Package 'perm'

May 9, 2026

Type Package

Title Exact or Asymptotic Permutation Tests

Version 1.0-0.4

Date 2023-08-24

Author Michael Fay

Maintainer Michael P. Fay <mfay@niaid.nih.gov>

Depends R (>= 2.2.1), stats

Suggests coin

Description
Perform Exact or Asymptotic permutation tests [see Fay and Shaw <doi:10.18637/jss.v036.i02>].

License GPL

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-24 21:00:02 UTC

Contents

perm-package *Exact or Asymptotic linear permutation tests*

Description

This package gives several methods for performing permutation tests.

Details

The package has three main functions, to perform linear permutation tests. These tests are tests where the test statistic is the sum of the product of a covariate (usually group indicator) and the scores. The three tests are: `permTS` to perform two sample permutation tests, `permKS` to perform K-sample permutation tests, `permTREND` to perform trend permutation tests on numeric values. By using suitable scores one can create for example, the permutation t-test (general scores), the Wilcoxon rank sum test (rank scores), the logrank test (need to use other functions to create these scores). The two sample test uses either exact (network algorithm, complete enumeration, or Monte Carlo) or asymptotic calculations (using permutational central limit theorem [pclt]), while the other tests use only the exact Monte Carlo or the pclt. Most (if not all) of the tests here are also implemented in the `coin` package. This package provides an independent validation of that package.

The `perm` package used by the `interval` package, and `perm` is described in Fay and Shaw (2010, Section 5).

Author(s)

Michael Fay

Maintainer: Michael Fay <mfay@niaid.nih.gov>

References

Fay, MP and Shaw, PA (2010). Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R package. Journal of Statistical Software. doi:10.18637/jss.v036.i02. 36 (2):1-34.

chooseMatrix *Create matrix with choose(n,m) unique rows with m ones in each row the rest 0.*

Description

Create a `choose(n,m)` by n matrix. The matrix has unique rows with m ones in each row and the rest zeros.

Usage

```
chooseMatrix(n, m)
```

Arguments

n	an integer
m	an integer<=n

Value

A matrix with choose(n,m) rows n columns. The matrix has unique rows with m ones in each row and the rest zeros.

Note

Used for complete enumeration when method='exact.ce' in permTS

Author(s)

M.P.Fay

See Also

permTS

Examples

```
chooseMatrix(5,2)
```

---

methodRuleKS1 *Rule for determining method for permKS*

Description

This is the default function which determines which method to use in permKS.

Usage

```
methodRuleKS1(x, group, exact, Nbound = c(5))
```

Arguments

x	vector of response scores
group	group membership vector
exact	logical, TRUE=exact method chosen, FALSE=pclt
Nbound	gives 'pclt' if minimum sample size of any group > Nbound

Details

This function determines which of two methods will be used in permKS; see that help for description of methods.

When exact=FALSE then returns 'pclt'. When exact=TRUE then returns 'exact.mc'. When exact=NULL then returns either 'exact.mc' if the minimum sample size for any group is less than or equal to Nbound, otherwise returns 'pclt'.

Value

a character vector with one of the following values: "pclt","exact.mc"

See Also

permKS

`methodRuleTREND1` *Rule for determining method for permTREND*

Description

This is the default function which determines which method to use in permTREND.

Usage

```
methodRuleTREND1(x, y, exact, Nbound = c(20))
```

Arguments

x	vector of response scores
y	group membership vector
exact	logical, TRUE=exact method chosen, FALSE=pclt
Nbound	gives 'pclt' if length(x) > Nbound

Details

This function determines which of two methods will be used in permTREND; see that help for description of methods.

When exact=FALSE then returns 'pclt'. When exact=TRUE then returns 'exact.mc'. When exact=NULL then returns either 'exact.mc' if length(x) is less than or equal to Nbound, otherwise returns 'pclt'.

Value

a character vector with one of the following values: "pclt","exact.mc"

See Also

permKS

`methodRuleTS1` *Rule for determining method for permTS*

Description

This is the default function which determines which method to use in `permTS`.

Usage

```
methodRuleTS1(x, group, exact, Nbound = c(1000, 200, 100, 50, 16))
```

Arguments

`x`	vector of response scores
`group`	group membership vector
`exact`	logical, TRUE=exact method chosen, FALSE=pclt
`Nbound`	vector of bounds (see details)

Details

This function determines which of several methods will be used in `permTS`; see that help for description of methods.

When exact=FALSE then returns 'pclt'. When exact=TRUE then returns either 'exact.network' if the estimated time of calculation is not too large or 'exact.mc' otherwise. When exact=NULL then returns either 'exact.network' if the estimated time is not too large or 'pclt' otherwise. The estimation of the calculation time is as follows: if the smallest number of unique values in one of the two groups is equal to kmin, then calculation time is large if the sample size <=Nbound[kmin-1], if Nbound[kmin-1] exists, or is large if the sample size <= min(Nbound) otherwise.

Value

a character vector with one of the following values: "pclt","exact.network","exact.mc"

See Also

`permTS`

Examples

```
N<-100
set.seed(1)
methodRuleTS1(x=sample(1:2,N,replace=TRUE),group=sample(c(0,1),N,replace=TRUE),exact=NULL)
N<-100
methodRuleTS1(sample(1:500,N,replace=TRUE),sample(c(0,1),N,replace=TRUE),TRUE)
```

perm *Exact or Asymptotic 2-sample, k-sample, and trend permutation tests*

Description

These functions perform either: two-sample permutation tests (permTS), k-sample permutation tests (permKS), or trend permutation tests (permTREND). The test function can be transformed to a linear function of the scores times the covariate, where the covariate may be either a factor or character vector with two (permTS) or more (permKS) levels or a numeric vector (permTREND). By using suitable scores one can create for example, the permutation t-test (general scores), the Wilcoxon rank sum test (rank scores), the logrank test (need to use other functions to create these scores). It performs either exact (network algorithm, complete enumeration, or Monte Carlo) asymptotic calculations (using permutational central limit theorem).

Usage

```
permTS(x, ...)

## Default S3 method:
permTS(x, y, alternative = c("two.sided", "less", "greater"),
    exact = NULL, method = NULL, methodRule = methodRuleTS1,
    control=permControl(), ...)

## S3 method for class 'formula'
permTS(formula, data, subset, na.action, ...)


permKS(x,...)


## Default S3 method:
permKS(x, g, exact = NULL, method = NULL,
   methodRule = methodRuleKS1, control=permControl(), ...)

## S3 method for class 'formula'
permKS(formula,data,subset, na.action,...)

permTREND(x,...)

## Default S3 method:
permTREND(x, y, alternative = c("two.sided", "less", "greater"),
  exact = NULL, method = NULL, methodRule = methodRuleTREND1, control=permControl(),...)

## S3 method for class 'formula'
permTREND(formula,data,subset,na.action,...)
```

Arguments

x	numeric vector of respose scores for the first group
y	numeric vector of either response scores for the second group (for permTS) or trend scores for each observation (for permTREND)
g	a factor or character vector denoting group membership
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater","less" (see details)
exact	a logical value, TRUE denotes exact test, ignored if method is not NULL
method	a character value, one of 'pclt','exact.network','exact.ce','exact.mc'. If NULL method chosen by methodRule
methodRule	a function used to choose the method (see details)
control	a list with arguments that control the algortihms, see permControl
formula	a formula of the form lhs~rhs where lhs is a numeric variable giving the response scores and rhs a factor with two levels giving the corresponding groups.
data	an optional matrix or data frame containing the variables in the formula
subset	an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").
...	further arguments to be passed to or from methods.

Details

There are 4 different methods for deciding how to determine the p-value by defining which test statistics are extreme. For alternative there are 3 choices, "two.sided", "less" or "greater", but within alternative="two.sided" there are 2 methods defined by the tsmethod given within control, see permControl. If Ti is a vector of test statistics, and T0 is the observed test statistic, then alternative="less" gives p.lte=Pr[Ti<=T0], alternative="greater" gives p.gte=Pr[Ti>=T0], alternative="two.sided" with tsmethod="central" (default) gives p.twosided=max(1, 2*min(p.lte,p.gte)), and alternative="two.sided" with tsmethod="abs" gives p.twosidedAbs=Pr[abs(Ti - mean(Ti)) >=abs(T0-mean(Ti))]. For permTS the test statistic is equivalent to the mean of one group minus the mean of the other group. For permTREND the test statistic is equivalent to the correlation between the response (x) and the trend scores (y). For permKS only a twosided pvalue based on Pr[Ti>=T0] is allowed, where the test statistic, Ti, is the weighted sum of the square of the mean within group, where the weights are the sample size for each group. This will give for example, the usual Kruskal-Wallis test when the ranks are used on the responses.

Many standard statistical tests may be put into the form of the permutation test (see Graubard and Korn, 1987). There is a choice of four different methods to calculate the p-values (the last two are only available for permTS):

1. pclt: using permutational central limit theorem (see e.g., Sen, 1985).
2. exact.mc:exact using Monte Carlo.
3. exact.network: exact method using a network algorithm (see e.g., Agresti, Mehta, and Patel, 1990). Currently the network method does not implement many of the time saving suggestions such as clubbing.

4. exact.ce: exact using complete enumeration. This is good for very small sample sizes and when doing simulations, since the cm need only be calculated once for the simulation.

The `exact.network` and `exact.ce` may give errors related to running out of memory when the sample size is not small and will depend on the system you are using (e.g., about 15 in each group for `exact.network` or 14 in each group for `exact.ce`).

These associated functions for the above methods (e.g., `twosample.pclt`, `twosample.exact.network`, etc), are internal and are not to be called directly.

The `methodRule` is a function which takes the first two objects of the default implementation, and returns the method. This function can be used to appropriately choose the method based on the size of the data. For explanation of the default method rules see `methodRuleTS1`, `methodRuleKS1`, or `methodRuleTREND1`.

For more details see Fay and Shaw (2010, Section 5).

Value

An object of class `htest` or for 'exact.mc' of class `mchtest`, a list with the following elements:

`p.value`	p value associated with alternative
`alternative`	description of alternative hypothesis
`p.values`	a vector giving lower, upper, and two-sided p-values as well as p.equal which is the proportion equal to the observed test statistic
`method`	a character vector describing the test
`estimate`	an estimate of the test statistic
`statistic`	statistic used for asymptotics, either Z statistics or chi square statistic, output if method="pclt"
`parameter`	degrees of freedom for chi square statistic, output if 'statistic' is the chi square statistic
`data.name`	character vector describing the response and group variables
`p.conf.int`	a confidence interval on the p-value if method='exact.mc' (see `calcPvalsMC`)
`nmc`	number of Monte Carlo replications if method='exact.mc', NULL otherwise

Author(s)

Michael Fay

References

Agresti, A, Mehta, CR, Patel, NR (1990). JASA 85: 453-458.

Fay, MP and Shaw, PA (2010). Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The interval R package. Journal of Statistical Software. doi:10.18637/jss.v036.i02. 36 (2):1-34.

Graubard, BI, and Korn, EL (1987). Biometrics 43: 471-476.

Sen, PK (1985) 'Permutational central limit theorems' in Encyclopedia of Statistics, Vol 6.

Examples

```
## Example from StatExact manual
dBP<-c(94,108,110,90,80,94,85,90,90,90,108,94,78,105,88)
treatment<-c(rep("treated",4),rep("control",11))
permTS(dBP~treatment,alternative="less",method="pclt")
result<-permTS(dBP[treatment=="treated"],dBP[treatment=="control"],alternative="greater")
result
result$p.values
```

`permControl` *Auxiliary for controling permutation tests*

Description

A function to create a list of arguments for `permTS`, `permKS` or `permTREND`.

Usage

```
permControl(cm=NULL,nmc=10^3-1,seed=1234321,digits=12,
    p.conf.level=.99,setSEED=TRUE,tsmethod="central")
```

Arguments

`cm`	a choose(n,m) by n matrix, used if method='exact.ce', ignored otherwise
`nmc`	number of Monte Carlo replications, used if method='exact.mc', ignored otherwise
`seed`	value used in `set.seed` if method='exact.mc', ignored otherwise
`setSEED`	logical, set to FALSE when performing simulations that use method='exact.mc'
`p.conf.level`	confidence level for p value estimate, used if method='exact.mc', ignored otherwise
`digits`	number of digits to use in `signif` for precision of test statistics
`tsmethod`	method for calculating two-sided p-values, character, either 'central' or 'abs' (see details)

Details

When `cm`=NULL the resulting matrix is created by `chooseMatrix`, it may be optionally provided here only so that `chooseMatrix` does not need to be repeatedly called in simulations. Also when doing simulations with method='exact.mc', use setSEED=FALSE so that the seed is not reset to the same value each time you call the permutation test function.

See `calcPvalsMC` for description of how p.conf.level is used.

The two-sided method is given by `tsmethod`. The default 'central' two-sided method is just p=min(1, 2*min(pless,pgreater)), where pless and pgreater are the one-sided p-values. The name 'central' follows the convention of the `exact2x2` and `exactci` packages; so that for example, a two-sample permutation test on a binary response with tsmethod='central' will match the Central Fisher's Exact

test (see Fay, 2010). The option tsmethod='abs' defines another method for defining the two-sided p-value. We define it for complete enumeration, but the algorithms may differ. Let Tj be the vector of length N of all possible values of the test statistic under each of N possible permutations. The p-value with tsmethod='abs' is defined as 1/N times the number of times `abs(Tj-mean(Tj)) >= abs(T0-mean(Tj))`, where T0 is the observed value of the test statistic. This option matches the default two-sided method for the `coin` package.

Value

An list with the arguments as components.

References

Fay, M.P. (2010). Confidence intervals that match Fisher's exact or Blaker's exact tests. Biostatistics 11(2):373-373.

Index