Package 'DeltaMAN'

July 21, 2025

Title Delta Measurement of Agreement for Nominal Data

Version 0.5.0

Description Analysis of agreement for nominal data between two raters using the Delta model. This model is proposed as an alternative to the widespread measure Cohen kappa coefficient, which performs poorly when the marginal distributions are very asymmetric (Martin-Andres and Femia-Marzo (2004), <doi:10.1348/000711004849268>; Martin-Andres and Femia-Marzo (2008) <doi:10.1080/03610920701669884>). The package also contains a function to perform a massive analysis of multiple raters against a gold standard. A shiny app is also provided to obtain the measures of nominal agreement between two raters.

License LGPL-3

Depends R (>= 3.5.0)

Imports shiny, shinyMatrix, xtable, shinyBS, knitr

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation no

Author Ana D. Maldonado [aut, cre] (ORCID:
<https://orcid.org/0000-0001-8253-2526>),
Pedro Femia Marzo [aut] (ORCID:
<https://orcid.org/0000-0002-1746-5619>),
Antonio Martín Andrés [aut] (ORCID:
<https://orcid.org/0000-0002-2548-2638>)

Maintainer Ana D. Maldonado <ana.d.maldonado@ual.es>

Repository CRAN

Date/Publication 2022-06-23 18:10:18 UTC

Contents

`delta` *Compute the Delta coefficient*

Description

`delta()` computes Delta coefficient, or proportion of agreements that are not due to chance, which is used to measure nominal agreement between two raters.

`delta()` and `Delta()` are synonyms.

Usage

```
delta(
  data,
  standard = FALSE,
  fixedRows = FALSE,
  rawdata = NULL,
  tol = 1e-07,
  mxits = 100
)

Delta(
  data,
  standard = FALSE,
  fixedRows = FALSE,
  rawdata = NULL,
  tol = 1e-07,
  mxits = 100
)
```

Arguments

`data`	either a contingency table or raw data.
`standard`	a logical value indicating whether the observer on the rows of the contingency table (or first column of raw data) is a goldstandard (i.e., gives the correct responses).
`fixedRows`	a logical value indicating whether the row marginals are fixed in advance (sampling type II) or not (sampling type I).
`rawdata`	a logical value indicating whether the data is raw (TRUE) or a contingency table (FALSE). If not specified, the function will try to guess the data type.
`tol`	the desired tolerance applied to find the root of the unknown constant B, needed to estimate the model parameters.
`mxits`	the maximum numer of iterations applied to find the root of the unknown constant B, needed to estimate the model parameters.

Details

The allowed input data type are (1) contingency tables (of class `"table"` or `"matrix"`) or (2) raw data (of class `"data.frame"`). If the data is of type (1), the empty classes (if any) are removed. If the data is of class (2), the function checks the number of columns (n) and throws an error if n < 2 or n > 3. If n = 2, a frequency table is computed. If n = 3, it is assumed that one column represents the row index (normally, it is expected to be the first one). Once the row index column is identified, it is removed and a frequency table is computed using the remaining two columns. In all cases, the result is always a squared matrix, which will be used in the subsequent computation of the Delta coefficient. The observer on the rows will be referred to as observer (or rater) R and the one on the columns will be referred to as observer (or rater) C.

The function returns a list of 9 elements (if the number of classes is >2):

- Delta: This is a list of 2 elements: (1) the estimates of the model: overall delta (Delta), partial delta for class j (partial_delta) and the distribution of responses made at random by observer C (proportions); and the agreement measurements: agreement, conformity, predictivity and consistency (only some of the are shown, depending on the model assumed) (2) the standard error of the estimates under the model assumed (sampling type I or II).
- Kappa: The estimate and standard error of the Cohen's Kappa coffiecient.
- Data: Input data and analyzed data (may be the same)
- GOF: Goodness of Fit for the Delta model. The chi squares statistic is computed. If the performed test is significative (p-value < alpha), the model Delta is not suitable for the data.
- fixedRows: logical value that matches the "fixedRows" argument.
- standard: logical value that matches the "standard" argument.
- all.measures: This is a list including all the estimates and standard errors (disregarding the model assumed by the user).
- problem.parameters: This list contains information about the estimation of the auxiliary constant B, needed to estimate the model parameters.
- cov: This list contains 3 elements: the covariance matrix of the partial delta estimates; the covariance matrix of the proportion estimates; and the covariance matrix of the partial delta and proportion estimates.

If the number of classes is k = 2, another element is added to the aforementioned list, including the asymptotic analysis (asymptoticDelta).

Value

An object of class `"delta"`, which is a list of 9 elements (or 10 if the dimension of the contingency table is 2x2). See details.

References

Andrés, A. M., & Marzo, P. F. (2004). Delta: A new measure of agreement between two raters. British journal of mathematical and statistical psychology, 57(1), 1-19.

Andrés, A. M., & Marzo, P. F. (2005). Chance-corrected measures of reliability and validity in KK tables. Statistical methods in medical research, 14(5), 473-492.

See Also

summary.delta() for the summary method created for objects of class delta, and print.deltaMAN() for the print method.

Examples

```
# Create a 3x3 matrix
m = matrix(c(15, 5, 0, 4, 21, 1, 3, 4, 25), ncol = 3)
# Compute the Delta coefficient assuming the rater on the rows
# is a goldstandard and type II sampling.
obj = delta(m, standard = TRUE, fixedRows = TRUE)
# Get the complete report
summary(obj, fullReport = TRUE)

# Create a 2x2 matrix
m = matrix(c(15, 7, 3, 21), ncol = 2)
# Compute the Delta coefficient assuming no one is a goldstandard
# and type I sampling.
obj = delta(m, standard = FALSE, fixedRows = FALSE)
# Get the report
summary(obj, fullReport = FALSE)
```

Kappa *Compute the Cohen's kappa coefficient*

Description

Kappa() computes de Cohen's kappa coefficient for nominal or ordinal data. If data is ordinal, weigthed kappa can be applied to allow disagreements to be weighted differently.

Usage

```
Kappa(
  m,
  r = 0,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  partial = FALSE
)
```

Arguments

m	a squared matrix of frequencies between two observers.
r	an integer (0, 1, or 2) to create a matrix of weigths. See details.
alternative	a string specifying the alternative hypothesis to construct the confidence interval: either "two.sided" (default), "greater" or "less".

conf.level	confidence level of the interval.
partial	a logical value indicating whether to evaluate the degree of agreement of each category by collapsing the contingency table.

Details

The weighted kappa can be computed when data are ordinal and the argument r is eigher 1 or 2:

- if r = 0, unweighted kappa is computed (used for nominal variables)
- if r = 1, weighted kappa with linear formula is applied
- if r = 2, weighted kappa with quadratic formula is applied

Value

A list of 3 elements containing the kappa statistic, the standard error and the confidence interval. If "partial = TRUE", a data.frame containing 3 columns (the class, the unweighted partial kappa coefficient for each class and the standards error of each estimate) is added to the list.

Examples

```
# Create a 3x3 matrix
m = matrix(c(15, 5, 0, 4, 21, 1, 3, 4, 25), ncol = 3)
# Compute the Kapa coefficient for nominal data
Kappa(m, r = 0, partial = TRUE)
# Compute the Kapa coefficient for ordinal data, using linear formula
Kappa(m, r = 1)
```

---

multiDelta *Performe massive Delta analysis*

Description

multiDelta() performs the analysis of Delta for multiple raters against a goldstandard.

Usage

```
multiDelta(
  data,
 which.measure = c("Delta", "Agreement", "Conformity", "Predictivity", "Consistency"),
  tol = 1e-07,
  mxits = 100
)
```

Arguments

data	a data.frame whose first column is the goldstandard.
which.measure	character string indicating the measure of interest to retrieve from the analysis. Valid options are: "Delta" (returns global Delta and SE), "Agreement" (returns measure of agreement of each category and SE), "Conformity" (returns measure of conformity of each category and SE), "Predictivity" (returns measure of predictivity of each category and SE), and "Consistency" (returns measure of consistency of each category and SE).
tol	the desired tolerance applied to find the root of the unknown constant B, needed to estimate the model parameters.
mxits	the maximum numer of iterations applied to find the root of the unknown constant B, needed to estimate the model parameters.

Details

A print method is available for "multiDelta" objects. The results can be reported as plain tex (tex = F) or LaTeX formatted (tex = T). In the latter case, the table can be transposed (transpose = T).

Value

An object of class "multiDelta", which is a list of as many elements as measures selected.

Examples

```
# Create a data.frame for 1 goldstandards and 9 raters
dat = data.frame(replicate(10, sample(1:3, 120, replace = TRUE)))

# Compute de Delta model and return the Consistency a Conformity measures
mDelta = multiDelta(dat, which.measure = c('Consistency', 'Conformity'))
print(mDelta, tex = TRUE, transpose = TRUE)
```

---

print.deltaMAN *Print object of class deltaMAN.*

Description

print method for class "deltaMAN".

Usage

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

Arguments

x	an object of class deltaMAN.
...	optional arguments passed to print for other classes created in the deltaMAN package. Currently, it supports "digits" (the significant digits to be used) and "tex" (a logical value indicating whether to generate formatted LaTeX output or plain output). Also, optional argument "transpose" is available for objects of class "multiDelta".

Value

No return value. Results are printed on console.

---

runDeltaShinyApp *Run Delta Shiny App*

Description

Launch the Delta Shiny App to compute the Delta Measurement of Agreement for Nominal, from a user friendly interface. The app also allows to download a report of results in .pdf or .tex (LaTeX) format. The latter format is compressed in a .zip file.

Usage

```
runDeltaShinyApp()
```

Value

No return value. A shiny app is launched.

---

summary.delta *Summary of Delta model*

Description

summary method for class "delta". This functions creates a report with the information contained in an object of class "delta".

Usage

```
## S3 method for class 'delta'
summary(object, fullReport = FALSE, digits = 4, tex = FALSE, ...)
```

Arguments

`object`	and object of class `"delta"`
`fullReport`	a logical value indicating whether to generate an exhaustive report of the results (TRUE) or not (FALSE, default).
`digits`	an integer value indicating the significant digits to be used.
`tex`	a logical value indicating whether to generate formatted LaTeX output (TRUE) or plain output (FALSE, default).
`...`	further arguments passed to summary. No one else is currently available.

Value

No return value. Results are printed on console.

Index