test
<code>data.name</code>	a character string giving the names of the data

Author(s)

Jennifer Kirk

References

Fay, M.P. (2005). Random marginal agreement coefficients: Rethinking the adjustment for chance in agreement coefficients. *Biostatistics*, 6: 171-180.

See Also

See also: [rmac-package](#), [cac](#), [wkappa](#)

Examples

```
#A simple example of a categorical data set (scores are 1 or 2)
set.seed(41919)
measure1 <- c( rep(1,15), rep(2,5) )
measure2 <- c( rep(1,12), rep(2,8) )
measures<- cbind(measure1, measure2)

#the cost function
nominal <- function(p,q){ test <- abs(p-q)
ifelse(test > 0, 1, 0) }

#the default method
fmacBoot(measures, cost = "nominal")
rmacBoot(measures, cost = "nominal")

#the table method
mtable<- table(measure1, measure2)
fmacBoot(mtable, cost = "nominal")
rmacBoot(mtable, cost = "nominal")

#A simple example of a continuous data set
set.seed(12321)
x<-rnorm(10,1+(1:10))
y<-rnorm(10,(1:10))

plot(x,y,xlim=c(0,12),ylim=c(0,12))
lines(c(0,12),c(0,12))

example<- cbind(x,y)
absolute<- function(p,q){ abs(p-q) }

#Now with the absolute cost function
fmacBoot(example,"absolute")

#this is an alternative method to the cac function
rmacBoot(example,"absolute")
cor.test(x,y,alternative="two.sided")
```

Description

Calculates Cohen's kappa and Scott's pi (also called random marginal agreement coefficient).

Usage

```
wkappa(x, ...)
```

```
## Default S3 method:
wkappa(x, Wcode = 0, method = c("fmac", "rmac"), conf.int = TRUE, conf.level = 0.95, s = 1:k, ...)
## S3 method for class 'table'
wkappa(x, Wcode = 0, method = c("fmac", "rmac"), conf.int = TRUE, conf.level = 0.95, s = 1:k, ...)
```

Arguments

x	table, matrix or data frame of responses, where the responses are in columns 1 and 2 of the matrix or data frame
Wcode	0=categorical weights, 1=absolute value of difference, 2=squared difference
method	indicates which method of calculating the agreement coefficient to use
conf.int	logical, calculate confidence intervals
conf.level	the confidence level
s	vector of scores for values
...	any other arguments passed to the function; not currently used

Details

If the input data is individual scores, it is converted to a contingency table and a list of factors is created based on the combined set of unique levels from each input data set. This method includes factors that are present in one data set and absent in the other.

The default method assumes that the data is in the first two columns of x.

The confidence intervals are calculated using the delta method (See the supplement to Fay (2005) for more details.). If the number of zeros in the contingency matrix is too great, then the confidence intervals cannot be calculated and are set to (-1,1).

Value

Returns a vector of doubles containing the weighted kappa and/or random marginal agreement coefficient statistic and the upper and lower bounds of the confidence interval for each statistic.

Note that if `do.ci = FALSE`, then the confidence interval upper and lower bounds are not returned.

Author(s)

Jennifer Kirk (using functions written by M.P. Fay)

References

- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.*, 20: 37-46.
- Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychol. Bull.*, 70: 213-220.
- Fay, M.P. (2005). Random marginal agreement coefficients: Rethinking the adjustment for chance in agreement coefficients. *Biostatistics*, 6: 171-180.
- Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quart.*, 19: 321-325.

See Also

See also: [rmac-package](#), [rmacBoot](#), [cac](#)

Examples

```
#a simple example with two vectors of measurements (scores are 1 or 2)
set.seed(41919)
measure1 <- c( rep(1,15), rep(2,5) )
measure2 <- c( rep(1,12), rep(2,8) )
measures<- cbind(measure1, measure2)

#the default method
wkappa(measures, method = "fmac")
wkappa(measures, method = "rmac")

#the table method
mtable<- table(measure1, measure2)
wkappa(mtable, method = "fmac")
wkappa(mtable, method = "rmac")
```

Index

*Topic **htest**

cac, [3](#)

rmacBoot, [5](#)

wkappa, [7](#)

*Topic **package**

rmac-package, [2](#)

cac, [2](#), [3](#), [3](#), [6](#), [9](#)

fmacBoot, [2](#), [4](#)

fmacBoot (rmacBoot), [5](#)

rmac (rmac-package), [2](#)

rmac-package, [5](#), [6](#), [9](#)

rmac-package, [2](#)

rmacBoot, [2-4](#), [5](#), [5](#), [9](#)

wkappa, [2](#), [3](#), [5](#), [6](#), [7](#)