Package: exactRankTests (via r-universe)

October 18, 2026

Title Exact Distributions for Rank and Permutation Tests

Date 2026-05-02

Version 0.8-37

Description Computes exact conditional p-values and quantiles using an implementation of the Shift-Algorithm by Streitberg & Roehmel.

Depends R (>= 2.4.0), stats, utils

Suggests survival

License GPL (>= 2)

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

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

Date/Publication 2026-05-21 12:40:34 UTC

RemoteUrl https://codeberg.org/thothorn/exactRankTests

RemoteRef HEAD

RemoteSha bd5d0fd7cd28e4cf6f200c77dd84f14dd751489e

Contents

ansari.exact . . . 2
ASAT . . . 4
bloodp . . . 5
cscores . . . 6
dperm . . . 8
ears . . . 11
glioma . . . 13
globulin . . . 14
irank . . . 15
lungcancer . . . 16
neuropathy . . . 17
ocarcinoma . . . 18
perm.test . . . 19
rotarod . . . 21
sal . . . 22
wilcox.exact . . . 23

Index 27

`ansari.exact` *Ansari-Bradley Test*

Description

Performs the Ansari-Bradley two-sample test for a difference in scale parameters for possibly tied observations.

Usage

```
## Default S3 method:
ansari.exact(x, y, alternative = c("two.sided", "less", "greater"),
             exact = NULL, conf.int = FALSE, conf.level = 0.95, ...)
## S3 method for class 'formula'
ansari.exact(formula, data, subset, na.action, ...)
```

Arguments

`x`	numeric vector of data values.
`y`	numeric vector of data values.
`alternative`	indicates the alternative hypothesis and must be one of `"two.sided"`, `"greater"` or `"less"`. You can specify just the initial letter.
`exact`	a logical indicating whether an exact p-value should be computed.
`conf.int`	a logical,indicating whether a confidence interval should be computed.
`conf.level`	confidence level of the interval.
`formula`	a formula of the form `lhs ~ rhs` where `lhs` is a numeric variable giving the data values and `rhs` a factor with two levels giving the corresponding groups.
`data`	an optional data frame containing the variables in the model 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 `NA`s. Defaults to `getOption("na.action")`.
`...`	further arguments to be passed to or from methods.

Details

Suppose that `x` and `y` are independent samples from distributions with densities $f((t-m)/s)/s$ and $f(t-m)$, respectively, where m is an unknown nuisance parameter and s, the ratio of scales, is the parameter of interest. The Ansari-Bradley test is used for testing the null that s equals 1, the two-sided alternative being that $s \neq 1$ (the distributions differ only in variance), and the one-sided alternatives being $s > 1$ (the distribution underlying `x` has a larger variance, `"greater"`) or $s < 1$ (`"less"`).

By default (if `exact` is not specified), an exact p-value is computed if both samples contain less than 50 finite values. Otherwise, a normal approximation is used.

Optionally, a nonparametric confidence interval and an estimator for s are computed. If exact p-values are available, an exact confidence interval is obtained by the algorithm described in Bauer (1972), and the Hodges-Lehmann estimator is employed. Otherwise, the returned confidence interval and point estimate are based on normal approximations.

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the Ansari-Bradley test statistic.
`p.value`	the p-value of the test.
`null.value`	the ratio of scales s under the null, 1.
`alternative`	a character string describing the alternative hypothesis.
`method`	the string `"Ansari-Bradley test"`.
`data.name`	a character string giving the names of the data.
`conf.int`	a confidence interval for the scale parameter. (Only present if argument `conf.int = TRUE`.)
`estimate`	an estimate of the ratio of scales. (Only present if argument `conf.int = TRUE`.)

Note

To compare results of the Ansari-Bradley test to those of the F test to compare two variances (under the assumption of normality), observe that s is the ratio of scales and hence s^2 is the ratio of variances (provided they exist), whereas for the F test the ratio of variances itself is the parameter of interest. In particular, confidence intervals are for s in the Ansari-Bradley test but for s^2 in the F test.

References

Myles Hollander & Douglas A. Wolfe (1973), *Nonparametric statistical inference*. New York: John Wiley & Sons. Pages 83–92.

David F. Bauer (1972), Constructing confidence sets using rank statistics. *Journal of the American Statistical Association* **67**, 687–690.

See Also

`fligner.test` for a rank-based (nonparametric) k-sample test for homogeneity of variances; `mood.test` for another rank-based two-sample test for a difference in scale parameters; `var.test` and `bartlett.test` for parametric tests for the homogeneity in variance.

Examples

```
## Hollander & Wolfe (1973, p. 86f):
## Serum iron determination using Hyland control sera
ramsay <- c(111, 107, 100, 99, 102, 106, 109, 108, 104, 99,
            101, 96, 97, 102, 107, 113, 116, 113, 110, 98)
jung.parekh <- c(107, 108, 106, 98, 105, 103, 110, 105, 104,
            100, 96, 108, 103, 104, 114, 114, 113, 108, 106, 99)
```

```
ansari.test(ramsay, jung.parekh)
ansari.exact(ramsay, jung.parekh)

ansari.exact(rnorm(20), rnorm(20, 0, 2), conf.int = TRUE)
```

ASAT *Toxicological Study on Female Wistar Rats*

Description

ASAT-Values for a new compound and a control group of 34 female Wistar rats.

Usage

```
data(ASAT)
```

Format

A data frame with 34 observations on the following 2 variables.

asat the ASAT-values (a liver enzyme)

group a factor with levels `Compound` and `Control`.

Details

The aim of this toxicological study is the proof of safety for the new compound. The data are originally given in Hothorn (1992) and reproduced in Hauschke et al. (1999).

Source

Ludwig A. Hothorn (1992), Biometrische Analyse toxikologischer Untersuchungen. In: J. Adams (ed.): *Statistisches Know how in der medizinischen Forschung.* Ullstein-Mosby, Berlin, 475–590.

References

Dieter Hauschke, Meinhard Kieser & Ludwig A. Hothorn (1999), Proof of safety in toxicology based on the ratio of two means for normally distributed data. *Biometrical Journal*, **41**(3), 295–304.

Rafael Pfl\"uger & Torsten Hothorn (2002), Assessing Equivalence Tests with Respect to their Expected p-Value. *Biometrical Journal*, **44**(8), 1002–1027.

Examples

```
set.seed(29)

data(ASAT)
# does not really look symmetric

plot(asat ~ group, data=ASAT)

# proof-of-safety based on ration of medians
pos <- wilcox.exact(I(log(asat)) ~ group, data = ASAT, alternative = "less",
             conf.int=TRUE)

# one-sided confidence set. Safety cannot be concluded since the effect of
# the compound exceeds 20% of the control median
exp(pos$conf.int)
```

---

bloodp *Diastolic Blood Pressure*

Description

Diastolic blood pressure for a two groups of patients.

Usage

```
data(bloodp)
```

Format

A data frame with 15 observations on the following 2 variables.

bp the diastolic blood pressure.

group a factor with levels `group1` and `group2`.

Details

The data is given in Table 9.6, page 227, of Metha and Pathel (2001). Note that there are some tied observations. The permutation test using the raw blood pressure values does not lead to a rejection of the null hypothesis of exchangeability: p-value = 0.1040 (two-sided) and p-value = 0.0564 (one-sided). The asymptotic two-sided p-value is 0.1070.

For the Wilcoxon-Mann-Whitney test, the one-sided p-value is 0.0542 and the two-sided one is 0.0989 (Metha & Patel, 2001, page 229).

The one-sided p-value for the v.d.Waeren test is 0.0462 (Metha & Patel, 2001, page 241) and the two-sided p-value is 0.0799.

References

Cyrus R. Mehta & Nitin R. Patel (2001), *StatXact-5 for Windows.* Manual, Cytel Software Cooperation, Cambridge, USA

Examples

```
data(bloodp)

# Permutation test

perm.test(bp ~ group, data=bloodp)
perm.test(bp ~ group, data=bloodp, alternative="greater")
perm.test(bp ~ group, data=bloodp, exact=FALSE)

# Wilcoxon-Mann-Whitney test

wilcox.exact(bp ~ group, data=bloodp, conf.int=TRUE, alternative="l")
wilcox.exact(bp ~ group, data=bloodp, conf.int=TRUE)

# compute the v.d. Waerden test

sc <- cscores(bloodp$bp, type="NormalQuantile")
X <- sum(sc[bloodp$group == "group2"])
round(pperm(X, sc, 11), 4)
## IGNORE_RDIFF_BEGIN
round(pperm(X, sc, 11, simulate=TRUE), 4)
round(pperm(X, sc, 11, alternative="two.sided"), 4)
round(pperm(X, sc, 11, alternative="two.sided", simulate=TRUE), 4)
## IGNORE_RDIFF_END

# use scores mapped into integers (cf. dperm)

sc <- cscores(bloodp$bp, type="NormalQuantile", int=TRUE)
X <- sum(sc[bloodp$group == "group2"])
round(pperm(X, sc, 11), 4)
round(pperm(X, sc, 11, alternative="two.sided"), 4)
```

`cscores` *Computation of Scores*

Description

This function can be used to compute several scores for a data vector.

Usage

```
## Default S3 method:
cscores(y, type=c("Data", "Wilcoxon", "NormalQuantile",
        "AnsariBradley", "Median", "Savage", "ConSal"), int=FALSE,
        maxs=length(y), ... )
## S3 method for class 'factor'
cscores(y, ...)
## S3 method for class 'Surv'
cscores(y, type="LogRank", int=FALSE, maxs=nrow(y), ...)
```

Arguments

y	a numeric, factor or logical vector or an object of class Surv.
type	a character string which specifies the type of the scores to be computed. Data just returns y if y is numeric.
int	a logical, forcing integer valued scores.
maxs	an integer defining the maximal value of the scores if int=TRUE.
...	additional arguments, not passed to anything at the moment.

Details

This function will serve as the basis for a more general framework of rank and permutation tests in future versions of this package. Currently, it is only used in the examples.

The logrank scores are computed as given in Hothorn & Lausen (2002).

If integer valued scores are requested (int = TRUE), the scores are mapped into integers by round(scores*length(scores) See dperm for more details.

type is self descriptive, except for ConSal which implements scores suggested by Conover & Salsburg (1988).

Value

A vector of scores for y with an attribute scores indicating the kind of scores used is returned.

References

Torsten Hothorn & Berthold Lausen (2003), On the exact distribution of maximally selected rank statistics. *Computational Statistics & Data Analysis*, **43**(2), 121-137.

William J. Conover & David S. Salsburg (1988), Locally most powerful tests for detecting treatment effects when only a subset of patients can be expected to "respond" to treatment. *Biometrics*, **44**, 189-196.

Examples

```
y <- rnorm(50)
# v.d. Waerden scores
nq <- cscores(y, type="Normal", int=TRUE)
# quantile for m=20 observations in the first group
qperm(0.1, nq, 20)
```

dperm *Distribution of One and Two Sample Permutation Tests*

Description

Density, distribution function and quantile function for the distribution of one and two sample permutation tests using the Shift-Algorithm by Streitberg & R\"ohmel.

Usage

```
dperm(x, scores, m, paired=NULL, tol = 0.01, fact=NULL, density=FALSE,
      simulate=FALSE, B=10000)
pperm(q, scores, m, paired=NULL, tol = 0.01, fact=NULL,
      alternative=c("less", "greater", "two.sided"), pprob=FALSE,
      simulate=FALSE, B=10000)
qperm(p, scores, m, paired=NULL, tol = 0.01, fact=NULL,
      simulate=FALSE, B=10000)
rperm(n, scores, m)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
scores	arbitrary scores of the observations of the x (first m elements) and y sample.
m	sample size of the x sample. If m = length(x) scores of paired observations are assumed.
paired	logical. Indicates if paired observations are used. Needed to discriminate between a paired problem and the distribution of the total sum of the scores (which has mass 1 at the point sum(scores)).
.	
tol	real. Real valued scores are mapped into integers by rounding after multiplication with an appropriate factor. Make sure that the absolute difference between the each possible test statistic for the original scores and the rounded scores is less than tol. This might not be possible due to memory/time limitations, a warning is given in this case.
fact	real. If fact is given, real valued scores are mapped into integers using fact as factor. tol is ignored in this case.
n	number of random observations to generate.
alternative	character indicating whether the probability $P(T \leq q)$ (less), $P(T \geq q)$ (greater) or a two-sided p-value (two.sided) should be computed in pperm.
pprob	logical. Indicates if the probability $P(T = q)$ should be computed additionally.
density	logical. When x is a scalar and density is TRUE, dperm returns the density for all possible statistics less or equal x as a data frame.
simulate	logical. Use conditional Monte-Carlo to compute the distribution.
B	number of Monte-Carlo replications to be used.

Details

The exact distribution of the sum of the first `m` scores is evaluated using the Shift-Algorithm by Streitberg & R\"ohmel under the hypothesis of exchangeability (or, equivalent, the hypothesis that all permutations of the scores are equally likely). The algorithm is able to deal with tied scores, so the conditional distribution can be evaluated.

The algorithm is defined for positive integer valued scores only. There are two ways dealing with real valued scores. First, one can try to find integer valued scores that lead to statistics which differ not more than `tol` from the statistics computed for the original scores. This can be done as follows.

Without loss of generality let $a_i > 0$ denote real valued scores in reverse ordering and f a positive factor (this is the `fact` argument). Let $R_i = f \cdot a_i - round(f \cdot a_i)$. Then

$$\sum_{i=1}^{m} f \cdot a_i = \sum_{i=1}^{m} round(f \cdot a_i) - R_i.$$

Clearly, the maximum difference between $1/f \sum_{i=1}^{m} f \cdot a_i$ and $1/f \sum_{i=1}^{n} round(f \cdot a_i)$ is given by $|\sum_{i=1}^{m} R_i|$. Therefore one searches for f with

$$|\sum_{i=1}^{m} R_i| \leq \sum_{i=1}^{m} |R_i| \leq tol.$$

If f induces more that 100.000 columns in the Shift-Algorithm by Streitberg & R\"ohmel, f is restricted to the largest integer that does not.

The second idea is to map the scores into integers by taking the integer part of $a_i N / \max(a_i)$ (Hothorn & Lausen, 2002). This induces additional ties, but the shape of the scores is very similar. That means we do not try to approximate something but use a different test (with integer valued scores), serving for the same purpose (due to a similar shape of the scores). However, this has to be done prior to calling `pperm` (see the examples).

Exact two-sided p-values are computed as suggested in the StatXact-5 manual, page 225, equation (9.31) and equation (8.18), p. 179 (paired case). In detail: For the paired case the two-sided p-value is just twice the one-sided one. For the independent sample case the two sided p-value is defined as

$$p_2 = P(|T - E(T)| \geq |q - E(T)|)$$

where q is the quantile passed to `pperm`.

Value

`dperm` gives the density, `pperm` gives the distribution function and `qperm` gives the quantile function. If `pprob` is true, `pperm` returns a list with elements

`PVALUE`	the probability specified by `alternative`.
`PPROB`	the probability $P(T = q)$.

`rperm` is a wrapper to `sample`.

References

Bernd Streitberg & Joachim R\"ohmel (1986), Exact distributions for permutations and rank tests: An introduction to some recently published algorithms. *Statistical Software Newsletter* **12**(1), 10–17.

Bernd Streitberg & Joachim R\"ohmel (1987), Exakte Verteilungen f\"ur Rang- und Randomisierungstests im allgemeinen c-Stichprobenfall. *EDV in Medizin und Biologie* **18**(1), 12–19.

Torsten Hothorn (2001), On exact rank tests in R. *R News* **1**(1), 11–12.

Cyrus R. Mehta & Nitin R. Patel (2001), *StatXact-5 for Windows.* Manual, Cytel Software Cooperation, Cambridge, USA

Torsten Hothorn & Berthold Lausen (2003), On the exact distribution of maximally selected rank statistics. *Computational Statistics & Data Analysis*, **43**(2), 121-137.

Examples

```
# exact one-sided p-value of the Wilcoxon test for a tied sample

x <- c(0.5, 0.5, 0.6, 0.6, 0.7, 0.8, 0.9)
y <- c(0.5, 1.0, 1.2, 1.2, 1.4, 1.5, 1.9, 2.0)
r <- cscores(c(x,y), type="Wilcoxon")
pperm(sum(r[seq(along=x)]), r, 7)

# Compare the exact algorithm as implemented in ctest and the
# Shift-Algorithm by Streitberg & Roehmel for untied samples

# Wilcoxon:

n <- 10
x <- rnorm(n, 2)
y <- rnorm(n, 3)
r <- cscores(c(x,y), type="Wilcoxon")

# exact distribution using the Shift-Algorithm

dwexac <- dperm((n*(n+1)/2):(n^2 + n*(n+1)/2), r, n)
sum(dwexac)           # should be something near 1 :-)

# exact distribution using dwilcox

dw <- dwilcox(0:(n^2), n, n)

# compare the two distributions:

plot(dw, dwexac, main="Wilcoxon", xlab="dwilcox", ylab="dperm")
# should give a "perfect" line

# Wilcoxon signed rank test

n <- 10
x <- rnorm(n, 5)
y <- rnorm(n, 5)
```

```
r <- cscores(abs(x - y), type="Wilcoxon")
pperm(sum(r[x - y > 0]), r, length(r))
wilcox.test(x,y, paired=TRUE, alternative="less")
psignrank(sum(r[x - y > 0]), length(r))

# Ansari-Bradley

n <- 10
x <- rnorm(n, 2, 1)
y <- rnorm(n, 2, 2)

# exact distribution using the Shift-Algorithm

sc <- cscores(c(x,y), type="Ansari")
dabexac <- dperm(0:(n*(2*n+1)/2), sc, n)
sum(dabexac)

# real scores are allowed (but only result in an approximation)
# e.g. v.d. Waerden test

n <- 10
x <- rnorm(n)
y <- rnorm(n)
scores <- cscores(c(x,y), type="NormalQuantile")
X <- sum(scores[seq(along=x)])  # <- v.d. Waerden normal quantile statistic

# critical value, two-sided test

abs(qperm(0.025, scores, length(x)))

# p-values

p1 <- pperm(X, scores, length(x), alternative="two.sided")

# generate integer valued scores with the same shape as normal quantile
# scores, this no longer v.d.Waerden, but something very similar

scores <- cscores(c(x,y), type="NormalQuantile", int=TRUE)

X <- sum(scores[seq(along=x)])
p2 <- pperm(X, scores, length(x), alternative="two.sided")

# compare p1 and p2

p1 - p2
```

---

ears *Survival of Ventilating Tubes*

Description

Survival times of ventilating tubes of left and right ears in 78 children with otitis media.

Usage

```
data(ears)
```

Format

A data frame with 78 observations on the following 5 variables.

left Survival time in month of tube in left ear.

lcens Censoring indicator for left ear: `0` censored and `1` event.

right Survival time in month of tube in right ear.

rcens Censoring indicator for right ear: `0` censored and `1` event.

group a factor with levels `control` and `treat`.

Source

Sin-Ho Jung and Jong-Hyeon Jeong (2003). Rank tests for clustered survival data. *Lifetime Data Analysis*, **9**, 21-33.

References

V.M. Howie and R.H. Schwarz (1983). Acute otitis media: One year in general pediatric practice. *American Journal of Diseases in Children*, **137**, 155-158.

D.W. Teele, J.O. Klein, B. Rosner et al. (1989). Epidemiology of otitis media during the first seven years of life in children in greater Boston. *Journal of Infectious Diseases*, **160**, 89-94.

Examples

```
data(ears)
if (require(survival, quietly=TRUE)) {
  ls <- cscores(Surv(ears$left, ears$lcens), int=TRUE)
  perm.test(ls ~ group, data=ears)
}
```

glioma *Malignant Glioma Pilot Study*

Description

A non-randomized pilot study on malignant glioma patients with pretargeted adjuvant radioimmunotherapy using Yttrium-90-biotin.

Usage

```
data(glioma)
```

Format

A data frame with 37 observations on the following 7 variables.

No. patient number.

Age patients ages in years.

Sex a factor with levels `F`(emale) and `M`(ale).

Histology a factor with levels `GBM` (grade IV) and `Grade3` (grade III)

Survival survival times in month.

Cens censoring indicator: `0` censored and `1` dead.

Group a factor with levels `Control` and `RIT`.

Details

The primary endpoint of this small pilot study is survival. Survival times are tied, the usual asymptotic log-rank test may be inadequate in this setup. Therefore, a permutation test (via Monte-Carlo sampling) was conducted in the original paper. The data are taken from Tables 1 and 2 of Grana et al. (2002).

Source

C. Grana, M. Chinol, C. Robertson, C. Mazzetta, M. Bartolomei, C. De Cicco, M. Fiorenza, M. Gatti, P. Caliceti & G. Paganelli (2002), Pretargeted adjuvant radioimmunotherapy with Yttrium-90-biotin in malignant glioma patients: A pilot study. *British Journal of Cancer*, **86**(2), 207–212.

Examples

```
data(glioma)

if(require(survival, quietly = TRUE)) {

  par(mfrow=c(1,2))

  # Grade III glioma
  g3 <- glioma[glioma$Histology == "Grade3",]
```

```
  # Plot Kaplan-Meier curves
  plot(survfit(Surv(Survival, Cens) ~ Group, data=g3),
       main="Grade III Glioma", lty=c(2,1),
       legend.text=c("Control", "Treated"),
       legend.bty=1, ylab="Probability",
       xlab="Survival Time in Month")

  # log-rank test
  survdiff(Surv(Survival, Cens) ~ Group, data=g3)

  # permutation test with integer valued log-rank scores
  lsc <- cscores(Surv(g3$Survival, g3$Cens), int=TRUE)
  perm.test(lsc ~ Group, data=g3)

  # permutation test with real valued log-rank scores
  lsc <- cscores(Surv(g3$Survival, g3$Cens), int=FALSE)
  tr <- (g3$Group == "RIT")
  T <- sum(lsc[tr])
  pperm(T, lsc, sum(tr), alternative="tw")
  pperm(T, lsc, sum(tr), alternative="tw", simulate=TRUE)

  # Grade IV glioma
  gbm <- glioma[glioma$Histology == "GBM",]

  # Plot Kaplan-Meier curves
  plot(survfit(Surv(Survival, Cens) ~ Group, data=gbm),
       main="Grade IV Glioma", lty=c(2,1),
       legend.text=c("Control", "Treated"),
       legend.bty=1, legend.pos=1, ylab="Probability",
       xlab="Survival Time in Month")

  # log-rank test
  survdiff(Surv(Survival, Cens) ~ Group, data=gbm)

  # permutation test with integer valued log-rank scores
  lsc <- cscores(Surv(gbm$Survival, gbm$Cens), int=TRUE)
  perm.test(lsc ~ Group, data=gbm)

  # permutation test with real valued log-rank scores
  lsc <- cscores(Surv(gbm$Survival, gbm$Cens), int=FALSE)
  tr <- (gbm$Group == "RIT")
  T <- sum(lsc[tr])
  pperm(T, lsc, sum(tr), alternative="tw")
  pperm(T, lsc, sum(tr), alternative="tw", simulate=TRUE)
}
```

---

globulin *Differences in Globulin Fraction in Two Groups*

Description

Globulin fraction of plasma (g/l) in two groups of 10 patients.

Usage

```
data(globulin)
```

Format

This data frame contains the following variables:

gfrac Globulin fraction of plasma

group a factor with levels `group1` and `group2`

Details

See page 75 of Gardner & Altman (1989).

Source

M. J. Gardner & D. G. Altman (1989), *Statistics with Confidence*. Published by the British Medical Journal.

References

Joachim R\"ohmel (1996), Precision intervals for estimates of the difference in success rates for binary random variables based on the permutation principle. *Biometrical Journal*, **38**(8), 977–993.

Examples

```
data(globulin)
perm.test(gfrac ~ group, data=globulin, conf.int=TRUE)
```

---

irank *Integer Ranks*

Description

Compute the number of elements less or equal the elements in a given vector.

Usage

```
irank(x, ox = NULL)
```

Arguments

`x`	a numeric vector.
`ox`	`order(x)`, optionally (for efficiency in case `order(x)` is already known).

Value

A vector of integers.

Examples

```
x <- rnorm(10)
irank(x)
rank(x)
x <- c(1,2,3,3,0)
irank(x)
rank(x)
```

`lungcancer` *Lung Cancer Clinical Trial*

Description

Survival times for patients suffering lung cancer for a treatment and control group.

Usage

```
data(lungcancer)
```

Format

A data frame with 14 observations on the following 3 variables.

time survival time in days.

cens censoring indicator: 0 censored, 1 event.

group a factor with levels `control` and `newdrug`.

Details

The data is given in Table 9.19, page 293, of Metha and Pathel (2001). The two-sided p-value for the log-rank test is 0.001 (page 295).

References

Cyrus R. Mehta & Nitin R. Patel (2001), *StatXact-5 for Windows.* Manual, Cytel Software Cooperation, Cambridge, USA

Examples

```
data(lungcancer)
attach(lungcancer)

# round logrank scores
scores <- cscores.Surv(cbind(time, cens))
T <- sum(scores[group=="newdrug"])
mobs <- sum(group=="newdrug")
(prob <- pperm(T, scores, m=mobs, al="le"))
pperm(T, scores, m=mobs, al="tw")
pperm(T, scores, m=mobs, al="tw", simulate=TRUE)

# map into integers, faster
scores <- cscores.Surv(cbind(time, cens), int=TRUE)
T <- sum(scores[group=="newdrug"])
mobs <- sum(group=="newdrug")
(prob <- pperm(T, scores, m=mobs, al="le"))
pperm(T, scores, m=mobs, al="tw")
pperm(T, scores, m=mobs, al="tw", simulate=TRUE)

detach(lungcancer)
```

`neuropathy` *Acute Painful Diabetic Neuropathy*

Description

The logarithm of the ratio of pain scores at baseline and after four weeks for a control and treatment group.

Usage

```
data(neuropathy)
```

Format

A data frame with 58 observations on the following 2 variables.

pain Pain scores: ln(baseline/final).

group a factor with levels `control` and `treat`.

Details

Data from Table 1 of Conover & Salsburg (1988).

Source

William J. Conover and David S. Salsburg (1988), Locally most powerful tests for detecting treatment effects when only a subset of patients can be expected to "respond" to treatment. *Biometrics*, **44**, 189–196.

Examples

```
data(neuropathy)
# compare with Table 2 of Conover & Salsburg (1988)
wilcox.exact(pain ~ group, data=neuropathy, alternative="less")
css <- cscores(neuropathy$pain, type="ConSal")
pperm(sum(css[neuropathy$group=="control"]),css,
      m=sum(neuropathy$group=="control"))
```

---

ocarcinoma *Ovarian Carcinoma*

Description

Survival times of 35 women suffering ovarian carcinoma at stadium II and IIA.

Usage

```
data(ocarcinoma)
```

Format

A data frame with 35 observations on the following 3 variables.

time time in days.

cens censoring indicator: 0 censored, 1 event.

stadium a factor with levels `II` and `IIA`.

Details

Data from Fleming et al. (1980, 1984), reanalysed in Schumacher and Schulgen (2002).

Source

Thomas R. Fleming, Judith R. O'Fallon, Peter C. O'Brien & David P. Harrington (1980), Modified Kolmogorov-Smirnov test procedures with applications to arbitrarily censored data. *Biometrics*, **36**, 607–625.

Thomas R. Fleming, Stephanie J. Green & David P. Harrington (1984), Considerations of monitoring and evaluating treatment effects in clinical trials. *Controlled Clinical Trials*, **5**, 55–66.

References

Martin Schumacher & Gabi Schulgen (2002), *Methodik klinischer Studien: methodische Grundlagen der Planung, Durchf\"uhrung und Auswertung.* Springer, Heidelberg.

Examples

```
data(ocarcinoma)
attach(ocarcinoma)
# compute integer valued logrank scores
logrsc <- cscores.Surv(cbind(time, cens), int=TRUE)
# the test statistic
lgT <- sum(logrsc[stadium == "II"])
# p-value
round(pperm(lgT, logrsc, m=sum(stadium=="II"), al="tw"), 4)

# compute logrank scores and simulate p-value
logrsc <- cscores.Surv(cbind(time, cens), int=FALSE)
# the test statistic
lgT <- sum(logrsc[stadium == "II"])
# p-value
round(pperm(lgT, logrsc, m=sum(stadium=="II"), al="tw", simulate=TRUE), 4)
```

---

perm.test *One and Two Sample Permutation Test*

Description

Performs the permutation test for the one and two sample problem.

Usage

```
## Default S3 method:
perm.test(x, y, paired=FALSE, alternative=c("two.sided", "less", "greater"),
          mu=0, exact=NULL, conf.int=FALSE, conf.level=0.95, tol=NULL, ...)
## S3 method for class 'formula'
perm.test(formula, data, subset, na.action, ...)
```

Arguments

x	numeric vector of integer data values.
y	numeric vector of integer data values.
paired	a logical indicating whether you want a paired test.
alternative	the alternative hypothesis must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
mu	a number specifying an optional location parameter.
exact	a logical indicating whether an exact p-value should be computed.

conf.int	a logical indicating whether a confidence interval should be computed.
conf.level	confidence level of the interval.
tol	real. real valued scores are mapped into integers by multiplication. Make sure that the absolute difference between the "true" quantile and the approximated quantile is less than tol. This might not be possible due to memory/time limitations. See pperm.
formula	a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs a factor with two levels giving the corresponding groups.
data	an optional data frame containing the variables in the model 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

The permutation test is performed for integer valued observations or scores. If real values x or y are passed to this function the following applies: if exact is true (i.e. the sample size is less than 50 observations) and tol is not given, the scores are mapped into $\{1, \ldots, N\}$, see pperm for the details. Otherwise the p-values are computed using tol. If the sample size exceeds 50 observations, the usual normal approximation is used.

P-values are computed according to the StatXact-manual, see pperm.

For (in principle) continuous variables the confidence sets represent the "largest shift in location being consistent with the observations". For discrete variables with only a few categories they are hard to interpret. In the case of binary data (e.g. success / failure) the confidence sets can be interpreted as the differences of two success-rates covered by the data. For a detailed description see R\"ohmel (1996).

Confidence intervals are only available for independent samples. When the sample sizes are unbalanced, length(x) needs to be smaller than length(y).

Value

A list with class "htest" containing the following components:

statistic	the value of the test statistic with a name describing it.
p.value	the p-value for the test.
pointprob	this gives the probability of observing the test statistic itself.
null.value	the location parameter mu.
alternative	a character string describing the alternative hypothesis.
method	the type of test applied.
data.name	a character string giving the names of the data.
conf.int	a confidence interval for the location parameter. (Only present if argument conf.int = TRUE.)

Note

Confidence intervals may need some cpu-time ...

References

Joachim R\"ohmel (1996), Precision intervals for estimates of the difference in success rates for binary random variables based on the permutation principle. *Biometrical Journal*, **38**(8), 977–993.

Cyrus R. Mehta & Nitin R. Patel (2001), *StatXact-5 for Windows.* Manual, Cytel Software Cooperation, Cambridge, USA

Examples

```
# Example from Gardner & Altman (1989), p. 30
# two treatments A and B, 1 means improvement, 0 means no improvement
# confidence sets cf. R\"ohmel (1996)

A <- c(rep(1, 61), rep(0, 19))
B <- c(rep(1, 45), rep(0, 35))
perm.test(A, B, conf.int=TRUE, exact=TRUE)

# one-sample AIDS data (differences only), Methta and Patel (2001),
# Table 8.1 page 181

data(sal)
attach(sal)
ppdiff <- pre - post
detach(sal)

# p-values in StatXact == 0.0011 one-sided, 0.0021 two.sided, page 183

perm.test(ppdiff)
perm.test(ppdiff, alternative="less")
perm.test(ppdiff, exact=FALSE)
```

`rotarod` *Rotating Rats Data*

Description

The endurance time of 24 rats in two groups in a rotating cylinder.

Usage

```
data(rotarod)
```

Format

A data frame with 24 observations on the following 2 variables.

time the endurance time

group a factor with levels `control` and `treatment`.

Details

The 24 rats received a fixed oral dose of a centrally acting muscle relaxant (treatment) or a saline solvent (control). They were placed on a rotating cylinder and the length of time each rat remains on the cylinder is measured, up to a maximum of 300 seconds. The rats were randomly assigned to the control and treatment group.

Note that the empirical variance in the control group is 0 and that the group medians are identical.

This dataset serves as the basis of an comparision of the results of the Wilcoxon-Mann-Whitney test computed by 11 statistical packages in Bergmann et al. (2000). The exact conditional p-value is 0.0373 (two-sided) and 0.0186 (one-sided). The asymptotic two-sided p-value (corrected for ties) is reported as 0.0147.

Source

Reinhard Bergmann, John Ludbrook & Will P.J.M. Spooren (2000), Different outcomes of the Wilcoxon-Mann-Whitney test from different statistics packages. *The American Statistician*, **54**(1), 72–77.

Examples

```
data(rotarod)
wilcox.exact(time ~ group, data=rotarod, alternative="g")
wilcox.exact(time ~ group, data=rotarod, conf.int=TRUE)
wilcox.exact(time ~ group, data=rotarod, exact=FALSE)
# the permutation test
perm.test(time ~ group, data=rotarod)
perm.test(time ~ group, data=rotarod, exact=FALSE)
```

---

sal *Serum Antigen Level*

Description

The response of serum antigen level to AZT in 20 patients suffering AIDS.

Usage

```
data(sal)
```

Format

A data frame with 20 observations on the following 2 variables.

pre level pre treatment.

post level post treatment.

Details

The data is given in Metha and Patel (2001), Table 8.1, page 181. Two-sided p-value for the Wilcoxon-Signed Rank Test: 0.0021 (page 183) or 0.0038 (asymptotically).

References

Cyrus R. Mehta & Nitin R. Patel (2001), *StatXact-5 for Windows.* Manual, Cytel Software Cooperation, Cambridge, USA

Examples

```
data(sal)
attach(sal)

wilcox.exact(pre, post, paired=TRUE, conf.int=TRUE)
wilcox.exact(pre,post, paired=TRUE, conf.int=TRUE, exact=FALSE)

detach(sal)
```

---

wilcox.exact *Wilcoxon Rank Sum and Signed Rank Tests*

Description

Performs one and two sample Wilcoxon tests on vectors of data for possibly tied observations.

Usage

```
## Default S3 method:
wilcox.exact(x, y = NULL, alternative = c("two.sided", "less", "greater"),
             mu = 0, paired = FALSE, exact = NULL,
             conf.int = FALSE, conf.level = 0.95, ...)
## S3 method for class 'formula'
wilcox.exact(formula, data, subset, na.action, ...)
```

Arguments

x	numeric vector of data values.
y	an optional numeric vector of data values.
alternative	the alternative hypothesis must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
mu	a number specifying an optional location parameter.
paired	a logical indicating whether you want a paired test.
exact	a logical indicating whether an exact p-value should be computed.
conf.int	a logical indicating whether a confidence interval should be computed.
conf.level	confidence level of the interval.
formula	a formula of the form lhs ~ rhs where lhs is a numeric variable giving the data values and rhs a factor with two levels giving the corresponding groups.
data	an optional data frame containing the variables in the model 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

This version computes exact conditional (on the data) p-values and quantiles using the Shift-Algorithm by Streitberg & R\"ohmel for both tied and untied samples.

If only x is given, or if both x and y are given and paired is TRUE, a Wilcoxon signed rank test of the null that the median of x (in the one sample case) or of x-y (in the paired two sample case) equals mu is performed.

Otherwise, if both x and y are given and paired is FALSE, a Wilcoxon rank sum test (equivalent to the Mann-Whitney test) is carried out. In this case, the null hypothesis is that the location of the distributions of x and y differ by mu.

By default (if exact is not specified), an exact p-value is computed if the samples contain less than 50 finite values and there are no ties. Otherwise, a normal approximation is used.

Optionally (if argument conf.int is true), a nonparametric confidence interval for the median (one-sample case) or for the difference of the location parameters x-y is computed. If exact p-values are available, an exact confidence interval is obtained by the algorithm described in Bauer (1972). Otherwise, an asymptotic confidence interval is returned.

Value

A list with class "htest" containing the following components:

statistic	the value of the test statistic with a name describing it.
p.value	the p-value for the test.
pointprob	this gives the probability of observing the test statistic itself (called point-prob).
null.value	the location parameter mu.

`alternative`	a character string describing the alternative hypothesis.
`method`	the type of test applied.
`data.name`	a character string giving the names of the data.
`conf.int`	a confidence interval for the location parameter. (Only present if argument `conf.int = TRUE`.)
`estimate`	Hodges-Lehmann estimate of the location parameter. (Only present if argument `conf.int = TRUE`.)

References

Myles Hollander & Douglas A. Wolfe (1973), *Nonparametric statistical inference*. New York: John Wiley & Sons. Pages 27–33 (one-sample), 68–75 (two-sample).

David F. Bauer (1972), Constructing confidence sets using rank statistics. *Journal of the American Statistical Association* **67**, 687–690.

Cyrus R. Mehta & Nitin R. Patel (2001), *StatXact-5 for Windows.* Manual, Cytel Software Cooperation, Cambridge, USA

See Also

`perm.test` for the one and two sample permutation test.

Examples

```
## One-sample test.
## Hollander & Wolfe (1973), 29f.
## Hamilton depression scale factor measurements in 9 patients with
##  mixed anxiety and depression, taken at the first (x) and second
##  (y) visit after initiation of a therapy (administration of a
##  tranquilizer).
x <- c(1.83,  0.50,  1.62,  2.48, 1.68, 1.88, 1.55, 3.06, 1.30)
y <- c(0.878, 0.647, 0.598, 2.05, 1.06, 1.29, 1.06, 3.14, 1.29)
wilcox.exact(x, y, paired = TRUE, alternative = "greater")
wilcox.exact(y - x, alternative = "less")    # The same.

## Two-sample test.
## Hollander & Wolfe (1973), 69f.
## Permeability constants of the human chorioamnion (a placental
##  membrane) at term (x) and between 12 to 26 weeks gestational
##  age (y).  The alternative of interest is greater permeability
##  of the human chorioamnion for the term pregnancy.
x <- c(0.80, 0.83, 1.89, 1.04, 1.45, 1.38, 1.91, 1.64, 0.73, 1.46)
y <- c(1.15, 0.88, 0.90, 0.74, 1.21)
wilcox.exact(x, y, alternative = "g")        # greater

## Formula interface.
data(airquality)
boxplot(Ozone ~ Month, data = airquality)
wilcox.exact(Ozone ~ Month, data = airquality,
             subset = Month %in% c(5, 8))
```

```
# Hollander & Wolfe, p. 39, results p. 40 and p. 53

x <- c(1.83, 0.50, 1.62, 2.48, 1.68, 1.88, 1.55, 3.06, 1.30)
y <- c(0.878, 0.647, 0.598, 2.05, 1.06, 1.29, 1.06, 3.14, 1.29)

wilcox.exact(y,x, paired=TRUE, conf.int=TRUE)

# Hollander & Wolfe, p. 110, results p. 111 and p. 126

x <- c(0.8, 0.83, 1.89, 1.04, 1.45, 1.38, 1.91, 1.64, 0.73, 1.46)
y <- c(1.15, 0.88, 0.90, 0.74, 1.21)

wilcox.exact(y,x, conf.int=TRUE)
```

Index

∗ **datasets**
  ASAT, 4
  bloodp, 5
  ears, 11
  glioma, 13
  globulin, 14
  lungcancer, 16
  neuropathy, 17
  ocarcinoma, 18
  rotarod, 21
  sal, 22
∗ **distribution**
  dperm, 8
∗ **htest**
  ansari.exact, 2
  perm.test, 19
  wilcox.exact, 23
∗ **misc**
  cscores, 6
∗ **univar**
  irank, 15

ansari.exact, 2
ASAT, 4

bartlett.test, *3*
bloodp, 5

cscores, 6

dperm, *7*, 8

ears, 11

fligner.test, *3*

glioma, 13
globulin, 14

irank, 15

lungcancer, 16

mood.test, *3*

neuropathy, 17

ocarcinoma, 18

perm.test, 19, *25*
pperm, *20*
pperm *(*dperm*)*, 8

qperm *(*dperm*)*, 8

rotarod, 21
rperm *(*dperm*)*, 8

sal, 22
sample, *9*

var.test, *3*

wilcox.exact, 23