Package: libcoin (via r-universe)

October 28, 2026

Title Linear Test Statistics for Permutation Inference

Date 2026-06-03

Version 1.0-13

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, bibtex

Imports stats, mvtnorm

LinkingTo mvtnorm

NeedsCompilation yes

License GPL-2

URL https://codeberg.org/thothorn/coin/

Author Torsten Hothorn [aut, cre] (ORCID: <https://orcid.org/0000-0001-8301-0471>), Henric Winell [aut] (ORCID: <https://orcid.org/0000-0001-7995-3047>)

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

Repository https://cran.r-universe.dev

Date/Publication 2026-06-04 10:46:03 UTC

RemoteUrl https://github.com/cran/libcoin

RemoteRef HEAD

RemoteSha e938c6d6c7d2e40866361895f4d02a57e8ebe5c2

Contents

ctabs 2
doTest 2
LinStatExpCov 3

Index **6**

ctabs *Cross Tabulation*

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 case 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 *Permutation Test*

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

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

Details

This function implements the permutation test framework by Strasser and Weber (1999), see also Hothorn, Hornik, van de Wiel, and Zeileis (2006) and Hothorn, Hornik, van de Wiel, and Zeileis (2008).

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, i.e., 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 or X and Y must be zero.

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

Value

A list.

References

Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2006). "A Lego System for Conditional Inference." *The American Statistician*, **60**(3), 257–263. doi:10.1198/000313006X118430.

Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2008). "Implementing a Class of Permutation Tests: The coin Package." *Journal of Statistical Software*, **28**(8), 1–23. doi:10.18637/jss.v028.i08.

Strasser H, Weber C (1999). "The Asymptotic Theory of Permutation Statistics." *Mathematical Methods of Statistics*, **8**(2), 220–250. Preprint available from https://epub.wu.ac.at/102.

Examples

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

aq <- subset(airquality, Month %in% c(5, 8))
X <- as.double(aq$Month == 5)
Y <- as.double(rank(aq$Ozone, na.last = "keep"))
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