Package ‘testtwice’

September 15, 2020

Type Package

Title Testing One Hypothesis Twice in Observational Studies

Version 1.0.3

Author Paul R. Rosenbaum

Maintainer Paul R. Rosenbaum <rosenbaum@wharton.upenn.edu>

Description Tests one hypothesis with several test statistics, correcting for multiple testing. The central function in the package is testtwice(). In a sensitivity analysis, the method has the largest design sensitivity of its component tests. The package implements the method and examples in Rosenbaum, P. R. (2012) <doi:10.1093/biomet/ass032> Testing one hypothesis twice in observational studies. Biometrika, 99(4), 763-774.

License GPL-2

Encoding UTF-8

LazyData true

Imports stats, mvtnorm

NeedsCompilation no

Repository CRAN

Date/Publication 2020-09-15 10:00:02 UTC

R topics documented:

testtwice-package .................................................. 2
bmhranks .......................................................... 3
multrnk ........................................................... 5
smokerlead ......................................................... 8
testtwice .......................................................... 9
tt ................................................................. 13

Index 16
Description

Tests one hypothesis with several test statistics, correcting for multiple testing. The central function in the package is testtwice(). In a sensitivity analysis, the method has the largest design sensitivity of its component tests. The package implements the method and examples in Rosenbaum, P. R. (2012) <doi:10.1093/biomet/ass032> Testing one hypothesis twice in observational studies. Biometrika, 99(4), 763-774.

Details

The DESCRIPTION file:

Package: testtwice
Type: Package
Title: Testing One Hypothesis Twice in Observational Studies
Version: 1.0.3
Author: Paul R. Rosenbaum
Maintainer: Paul R. Rosenbaum <rosenbaum@wharton.upenn.edu>
Description: Tests one hypothesis with several test statistics, correcting for multiple testing. The central function in the package is testtwice(). In a sensitivity analysis, the method has the largest design sensitivity of its component tests. The package implements the method and examples in Rosenbaum, P. R. (2012) <doi:10.1093/biomet/ass032> Testing one hypothesis twice in observational studies. Biometrika, 99(4), 763-774.
License: GPL-2
Encoding: UTF-8
LazyData: true
Imports: stats, mvtnorm

Index of help topics:

- bmhranks One-step and Two-step Signed Ranks.
- multrnk A family of Signed Ranks for Matched Pair Differences.
- smokerlead Pair Differences in Blood Lead Levels in 679 Matched Pairs of One Smoker and One Control.
- testtwice Computes the P-value and Sensitivity Bound for Testing Twice.
- testtwice-package Testing One Hypothesis Twice in Observational Studies
- tt Computes the P-value and Sensitivity Bound for Testing Twice From a User Supplied Matrix.

Author(s)

Paul R. Rosenbaum
Maintainer: Paul R. Rosenbaum <rosenbaum@wharton.upenn.edu>
References


Examples

data(smokerlead)
attach(smokerlead)

# The following example reproduces parts of the first column of Table 3 in Rosenbaum (2012).
w<-rank(abs(lead))
wd<-rank(abs(lead))*dose
wl<-rank(abs(llead))
H<-cbind(w,wd,wl)

##tt(lead,H,gamma=2.8)
##tt(lead,H,gamma=3.2)

rm(w,wd,wl,H)
detach(smokerlead)

bmhranks

One-step and Two-step Signed Ranks.

Description

Calculates either one-step or two-step signed ranks as proposed by Noether (1973), Brown (1981) and Markowski and Hettmansperger (1982).

Usage

bmhranks(y, q1 = 1/3, q2 = 2/3)

Arguments

y A vector of matched pair differences.
q1 Quantile at which step-ranks rise from 0 to 1.
q2 Quantile at which step-ranks rise from 1 to 2. If q1=q2, then there is only one step, from 0 to 1.

Value

A vector with the same length as y containing the step-ranks.

Note

The use of step-ranks in observational studies is discussed in Rosenbaum (1999, 2012, 2015). They can have larger design sensitivity than Wilcoxon ranks and higher Bahadur efficiency in a sensitivity analysis.
Author(s)
Paul R. Rosenbaum

References

See Also
The function multim() computes an alternative family of signed ranks.

Examples

```r
data(smokerlead)
attach(smokerlead)

w<-rank(abs(llead)) # Wilcoxon ranks
o<-order(w)
brn<-bmhranks(llead) # Brown (1981) ranks
plot(w[o],brn[o],type="n",xlab="Wilcoxon ranks",ylab="Step Ranks",
     main="Comparison of 3 Step Ranks")
lines(w[o],brn[o],col="black")

# The following two-step ranks were best for Normal data in
# Table 2 of Markowski–Hettmansperger (1982).

mhn<-bmhranks(llead,q1=.4,q2=.8)
lines(w[o],mhn[o],col="blue")

# Noether (1973) ranks take a single step. The case of a step
# at q1=q2=2/3 was evaluated in Rosenbaum (2015, Table 2).

noe<-bmhranks(llead,q1=2/3,q2=2/3)
lines(w[o],noe[o],col="red")
legend(20,1.75,c("Brown","MH","Noether"),
```

The adaptive choice of Brown or Noether ranks was considered in Rosenbaum (2012). In this case, an exact distribution is available, but function tt() uses a limiting Normal distribution instead.

```r
H <- cbind(brn, noe)
tt(llead, H, gamma = 1)
tt(llead, H, gamma = 3.7)
rm(w, brn, mhn, noe, H)
detach(smokerlead)
```

---

**multrnk**

* A family of Signed Ranks for Matched Pair Differences.

**Description**

Computes a family of signed ranks that includes the ranks of Wilcoxon, Stephenson (1981) and Rosenbaum (2011).

**Usage**

```r
multrnk(y, m1 = 2, m2 = 2, m = 2, exact = FALSE)
```

**Arguments**

- **y**
  - A vector of matched pair differences.

- **m1**
  - See m.

- **m2**
  - See m.

- **m**
  - The three integers (m, m1, m2) with 1 <= m1 <= m2 <= m define a U-statistic and an associated signed rank statistic. See Details.

- **exact**
  - In large samples, it is appropriate to set exact = FALSE, and this is the default. When testing hypotheses, it is reasonable to set exact = FALSE for all sample sizes.

  If exact = FALSE, then the rank scores are approximated using expression (9) in Rosenbaum (2011). If exact = TRUE, then ranks are proportional to (8) in Rosenbaum (2011). The exact ranks involve combinatorial coefficients that grow very quickly with the sample size, whereas the approximate ranks do not have this property.

  The confidence intervals for attributable effects in Rosenbaum (2003; 2007; 2011, Appendix) are based on the exact ranks.
Details

Setting m1=2, m2=2, and m=2 yields the U-statistic that is nearly the same as Wilcoxon’s signed rank statistic. See Lehmann (1975, Appendix, Example 6) or Pratt and Gibbons (1981, Section 3.5). Setting m1=m2=m>=2 yields the method of Stephenson (1981). The ranks used by Stephenson (1981) are closely related to ranks proposed by Conover and Salsburg (1988) who aimed for higher power when only a subset of treated individuals are affected by the treatment; see Rosenbaum (2007).

Stephenson (1981) looks at m differences in y, finds the one pair difference with the largest absolute value, and score a 1 if the difference is positive or a zero if it is negative. He sums this score over all subsets of m distinct differences.

The general case, 1<=m1<=m2<=m is discussed in Rosenbaum (2011), with further evaluation of performance in Rosenbaum (2015). The statistic in Rosenbaum (2011) may be inverted to obtain a confidence interval for an attributable effect; see also Rosenbaum (2003) for the special case of m1=m2=m=2 analogous to Wilcoxon’s test.

For instance, the statistic (8,5,8) has Pitman efficiency of .97 compared to Wilcoxon for Normal or logistic errors in a randomized experiment, but has higher design sensitivity when used in a sensitivity analysis in an observational study; see Rosenbaum (2011, Tables 1 and 3).

Value

A vector with the same length as y containing the ranks.

Note

The performance of various ranks when used in sensitivity analyses is discussed in Rosenbaum (2015).

Author(s)

Paul R. Rosenbaum

References


See Also

An alternative family of signed ranks is calculated by the function bmhranks().

Examples

# The following example reproduces part of column 3 of Table 3 of Rosenbaum (2012).
data("smokerlead")
attach(smokerlead)
u878 <- multrnk(lead, m1 = 7, m2 = 8, m = 8)
u867 <- multrnk(lead, m1 = 6, m2 = 7, m = 8)
u878l <- multrnk(llead, m1 = 7, m2 = 8, m = 8)
u867l <- multrnk(llead, m1 = 6, m2 = 7, m = 8)
H <- cbind(u878, u867, u878l, u867l)
tt(lead, H, gamma = 3)
tt(lead, H, gamma = 3.4)
rm(u878, u867, u878l, u867l, H)
detach(smokerlead)
#
# ----------------------
# The following examples are intended to aid understanding of
# some of the technical details.
#
# Exact and approximate ranks
# Exact and approximate ranks are highly correlated.
a <- multrnk(1:50, m1 = 4, m2 = 5, m = 5, exact = FALSE)
b <- multrnk(1:50, m1 = 4, m2 = 5, m = 5, exact = TRUE)
cor(a, b)

# Compare the following with Section 3.5 in Pratt and Gibbons (1981)
multrnk(1:10, exact = TRUE)

# Stephenson (1981) ranks for m=5 with 10 pair differences.
a <- multrnk(1:10, m1 = 5, m2 = 5, m = 5, exact = TRUE)
sum(a)
choose(10, 5)
a # There are 252 ways to pick 5 differences from the 10 differences.
# In 70/252 subsets of size 5, the pair with absolute rank 9
# has the largest absolute pair difference, choose(9-1, 5-1) = 70,
# and determines the sign.
Description

Data from the 2007-2008 US National Health and Nutrition Examination Survey. There are 679 matched pair differences in blood lead levels in ug per dl, comparing a daily smoker and a non-smoking control. A daily smoker smoked every day for the last 30 days and smoked at least 10 cigarettes per day. Nonsmokers smoked fewer than 100 cigarettes in their lives. The data are described further in Rosenbaum (2012).

Usage

data("smokerlead")

Format

A data frame with 679 observations on the following 3 variables.

lead Smoker-minus-control pair differences in blood lead levels, ug per dl.
llead Smoker-minus-control pair differences in logs of blood lead levels.
dose Number of cigarettes smoked per day for the smoker in the matched pair.

Details

Pairs were matched for age, gender, education, income and ethnicity.

Source

The data are originally from the 2007-2008 US National Health and Nutrition Examination Survey, but the current example appeared in Rosenbaum (2012).

References


Examples

data(smokerlead)
attach(smokerlead)
# Compare with Table 1 in Rosenbaum (2012).
quantile(lead,c(0,1/16,1/8,1/4,1/2,3/4,7/8,15/16,1))
quantile(dose,c(0,1/16,1/8,1/4,1/2,3/4,7/8,15/16,1))
oldpar<-par(mfrow=c(1,2))
boxplot(llead,ylab="Difference in Logs of Lead Levels",xlab="Smoker-Control")
Computes the P-value and Sensitivity Bound for Testing Twice.

Description

The function testtwice() is a convenient way to call the function tt(). Conversely, the function tt() is a less convenient but more flexible way to test twice. The function tt() requires you to build a matrix of signed ranks, but testtwice() builds that matrix for you.

The function tt() computes the P-value for testing twice from from a vector y of matched pair differences and a matrix H of ranks of the absolute values of y. In contrast, testtwice() follows your instructions and builds H according to those instructions. Alternatively, by setting do.test=FALSE, testtwice() will assist in constructing some columns of H for use by tt(). See Details.

The default is the same as u858 = TRUE and u878 = TRUE. If you do not take the default, then you must set at least two of the statistics to TRUE; otherwise, an error will result.

Usage

testtwice(y, dose = NULL, gamma = 1, u858 = FALSE, u888 = FALSE, u878 = FALSE, u868 = FALSE, u867 = FALSE, u222 = FALSE, brown = FALSE, noether = FALSE, tailored = FALSE, alternative="greater", do.test = TRUE)

Arguments

y A vector of matched pair differences.

dose If is.null(dose), then there are no doses. Otherwise, dose is a vector with length(dose)=length(y) giving nonnegative doses of treatment for the treated individual in a matched pair, where the control received dose zero. If there are doses, the ranks are multiplied by the doses. An error will result if some doses are negative.

gamma Value of the sensitivity parameter, gamma>=1, with gamma=1 for a randomization test.

u858 If u858 is TRUE, one column of H is created by the function multrnk() as multrnk(y, m1 = 5, m2 = 8, m = 8). See the documentation for multrnk and Rosenbaum (2011).

u888 If u888 is TRUE, one column of H is created by the function multrnk() as multrnk(y, m1 = 8, m2 = 8, m = 8). See the documentation for multrnk, Stephenson (1981), and Rosenbaum (2007, 2011).
If `u878` is TRUE, one column of H is created by the function `multrnk()` as `multrnk(y, m1 = 7, m2 = 8, m = 8)`. See the documentation for `multrnk` and Rosenbaum (2011).

If `u868` is TRUE, one column of H is created by the function `multrnk()` as `multrnk(y, m1 = 6, m2 = 8, m = 8)`. See the documentation for `multrnk` and Rosenbaum (2011).

If `u867` is TRUE, one column of H is created by the function `multrnk()` as `multrnk(y, m1 = 6, m2 = 7, m = 8)`. See the documentation for `multrnk` and Rosenbaum (2011).

If `u222` is TRUE, one column of H is created by the function `multrnk()` as `multrnk(y, m1 = 2, m2 = 2, m = 2)`. See the documentation for `multrnk` and Rosenbaum (2011). These ranks are nearly the same as Wilcoxon's ranks; see Stephenson (1981) and Pratt and Gibbons (1981, Section 3.5). Specifically, these ranks produce the U-statistic that is nearly identical to Wilcoxon's statistic.

If `brown` is TRUE, one column of H is created by the function `bmhranks()` as `bmhranks(y, q1=1/3, q2=2/3)`. This yields Brown (1981)'s test. See the documentation for `bmhranks()` and Rosenbaum (2012a). These ranks are one special case of the two-step ranks proposed by Markowski and Hettmansperger (1982).

If `noether` is TRUE, one column of H is created by the function `bmhranks()` as `bmhranks(y, q1=2/3, q2=2/3)`. This yields one version of Noether (1973)'s test and one version of the one-step tests proposed by Markowski and Hettmansperger (1982). See the documentation for `bmhranks` and Rosenbaum (2012a).

If `tailored` is TRUE, one column of H contains the tailored ranks in Rosenbaum (2015, Section 4.3 and Table 3). Although somewhat complex in form, these ranks have attractive design sensitivity and Bahadur efficiency compared with Noether's ranks above.

If `alternative` = "greater", then the null hypothesis of no effect is tested against an alternative of a positive effect. If `alternative` = "less", then the null hypothesis of no effect is tested against an alternative of a negative effect. For a two-sided test, do both tests, double the smaller of the two one-sided P-values, and replace values above 1 by 1.

If `do.test` = TRUE, then `testtwice` calls `tt()` to perform the test. If `do.test` = FALSE, then `testtwice` does not perform the test, but instead returns one or more columns of H. With `do.test` = FALSE, the user can build H using `cbind()` to combine several columns built by one or more calls to `testtwice`, and perhaps several other columns built by the user.

**Details**

The function `testtwice()` is a convenient way to call the function `tt()` which computes the P-value for testing twice from from a vector `y` of matched pair differences and a matrix `H` of ranks of the absolute values of `y`. The function `testtwice()` can create the matrix `H` and call `tt()`. Alternatively, `testtwice()` can create one or more columns of `H` that may be combined using `cbind()` to create `H` for use with `tt()`. The function `testtwice()` automates the construction of `H` in some common situations, but the function `tt()` gives the user total control over the construction of `H`, albeit with greater effort. Literally, one is testing twice if `H` has two columns, but `H` can have more than two columns.
With matched pair differences in an observational study, the functions `testtwice()` and `tt()` perform several signed rank tests with different ways of scoring the absolute ranks of the differences, and corrects for multiple testing using the joint limiting Normal distribution of the tests. For \( \gamma > 1 \), the functions perform a sensitivity analysis, reporting an upper bound on the one-sided P-value. The method and example are from Rosenbaum (2012b).

IMPORTANT: The default is equivalent to setting \( u858 = \text{TRUE} \) and \( u878 = \text{TRUE} \). That is, if no test statistic is selected, then \( u858 \) and \( u878 \) are selected.

Setting alternative=\"less\" has the same effect as replacing \( y \) by \(-y\) with alternative=\"greater\".

For a textbook discussion of adaptive inference by testing twice, see Rosenbaum (2020a, section 19.3). A different approach to adaptive inference in observational studies is discussed in Rosenbaum (2020b).

Use the function `senU()` in the DOS2 package if you do not wish to test twice, but do wish to do a sensitivity analysis using the U-statistic in Rosenbaum (2011), with confidence intervals and point estimates.

**Value**

If `do.test=TRUE`, then a list containing the following items is returned.

- `pval` The upper bound on the one-sided P-value from the joint test. If \( \gamma = 1 \), then this is the P-value, not an upper bound on the P-value.
- `dev` The standardized deviates from the joint test, one for each column of \( H \). The test uses the largest standardized deviate, correcting for multiple testing.
- `cr` The correlation matrix of the test statistics under the null hypothesis at the given value of \( \gamma \).

If `do.test=FALSE`, then a vector or matrix of signed ranks is returned for use in the function `tt()`. See the examples.

**Note**

Various signed rank statistics have been proposed by Brown (1981), Markowski and Hettmansperger (1982), Noether (1973), Rosenbaum (2007, 2011) and Stephenson (1981). The function `testtwice()` uses two or more of these signed ranks to perform the test. See also the documentation for `tt()`.

If \( y[i]=0 \), then the ith pair difference does not contribute to the permutation test.

**Author(s)**

Paul R. Rosenbaum

**References**


Examples

data(smokerlead)
attach(smokerlead)

testtwice(lead,gamma=3)
testtwice(llead,gamma=3)
testtwice(llead,u858=TRUE,u888=TRUE,gamma=3)
# Same calculation, done differently.
H<-testtwice(llead,u858=TRUE,u888=TRUE,do.test=FALSE)
dim(H)
rt(llead,H,gamma=3)
# The following example reproduces parts of the second
column (Brown) of Table 3 in Rosenbaum (2012).
# An example in the documentation for function tt()
does the same calculation in a different way.
brn <- testtwice(lead, brown = TRUE, do.test = FALSE)
brnd <- testtwice(lead, dose = dose, brown = TRUE, do.test = FALSE)
brnl <- testtwice(llead, brown = TRUE, do.test = FALSE)
tt(lead, cbind(brn, brnd, brnl), gamma = 3.2)

# The following example reproduces parts of the third
column (U-statistic) of Table 3 in Rosenbaum (2012).
u878 <- testtwice(lead, u878 = TRUE, do.test = FALSE)
u867 <- testtwice(lead, u867 = TRUE, do.test = FALSE)
u878L <- testtwice(llead, u878 = TRUE, do.test = FALSE)
u867L <- testtwice(llead, u867 = TRUE, do.test = FALSE)
tt(lead, cbind(u878, u867, u878L, u867L), gamma = 3.2)
tt(lead, cbind(u878, u867, u878L, u867L), gamma = 3.6)

# The following example compares noether=TRUE and tailored=TRUE.
testtwice(llead, brown = TRUE, noether = TRUE, gamma = 3.74)
testtwice(llead, brown = TRUE, tailored = TRUE, gamma = 3.74)
rm(brn, brnd, brnl, u878, u878L, u867, u867L, H)
detach(smokerlead)

---

**tt**

*Computes the P-value and Sensitivity Bound for Testing Twice From a User Supplied Matrix.*

**Description**

The function tt computes the P-value for testing twice from from a vector y of matched pair differences and a matrix H of ranks of the absolute values of y. Literally, one is testing twice if H has two columns, but H can have more than two columns.

The function testtwice() will create the matrix H and then call tt(); however, tt() allows you to create H according to your own specifications.

**Usage**

```r
tt(y, H, gamma = 1)
```

**Arguments**

- **y**  
  A vector of matched pair differences.

- **H**  
  A matrix with length(y) rows giving various ways of ranking the absolute y’s. Entries in H must be nonnegative. The matrix H must have at least two columns and at most 20. In a typical application, H might have 2 to 4 columns. If the
columns of H have names, then these names are used to label the output; so, it is helpful if H has column names.

\texttt{gamma} Value of the sensitivity parameter, \( \gamma \geq 1 \), with \( \gamma = 1 \) for a randomization test.

Details

With matched pair differences in an observational study, the function \texttt{tt()} performs several signed rank tests with different ways of scoring the absolute ranks of the differences, and corrects for multiple testing using the joint limiting Normal distribution of the tests. For \( \gamma > 1 \), the function \texttt{tt()} performs a sensitivity analysis, reporting an upper bound on the one-sided P-value. The method and example are from Rosenbaum (2012b).

Value

\texttt{pval} The upper bound on the one-sided P-value from the joint test. If \( \gamma = 1 \), then this is the P-value, not an upper bound on the P-value.

\texttt{dev} The standardized deviates from the joint test, one for each column of H. The test uses the largest standardized deviate, correcting for multiple testing.

\texttt{cr} The correlation matrix of the test statistics under the null hypothesis at the given value of \( \gamma \).

Note

Various signed rank statistics have been proposed by Brown (1981), Markowski and Hettmansperger (1982), Noether (1973), Rosenbaum (2007, 2011), and Stephenson (1981). The function \texttt{bmhranks()} generates ranks proposed by Brown (1981), Markowski and Hettmansperger (1982) and Noether (1973); see also Rosenbaum (1999, 2012a). The function \texttt{multrnk()} generates ranks proposed by Stephenson (1981), Rosenbaum (2007, 2011); see also Conover and Salsburg (1988). So, the columns of H may be built using \texttt{bmhranks()} and \texttt{multrnk()}.

If \( y[i]=0 \), then the ith pair difference does not contribute to the permutation test.

The P-value is one-sided, upper-tailed. To obtain a one sided, lower-tailed P-value, replace \( y \) by \(-y\). See the documentation for the \texttt{testtwice()} function.

Technical note: The function \texttt{tt()} calls the \texttt{pmvnorm()} function in the \texttt{mvtnorm} package. In this call, \texttt{tt()} sets the algorithm option to Miwa(steps=512). This choice of algorithm avoids the default algorithm in \texttt{pmvnorm()}, namely GenzBretz, which is a randomized algorithm, returning slightly different P-values each time it is called.

Author(s)

Paul R. Rosenbaum

References


Examples

data(smokerlead)
attach(smokerlead)

# The following example reproduces parts of the first column of Table 3 in Rosenbaum (2012).
w<-rank(abs(lead))
wd<-rank(abs(lead))*dose
wl<-rank(abs(llead))
H<-cbind(w,wd, wl)
tt(lead,H,gamma=2.8)
tt(lead,H,gamma=3.2)

# The following example reproduces parts of the second column of Table 3 in Rosenbaum (2012).
brn<-bmhranks(lead)
brnd<-brn*dose
brnl<-bmhranks(llead)
H2<-cbind(brn,brnd,brnl)
tt(lead,H2,gamma=3.2)
tt(lead,H2,gamma=3.4)

rm(w,wd, wl, H, brn, brnd, brnl, H2)
detach(smokerlead)
Index

* Adaptive inference
testtwice, 9
testtwice-package, 2
* Causal inference
testtwice, 9
testtwice-package, 2
tt, 13
* Design sensitivity
testtwice, 9
* Family-wise error rate
testtwice, 9
tt, 13
* Matched pairs
  smokerlead, 8
testtwice, 9
testtwice-package, 2
tt, 13
* Observational study
  smokerlead, 8
testtwice, 9
testtwice-package, 2
tt, 13
* Sensitivity analysis
testtwice, 9
testtwice-package, 2
tt, 13
* Signed rank test
  bmhranks, 3
  multrnk, 5
* Testing twice
testtwice, 9
testtwice-package, 2
tt, 13
* datasets
  smokerlead, 8
* htest
  bmhranks, 3
  multrnk, 5
testtwice, 9