

Package ‘libcoin’

August 14, 2020

Title Linear Test Statistics for Permutation Inference

Date 2020-08-13

Version 1.0-6

Description Basic infrastructure for linear test statistics and permutation inference in the framework of Strasser and Weber (1999) <<https://epub.wu.ac.at/102/>>. This package must not be used by end-users. CRAN package 'coin' implements all user interfaces and is ready to be used by anyone.

Depends R (>= 3.4.0)

Suggests coin

Imports stats, mvtnorm

LinkingTo mvtnorm

NeedsCompilation yes

License GPL-2

Author Torsten Hothorn [aut, cre]

Maintainer Torsten Hothorn <Torsten.Hothorn@R-project.org>

Repository CRAN

Date/Publication 2020-08-14 13:59:24 UTC

R topics documented:

ctabs	2
doTest	2
LinStatExpCov	3

Index	6
--------------	----------

ctabs	<i>Cross Tabulation</i>
-------	-------------------------

Description

Efficient weighted cross tabulation of two factors and a block

Usage

```
ctabs(ix, iy = integer(0), block = integer(0), weights = integer(0),
      subset = integer(0), checkNAs = TRUE)
```

Arguments

ix	a integer of positive values with zero indicating a missing.
iy	an optional integer of positive values with zero indicating a missing.
block	an optional blocking factor without missings.
weights	an optional vector of weights, integer or double.
subset	an optional integer vector indicating a subset.
checkNAs	a logical for switching off missing value checks.

Details

A faster version of `xtabs(weights ~ ix + iy + block, subset)`.

Value

If block is present, a three-way table. Otherwise, a one- or two-dimensional table.

Examples

```
ctabs(ix = 1:5, iy = 1:5, weights = 1:5 / 5)
```

doTest	<i>Permutation Test</i>
--------	-------------------------

Description

Perform permutation test for a linear statistic

Usage

```
doTest(object, teststat = c("maximum", "quadratic", "scalar"),
       alternative = c("two.sided", "less", "greater"), pvalue = TRUE,
       lower = FALSE, log = FALSE, PermutedStatistics = FALSE,
       minbucket = 10L, ordered = TRUE, maxselect = object$Xfactor,
       pargs = GenzBretz())
```

Arguments

object	an object returned by LinStatExpCov .
teststat	type of test statistic to use.
alternative	alternative for scalar or maximum-type statistics.
pvalue	a logical indicating if a p-value shall be computed.
lower	a logical indicating if a p-value (lower is FALSE) or 1 - p-value (lower is TRUE) shall be returned.
log	a logical, if TRUE probabilities are log-probabilities.
PermutedStatistics	a logical, return permuted test statistics.
minbucket	minimum weight in either of two groups for maximally selected statistics.
ordered	a logical, if TRUE maximally selected statistics assume that the cutpoints are ordered.
maxselect	a logical, if TRUE maximally selected statistics are computed. This requires that X was an implicitly defined design matrix in LinStatExpCov .
pargs	arguments as in GenzBretz .

Details

Computes a test statistic, a corresponding p-value and, optionally, cutpoints for maximally selected statistics.

Value

A list.

 LinStatExpCov

Linear Statistics with Expectation and Covariance

Description

Strasser-Weber type linear statistics and their expectation and covariance under the independence hypothesis

Usage

```
LinStatExpCov(X, Y, ix = NULL, iy = NULL, weights = integer(0),
              subset = integer(0), block = integer(0), checkNAs = TRUE,
              varonly = FALSE, nresample = 0, standardise = FALSE,
              tol = sqrt(.Machine$double.eps))
lmult(x, object)
```

Arguments

<code>X</code>	numeric matrix of transformations.
<code>Y</code>	numeric matrix of influence functions.
<code>ix</code>	an optional integer vector expanding <code>X</code> .
<code>iy</code>	an optional integer vector expanding <code>Y</code> .
<code>weights</code>	an optional integer vector of non-negative case weights.
<code>subset</code>	an optional integer vector defining a subset of observations.
<code>block</code>	an optional factor defining independent blocks of observations.
<code>checkNAs</code>	a logical for switching off missing value checks. This included switching off checks for suitable values of <code>subset</code> . Use at your own risk.
<code>varonly</code>	a logical asking for variances only.
<code>nresample</code>	an integer defining the number of permuted statistics to draw.
<code>standardise</code>	a logical asking to standardise the permuted statistics.
<code>tol</code>	tolerance for zero variances.
<code>x</code>	a contrast matrix to be left-multiplied in case <code>X</code> was a factor.
<code>object</code>	an object of class <code>LinStatExpCov</code> .

Details

The function, after minimal preprocessing, calls the underlying C code and computes the linear statistic, its expectation and covariance and, optionally, `nresample` samples from its permutation distribution.

When both `ix` and `iy` are missing, the number of rows of `X` and `Y` is the same, ie the number of observations.

When `X` is missing and `ix` a factor, the code proceeds as if `X` were a dummy matrix of `ix` without explicitly computing this matrix.

Both `ix` and `iy` being present means the code treats them as subsetting vectors for `X` and `Y`. Note that `ix = 0` or `iy = 0` means that the corresponding observation is missing and the first row of `X` and `Y` must be zero.

`mult` allows left-multiplication of a contrast matrix when `X` was (equivalent to) a factor.

Value

A list.

References

Strasser, H. and Weber, C. (1999). On the asymptotic theory of permutation statistics. *Mathematical Methods of Statistics* **8**(2), 220–250.

Examples

```
wilcox.test(Ozone ~ Month, data = airquality, subset = Month %in% c(5, 8))
```

```
aq <- subset(airquality, Month %in% c(5, 8))
```

```
X <- as.double(aq$Month == 5)
```

```
Y <- as.double(rank(aq$Ozone))
```

```
doTest(LinStatExpCov(X, Y))
```

Index

* **htest**

doTest, 2

LinStatExpCov, 3

* **univar**

ctabs, 2

ctabs, 2

doTest, 2

GenzBretz, 3

LinStatExpCov, 3, 3

lmult (LinStatExpCov), 3