Package 'rococo'

May 9, 2026

Type Package

Title Robust Rank Correlation Coefficient and Test

Version 1.1.10

Date 2025-09-09

Maintainer Ulrich Bodenhofer <ulrich@bodenhofer.com>

Description Provides the robust gamma rank correlation coefficient as introduced by Bodenhofer, Krone, and Klawonn (2013) <DOI:10.1016/j.ins.2012.11.026> along with a permutation-based rank correlation test. The rank correlation coefficient and the test are explicitly designed for dealing with noisy numerical data.

License GPL (>= 2)

LazyLoad yes

Depends R (>= 3.0.0)

Imports Rcpp (>= 0.11.1), methods, stats

Suggests compiler, datasets, knitr

Collate AllGenerics.R AllClasses.R show-methods.R rococo.R rococo.test-methods.R gauss.cor.R gauss.cor.test-methods.R

URL https://github.com/UBod/rococo

VignetteBuilder knitr

LinkingTo Rcpp

Repository CRAN

NeedsCompilation yes

Author Martin Krone [aut], Ulrich Bodenhofer [aut, cre]

Date/Publication 2025-09-22 17:48:47 UTC

Contents

`gauss.cor` *Gauss Rank Correlation Estimator*

Description

Compute the Gaussian rank correlation estimate

Usage

```
gauss.cor(x, y)
```

Arguments

`x`	a numeric vector; compulsory argument
`y`	a numeric vector; compulsory argument; `x` and `y` need to have the same length

Details

`gauss.cor` computes the Gaussian rank correlation estimate for `x` and `y`.

Note that `gauss.cor` only works for `x` and `y` being numeric vectors, unlike the classical correlation measures implemented in `cor` which can also be computed for matrices or data frames.

Value

Upon successful completion, the function returns the Gaussian rank correlation estimate.

Author(s)

Ulrich Bodenhofer

References

https://github.com/UBod/rococo

K. Boudt, J. Cornelissen, and C. Croux (2012). The Gaussian rank correlation estimator: robustness properties. *Stat. Comput.* **22**(2):471-483. DOI: doi:10.1007/s1122201192370.

See Also

`gauss.cor.test`

Examples

```
## create data
f <- function(x) ifelse(x > 0.9, x - 0.9, ifelse(x < -0.9, x + 0.9, 0))
x <- rnorm(25)
y <- f(x) + rnorm(25, sd=0.1)

## compute correlation
gauss.cor(x, y)
```

`gauss.cor.test-methods`

Gaussian rank correlation test

Description

Methods performing a Gaussian rank correlation test

Usage

```
## S4 method for signature 'numeric,numeric'
gauss.cor.test(x, y, ...)
## S4 method for signature 'formula,data.frame'
gauss.cor.test(x, y, na.action, ...)
```

Arguments

`x`	a numeric vector or a formula; compulsory argument
`y`	compulsory argument; if `x` is a vector, `y` must be vector of the same length as `x`. If `x` is a formula, `y` must be a data frame.
`na.action`	a function which indicates what should happen when the data contain `NA`'s. Defaults to `getOption("na.action")`.
`...`	all parameters specified are forwarded internally to the method `cor.test`, in particular, the `alternative` parameter.

Details

If called for numeric vectors, `gauss.cor.test` performs the Gaussian gamma rank correlation test for `x` and `y`. This is done by simply performing a Pearson correlation test on the normal scores of the data.

If `gauss.cor.test` is called for a formula `x` and a data frame `y`, then the method checks whether the formula `x` correctly extracts two columns from `y` (see examples below). If so, the two columns are extracted and the Gaussian gamma rank correlation test is applied to them according to the specified parameters.

Value

Upon successful completion, the function returns a list of class `htest` containing the results (see `cor.test`).

Author(s)

Ulrich Bodenhofer

References

https://github.com/UBod/rococo

K. Boudt, J. Cornelissen, and C. Croux (2012). The Gaussian rank correlation estimator: robustness properties. *Stat. Comput.* **22**(2):471-483. DOI: doi:10.1007/s1122201192370.

See Also

gauss.cor

Examples

```
## create data
f <- function(x) ifelse(x > 0.9, x - 0.9, ifelse(x < -0.9, x + 0.9, 0))
x <- rnorm(25)
y <- f(x) + rnorm(25, sd=0.1)

## perform correlation tests
gauss.cor.test(x, y, alternative="greater")

## the formula variant
require(datasets)
data(iris)
gauss.cor.test(~ Petal.Width + Petal.Length, iris,
               alternative="two.sided")
```

---

rococo *Robust Gamma Rank Correlation Coefficient*

Description

Compute the robust gamma rank correlation coefficient

Usage

```
rococo(x, y,
       similarity=c("linear", "exp", "gauss", "epstol", "classical"),
       tnorm="min", r=0, noVarReturnZero=TRUE)
```

Arguments

x	a numeric vector; compulsory argument
y	a numeric vector; compulsory argument; x and y need to have the same length

`similarity` a character string or a character vector identifying which type of similarity measure to use; valid values are `"linear"` (default), `"exp"`, `"gauss"`, `"epstol"`, and `"classical"` (abbreviations are allowed as long as they are unique). If `similarity` is a single string, the same similarity measure is taken for `x` and `y`. Different similarity measures can be used for `x` and `y` by supplying different similarity measures in `similarity[1]` and `similarity[2]`. Longer character vectors are allowed, but all but the first two entries are ignored.

`tnorm` can be any of the following strings identifying a standard tnorm: `"min"` (minimum t-norm; default), `"prod"` (product t-norm), or `lukasiewicz` (Lukasiewicz t-norm); abbreviations are allowed as long as they are unique. Alternatively, `tnorm` can be a two-argument function defining a t-norm.

`r` numeric vector defining the tolerances to be used; if a single value is supplied, the same value is used both for `x` and `y`. If a vector is supplied, `r[1]` is used as tolerance for `x` and `r[2]` is used as tolerance for `y`. If the classical crisp similarity is used, the corresponding entry/entries in `r` is/are ignored. Negative values are not allowed. Zeroes have a special meaning: if an entry in `r` is `0`, then the tolerance is automatically adapted to 10 percent of the interquartile range of the data.

`noVarReturnZero` if `TRUE` (default), a correlation of 0 is returned if there is no variation in at least one of the two observables. Otherwise, `NA` is returned and a warning is issued.

Details

`rococo` computes the robust gamma rank correlation coefficient of `x` and `y` according to the specified parameters (see literature for more details).

Note that `rococo` only works for `x` and `y` being numeric vectors, unlike the classical correlation measures implemented in `cor` which can also be computed for matrices or data frames.

Value

Upon successful completion, the function returns the robust gamma rank correlation coefficient.

Author(s)

Martin Krone and Ulrich Bodenhofer

References

https://github.com/UBod/rococo

U. Bodenhofer and F. Klawonn (2008). Robust rank correlation coefficients on the basis of fuzzy orderings: initial steps. *Mathware Soft Comput.* **15**(1):5-20.

U. Bodenhofer, M. Krone, and F. Klawonn (2013). Testing noisy numerical data for monotonic association. *Inform. Sci.* **245**:21-37. DOI: doi:10.1016/j.ins.2012.11.026.

See Also

`rococo.test`

Examples

```
## create data
f <- function(x) ifelse(x > 0.9, x - 0.9, ifelse(x < -0.9, x + 0.9, 0))
x <- rnorm(25)
y <- f(x) + rnorm(25, sd=0.1)

## compute correlation
rococo(x, y, similarity="classical")
rococo(x, y, similarity="linear")
rococo(x, y, similarity=c("classical", "gauss"), r=c(0, 0.1))
```

---

rococo.test-methods *Robust Gamma Rank Correlation Test*

Description

Methods performing a robust gamma rank correlation test

Usage

```
## S4 method for signature 'numeric,numeric'
rococo.test(x, y,
            similarity=c("linear", "exp", "gauss",
                         "epstol", "classical"),
            tnorm="min", r=0, numtests=1000, storeValues=FALSE,
            exact=FALSE, alternative=c("two.sided", "less", "greater"),
            noVarReturnZero=TRUE)
## S4 method for signature 'formula,data.frame'
rococo.test(x, y, na.action, ...)
```

Arguments

x	a numeric vector or a formula; compulsory argument
y	compulsory argument; if x is a vector, y must be vector of the same length as x. If x is a formula, y must be a data frame.
similarity	a character string or a character vector identifying which type of similarity measure to use; see rococo for more details.
tnorm	t-norm used for aggregating results; see rococo for more details.
r	numeric vector defining the tolerances to be used; see rococo for more details.
numtests	number of random shuffles to perform; see details below.
storeValues	logical indicating whether the vector of test statistics should be stored in the output object (in slot perm.gamma, see RococoTestResults).
exact	logical indicating whether exact p-value should be computed; see details below.
alternative	indicates the alternative hypothesis and must be one of "two.sided", "greater", or "less". Abbreviations are allowed as long as they are unique. "greater" corresponds to positive association, "less" to negative association.

noVarReturnZero
: if TRUE (default), a correlation of 0 and a p-value of 1 are returned if there is no variation in at least one of the two observables. Otherwise, a correlation of NA and a p-value of 1 are returned and a warning is issued.

na.action
: a function which indicates what should happen when the data contain NA's. Defaults to getOption("na.action").

...
: all parameters specified are forwarded internally to the method rococo.test with signature numeric,numeric.

Details

If called for numeric vectors, rococo.test computes the robust gamma rank correlation coefficient of x and y according to the specified parameters (see rococo) and then performs a permutation test to compute a p-value. If exact=TRUE, rococo.test attempts to compute an exact p-value and ignores the numtests argument. This is done by considering all possible permutations and computing the ratio of permutations for which the test statistic is at least as large/small as the test statistic for unshuffled data. This works only for 10 or less samples. Otherwise exact=TRUE is ignored, a warning is issued and random shuffles are considered to estimate the p-value (as follows next). If exact=FALSE, numtests random shuffles of y are performed and the empirical standard deviation of the robust gamma correlation values for these shuffled data sets is computed. Under the assumption that these values are normally distributed around mean zero, the p-value is then computed from this distribution in the usual way. Note that a too small choice of the number of shuffles (parameter numtests) leads to unreliable p-values.

If rococo.test is called for a formula x and a data frame y, then the method checks whether the formula x correctly extracts two columns from y (see examples below). If so, the two columns are extracted and the robust gamma rank correlation test is applied to them according to the specified parameters.

Note that exact=TRUE may result in long computation times for user-defined t-norms.

Value

Upon successful completion, the function returns an object of class RococoTestResults containing the results.

Author(s)

Martin Krone and Ulrich Bodenhofer

References

https://github.com/UBod/rococo

U. Bodenhofer, M. Krone, and F. Klawonn (2013). Testing noisy numerical data for monotonic association. *Inform. Sci.* **245**:21-37. DOI: doi:10.1016/j.ins.2012.11.026.

U. Bodenhofer and F. Klawonn (2008). Robust rank correlation coefficients on the basis of fuzzy orderings: initial steps. *Mathware Soft Comput.* **15**(1):5-20.

See Also

rococo

Examples

```
## create data
f <- function(x) ifelse(x > 0.9, x - 0.9, ifelse(x < -0.9, x + 0.9, 0))
x <- rnorm(25)
y <- f(x) + rnorm(25, sd=0.1)

## perform correlation tests
rococo.test(x, y, similarity="classical", alternative="greater")
rococo.test(x, y, similarity="linear", alternative="greater")
rococo.test(x, y, similarity=c("classical", "gauss"), r=c(0, 0.1),
            alternative="greater", numtests=10000)

## the formula variant
require(datasets)
data(iris)
rococo.test(~ Petal.Width + Petal.Length, iris, similarity="linear",
            alternative="two.sided")
```

RococoTestResults-class

Class "RococoTestResults"

Description

S4 class for storing results of the robust rank correlation test

Objects

Objects of this class can be created by calling `rococo.test`.

Slots

The following slots are defined for `RococoTestResults` objects:

`count`: number of times in which the test statistic for a random shuffle exceeded the test statistic of the true data; see `rococo.test`.

`tnorm`: list identifying t-norm to use or two-argument function; see `rococo`. If one of the standard choices `"min"`, `"prod"`, or `"lukasiewicz"` has been used, the list has one component, `name` that contains the string identifying the t-norm. If a user-defined function has been used, the list has two components: `name` contains `"user-defined t-norm"` or the `name` attribute of the function object if available and `def` contains the function object itself.

`input`: character string describing the input for which `rococo.test` has been called.

`length`: number of samples for which `rococo.test` has been called.

`p.value`: p-value of test.

`p.value.approx`: p-value as based on a normal approximation of the null distribution.

`r.values`: vector containing tolerance levels for the two inputs; see `rococo.test` or `rococo`.

`numtests`: number of (random) shuffles performed by `rococo.test`.

`exact`: logical indicating whether p-value has been computed exactly; see `rococo.test`.

`similarity`: character (vector) identifying the similarity measure(s) used by `rococo.test`.

`sample.gamma`: test statistic (robust gamma rank correlation coefficient) determined by `rococo.test`.

`H0gamma.mu`: empirical mean of test statistic for random shuffles

`H0gamma.sd`: empirical standard deviation of test statistic for random shuffles

`perm.gamma`: in case `rococo.test` was called with `storeValues=TRUE`, this slot contains the vector of test statistics for random shuffles.

`alternative`: alternative hypothesis used by `rococo.test`.

Methods

show `signature(object = "RococoTestResults")`: d displays the most important information stored in `object`

Author(s)

Martin Krone and Ulrich Bodenhofer

References

https://github.com/UBod/rococo

U. Bodenhofer, M. Krone, and F. Klawonn (2013). Testing noisy numerical data for monotonic association. *Inform. Sci.* **245**:21-37. DOI: doi:10.1016/j.ins.2012.11.026.

U. Bodenhofer and F. Klawonn (2008). Robust rank correlation coefficients on the basis of fuzzy orderings: initial steps. *Mathware Soft Comput.* **15**(1):5-20.

See Also

`rococo.test`, `rococo`, `show-methods`

Examples

```
## create data
f <- function(x) ifelse(x > 0.9, x - 0.9, ifelse(x < -0.9, x + 0.9, 0))
x <- rnorm(25)
y <- f(x) + rnorm(25, sd=0.1)

## perform correlation tests
ret <- rococo.test(x, y, similarity="classical", alternative="greater")
show(ret)

ret <- rococo.test(x, y, similarity="linear", alternative="greater")
show(ret)

ret <- rococo.test(x, y, similarity=c("classical", "gauss"),
                   r=c(0, 0.1), alternative="greater",
                   numtests=10000)
show(ret)
```

Index