Package ‘nopaco’

October 13, 2022

Type Package

Title Non-Parametric Concordance Coefficient

Version 1.0.6

Date 2019-07-04

Description A non-parametric test for multi-observer concordance and differences between concordances in (un)balanced data.

License GPL (>= 3)

Imports methods, Matrix (>= 1.1.5), parallel, stats

Suggests MASS

SystemRequirements C++11

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation yes

Author Rowan Kuiper [cre, aut] (<`https://orcid.org/0000-0002-3703-1762`>),
Remco Hoogenboezem [aut],
Sjoerd Huisman [ctb] (<`https://orcid.org/0000-0002-4322-8289`>),
Pieter Sonneveld [ths],
Mark van Duin [ths]

Maintainer Rowan Kuiper <`r.kuiper.emc@gmail.com`>

Repository CRAN

Date/Publication 2019-07-06 22:01:19 UTC

R topics documented:

`coef` *Extract test results from the results of a concordance.test*

Description

`coef` extract the test results from the results of a concordance.test

Usage

```
## S4 method for signature 'ConcordanceTest'
coef(object, ...)
```

Arguments

`object`	An object of `ConcordanceTest-class`
`...`	Not used

Value

A matrix

See Also

Other concordance functions: `concordance.test`, `getPsi`, `rfromPsi`

Examples

```
matRandom <- matrix(rnorm(3*20),20,3)
testResult <- concordance.test(matRandom)
getPsi(testResult)
coef(testResult)
```

`concordance.test` *Perform a nonparametric concordance test.*

Description

`concordance.test` performs a test for a random concordance (if a single matrix is given) or tests for equal concordance between two matrices.

Usage

```
concordance.test(x, y = NULL, alternative = NULL, alpha = 0.05, ...)
```

Arguments

`x`	a numeric matrix, subjects in the rows, repeated measurements in the columns
`y`	(optional) a numeric matrix of equal size as argument `x`
`alternative`	"less", "greater" or "two.sided". Only used when `y` is given.
`alpha`	significance level (default = 0.05)
`...`	see details

Details

- Testing the deviation from random concordance: if only one matrix is given (i.e. argument `x`), its concordance will be tested against alternative hypothesis of finding a higher concordance under random sampling conditions. For small matrices (depending on number of replicate measurements) an exact method will be used to determine to p-value. In case of larger matrices where the exact approach in not feasible, either the revised-beta approach (default), a beta approximation or a normal approximation is used. To enforce the use of either one method, the `method` argument can be used with value "exact","Rbeta","beta" or "normal".
- Testing for a difference between concordances: if both arguments `x` and `y` have been given, the equality of concordances of both matrices is tested. The default alternative hypothesis is 'two.sided'. Both matrices must be of equal size and have corresponding missing entries (`NA` values). In case of missing data in one matrix, the same entries in the other matrix will also be set to missing.

Unbalanced data due to randomly missing data or an unequal number of repeated measurements per subject is allowed. In that case, missing or unknown values must be set to `NA`.

Value

An object of `ConcordanceTest-class`

References

P.Rothery (1979) Biometrika 66(3):629-639

See Also

Other concordance functions: `coef`, `getPsi`, `rfromPsi`

Examples

```
require(MASS) ##to use mvrnorm function

#Generate a matrix without concordance
matRandom <- matrix(rnorm(3*20),20,3)
concordance.test(matRandom)

#Generate a matrix with strong concordance
sigma<-matrix(0.8,3,3)
diag(sigma)<-1
matConcordant <- mvrnorm(20,mu=rep(0,3),Sigma=sigma)
```

```
concordance.test(matConcordant)

#Test concordances between matrices
aTest <- concordance.test(matConcordant, matRandom)

getPsi(aTest)
coef(aTest)
```

`ConcordanceTest-class` *Class ConcordanceTest*

Description

This class stores results obtained from a concordance test.

Details

Class `ConcordanceTest` stores results from a concordance test.

Slots

`pvalue`	The pvalue
`psi1`	The concordance in matrix `x`
`psi2`	The concordance in matrix `y`
`method`	The method used to obtain the pvalue
`alternative`	The alternative hypothesis
`ci.lower`	The lower confidence boudary
`ci.upper`	The upper confidence boudary
`ci.method`	The method used to obtain the confidence interval
`alpha`	The significance level
`call`	The call made to the `concordance.test` function

`getPsi` *Obtain concordance coefficients.*

Description

`getPsi` returns the concordance coefficient(s) from a matrix or a result obtained by the `concordance.test` function.

Usage

```
getPsi(x, y, ...)

## S4 method for signature 'ConcordanceTest,missing'
getPsi(x)

## S4 method for signature 'matrix,missing'
getPsi(x, y, ...)

## S4 method for signature 'data.frame,missing'
getPsi(x, y, ...)

## S4 method for signature 'data.frame,data.frame'
getPsi(x, y, ...)

## S4 method for signature 'matrix,`NULL`'
getPsi(x, y, ...)

## S4 method for signature 'matrix,matrix'
getPsi(x, y, ...)
```

Arguments

`x`	A numeric matrix or an object `ConcordanceTest-class`
`y`	A numeric matrix (optional)
`...`	Not used

Value

A numeric vector with coefficient(s)

References

P.Rothery (1979) Biometrika 66(3):629-639

See Also

Other concordance functions: `coef`, `concordance.test`, `rfromPsi`

Examples

```
matRandom <- matrix(rnorm(30),10,3)
testResult <- concordance.test(matRandom)
getPsi(testResult)
getPsi(matRandom)
```

`names,ConcordanceTest-method` *Extract argument names from a ConcordanceTest object*

Description

`names` extracts argument names from a `ConcordanceTest-class` object

Usage

```
## S4 method for signature 'ConcordanceTest'
names(x)
```

Arguments

`x`	An object of `ConcordanceTest-class`

Value

A character vector

Examples

```
matRandom <- matrix(rnorm(3*20),20,3)
testResult <- concordance.test(matRandom)
names(testResult)
```

`rfromPsi` *Convertion between Pearson correlation and the non paramtric concordance coefficient*

Description

Convertion between Pearson correlation and the non paramtric concordance coefficient

Usage

```
rfromPsi(psi)

psifromR(r)
```

Arguments

`psi`	a (vector of) non paramtric concordance coefficient(s)
`r`	a (vector of) Pearson correlation coefficient(s)

Details

The convertion is performed following the relationship described by Rothery (1979). `2*cos(pi*(1-psi))-1`

Value

A (vector of) corresponding Pearson correlation coefficient(s).

References

Rothery, P. 'A nonparametric measure of intraclass correlation', Biometrika, 66, 3, 629-639 (1979).

See Also

Other concordance functions: `coef`, `concordance.test`, `getPsi`

Examples

```
#Generate a matrix without concordance
matRandom <- matrix(rnorm(30),10,3)
result<-concordance.test(matRandom)
getPsi(result) #concordance coefficient
result$ci      #95% confidence interval

#Corresonding Pearson correlation
rfromPsi(getPsi(result))
rfromPsi(result$ci)

#Plot the relation between Pearson correlation and the nonparamatric concordance coefficient.
r<-seq(-1,1,0.01)
psi<-psifromR(r)
plot(r,psi,type='l',xlab="Pearson correlation", ylab="nonparametric concordance")
```

---

scores *Hypothetical data of outcomes for two risk models*

Description

This data is generated as explained in the nopaco vignette. It represents the outcomes of the risk models (model A and model B). Both models were applied to gene expression profiles 100 subjects, each run in duplo.

Usage

```
data(scores)
```

Format

A list with two elements named 'modelA' and 'modelB' both containing a dataframe with outcome scores for 100 subjects in the rows each having two replicate measurements in the columns.

Source

`vignette("nopaco", package = "nopaco")`

Examples

```
data(scores)
str(scores)
plot(scores[['modelA']])
plot(scores[['modelB']])
```

`$,ConcordanceTest-method`

Extract argument values from a ConcordanceTest object

Description

Extracts argument values from a `ConcordanceTest-class` object

Usage

```
## S4 method for signature 'ConcordanceTest'
x$name
```

Arguments

`x`	An object of `ConcordanceTest-class`
`name`	The argument to get the value of

Value

The value of the requested argument

Examples

```
matRandom <- matrix(rnorm(3*20),20,3)
testResult <- concordance.test(matRandom)
names(testResult)
testResult$psi
```

Index