

Package ‘nestedRanksTest’

August 29, 2016

Type Package

Title Mann-Whitney-Wilcoxon Test for Nested Ranks

Version 0.2

Maintainer Douglas G. Scofield <douglasgscofield@gmail.com>

Description Calculate a Mann-Whitney-Wilcoxon test for a difference between treatment levels using nested ranks. This test can be used when observations are structured into several groups and each group has received both treatment levels. The p-value is determined via bootstrapping. The nested ranks test is intended to be one possible mixed-model extension of the Mann-Whitney-Wilcoxon test, for which treatment is a fixed effect and group membership is a random effect.

ByteCompile yes

URL <https://github.com/douglasgscofield/nestedRanksTest>

License LGPL-3 | file LICENSE

Depends R (>= 3.2.0)

Suggests knitr (>= 0.5), testthat

VignetteBuilder knitr

NeedsCompilation no

Author Douglas G. Scofield [aut, cre]

Repository CRAN

Date/Publication 2015-06-06 14:40:26

R topics documented:

nestedRanksTest-package	2
nestedRanksTest	2
nestedRanksTest_weights	5
nestedRanksTest_Z	5
plot.htest_boot	6
print.htest_boot	8
woodpecker_multiyear	9

Index	11
--------------	-----------

nestedRanksTest-package

Mann-Whitney-Wilcoxon ranks test when data are in groups.

Description

Calculate a Mann-Whitney-Wilcoxon test for a difference between treatment levels using nested ranks. This test can be used when observations are structured into several groups and each group has received both treatment levels. The p-value is determined via bootstrapping. This test is intended to be analogous to a mixed-model extension of the `wilcox.test`, for which treatment is a fixed effect and group membership is a random effect.

Details

The main function is `nestedRanksTest`, which includes a formula interface implementing the familiar "`|`" syntax for specifying group membership on the right-hand side of the formula. The value returned is a list of class `'htest_boot'`, which extends class `'htest'`. `print` and `plot` methods are provided to print and visualise results.

These statistical tools were developed in collaboration with Peter E. Smouse (Rutgers University) and Victoria L. Sork (UCLA) and were funded in part by U.S. National Science Foundation awards NSF-DEB-0514956 and NSF-DEB-0516529.

References

Thompson, P. G., Smouse, P. E., Scofield, D. G. and Sork, V. L. (2014) What seeds tell us about birds: a multi-year analysis of acorn woodpecker foraging movements. *Movement Ecology* 2:12. <http://www.movementecologyjournal.com/content/2/1/12>
<https://github.com/douglasgscfield/nestedRanksTest>

nestedRanksTest

Mann-Whitney-Wilcoxon ranks test when data are in groups.

Description

The statistic for the nested ranks test is a Z-score calculated by comparing ranks between treatment levels, with contributions of each group to the final Z-score weighted by group size. The p-value is determined by comparing the observed Z-score against a distribution of Z-scores calculated by bootstrapping ranks assuming no influence of treatment while respecting group sizes. When there is just one group, this test is essentially identical to a standard Mann-Whitney-Wilcoxon test. This test is intended to be a mixed-model extension of the `wilcox.test`, for which treatment is a fixed effect and group membership is a random effect.

Usage

```
## S3 method for class 'formula'
nestedRanksTest(formula, data, groups = NULL, subset, ...)

## Default S3 method:
nestedRanksTest(x, y, groups, n.iter = 10000,
  lightweight = FALSE, ...)
```

Arguments

formula	A formula of the form <code>lhs ~ rhs</code> or <code>lhs ~ rhs groups</code> , where <code>lhs</code> is a numeric variable giving the data values, <code>rhs</code> is a variable obeying conditions for <code>x</code> , and <code>groups</code> is a variable obeying conditions for groups. If " <code> groups</code> " is not included in the formula, group membership must be specified with the <code>groups</code> argument.
data	An optional matrix or data frame (or similar: see <code>model.frame</code>) containing the variables in the formula <code>formula</code> . By default the variables are taken from <code>environment(formula)</code> .
groups	A (non-empty) vector specifying group membership for each <code>y</code> , coerced to a factor. There must be at least one <code>y</code> in each group for each treatment level.
subset	An optional vector specifying a subset of observations to be used.
...	Further arguments to be passed to or from methods.
x	A (non-empty) vector of treatments for each <code>y</code> , coerced to factor. Must contain exactly two levels.
y	A (non-empty) numeric vector of data values.
n.iter	Number of bootstrap iterations to perform. The value of the final iteration is provided by the observed Z-score. Using <code>n.iter = 1</code> simply returns the observed Z-score.
lightweight	If TRUE, the vector of individual values of the null distribution is excluded from the return value of class <code>'hctest_boot'</code> . By default the null distribution is included. If <code>n.iter</code> is large, specifying TRUE for this option can save space, but note that calling <code>plot</code> on the return value will produce an error if so.

Value

A list with class `'hctest_boot'` based on class `'hctest'` containing the following components. Components marked with "*" are additions to `'hctest'`.

statistic	the value of the observed Z-score.
p.value	the p-value for the test.
alternative	a character string describing the alternative hypothesis.
method	a character string indicating the nested ranks test performed.
data.name	a character string giving the name(s) of the data.
bad.obs	the number of observations in the data excluded because of NA values.
null.values	quantiles of the null distribution used for calculating the p-value.
n.iter*	the number of bootstrap iterations used for generating the null distribution.

`weights*` the weights for groups, calculated by `nestedRanksTest_weights`.
`null.distribution*` null distribution of Z-scores, with statistic the last value.

The length of `null.distribution` equals `n.iter`. Note that `null.distribution` will not be present if the `lightweight = TRUE` option was given to `nestedRanksTest`.

Note

Cases for which any of `x`, `y` or groups are NA are removed.

The generation of a null distribution can take some time. For example, if any use of `nestedRanksTest` in the examples were run with the default `n.iter = 10000`, completion would require a few seconds.

References

Thompson, P. G., Smouse, P. E., Scofield, D. G. and Sork, V. L. (2014) What seeds tell us about birds: a multi-year analysis of acorn woodpecker foraging movements. *Movement Ecology* 2:12.
<http://www.movementecologyjournal.com/content/2/1/12>
<https://github.com/douglasgscfield/nestedRanksTest>

See Also

[wilcox.test](#), [print.htest_boot](#), [plot.htest_boot](#)

Examples

```
require(graphics)

data(woodpecker_multiyear)

## S3 method for class 'formula'

## n.iter set to 1000 to shorten completion time

## group in formula
nestedRanksTest(Distance ~ Year | Granary, n.iter = 1000,
                data = woodpecker_multiyear,
                subset = Species == "agrifolia")
## group in 'groups='
nestedRanksTest(Distance ~ Year, groups = Granary, n.iter = 1000,
                data = woodpecker_multiyear,
                subset = Species == "lobata")

## Default S3 method

dat.a <- subset(woodpecker_multiyear, Species == "agrifolia")
## arguments in default order
nestedRanksTest(dat.a$Year, dat.a$Distance, dat.a$Granary, n.iter = 1000)
## named arguments used in 'formula' order
```

```
res <- with(subset(woodpecker_multiyear, Species == "lobata"),
            nestedRanksTest(y = Distance, x = Year, groups = Granary,
                           n.iter = 1000))
plot(res)
```

`nestedRanksTest_weights`

Calculates weights for nestedRanksTest based on group sizes.

Description

`nestedRanksTest_weights` is used by `nestedRanksTest` to calculate group weights based on group sizes. The number of group members in each of the two treatment levels is determined (n_1 and n_2) together with their product ($n_1 \cdot n_2$), and the group-specific weight is calculated by dividing $n_1 \cdot n_2$ by the sum of $n_1 \cdot n_2$ for all groups.

Usage

```
nestedRanksTest_weights(x, groups)
```

Arguments

<code>x</code>	Treatments, coerced to factor. Must contain two levels.
<code>groups</code>	Groups, coerced to factor, with elements in the same order as for <code>x</code> .

Value

data.frame containing weights and other information for each group: columns `group`, a factor of group names, also used for row names; n_1 , n_2 , and $n_1 \cdot n_2$ for integer group sizes in the first and second treatment levels and their product; and numeric `weights` for the calculated weights.

See Also

[nestedRanksTest](#)

`nestedRanksTest_Z`

Calculates Z-score from ranks.

Description

`nestedRanksTest_Z` is used by `nestedRanksTest` to calculate the Z-score for the ranks of responses `y` divided into two treatment levels.

Usage

```
nestedRanksTest_Z(y, n1, n2)
```

Arguments

y	Values to be ranked for the test. Its length must be equal to the sum of n1 and n2.
n1	The first n1 values in y belong to the first treatment level.
n2	The final n2 values in y belong to the second treatment level.

Details

Values across both treatments are ranked using the base R function `rank` with `ties.method = "average"`, which assigns tied values their average rank. The Mann-Whitney-Wilcoxon test statistic is computed from these ranks. Because the value of the statistic is sample-size dependent (between $-n1*n2$ and $n1*n2$), it is scaled to be $[-1, +1]$ by dividing by $n1*n2$.

The bottleneck for bootstrapping is calculation of ranks, so the most straightforward way to speed up `nestedRanksTest` would come from speeding up `rank`. Because of the checks performed prior to calling this routine, it should be sufficient to use a stripped-down function that simply does the equivalent of making an `.Internal` call, which is not allowed within package code. As of this writing, this is sufficient:

```
rank_new <- function (x) .Internal(rank(x, length(x), "average"))
```

For the example data this is 8-9 times faster than the base R `rank`, because it avoids error-checking overhead. For longer vectors, the advantage decreases such that at 10000 elements it is 20-30%.

Value

The calculated Z-score

See Also

[nestedRanksTest](#), [wilcox.test](#)

plot.htest_boot

Diagnostic plot of result held in htest_boot object

Description

`plot.htest_boot` creates a diagnostic plot of a return value held in a list of class `'htest_boot'`, as returned by e.g., [nestedRanksTest](#). The plot contains a histogram of the null distribution generated by bootstrapping plotted with [hist](#), and a vertical line indicating the observed value plotted with [abline](#).

Usage

```
## S3 method for class 'htest_boot'
plot(x, breaks, col = "lightblue", border = NA,
     digits = getOption("digits"), main = paste(sep = "", x$method, ", ",
     x$data.name, "\n", names(x$statistic), " = ", format(signif(x$statistic,
     max(1L, digits - 2L))), ", P = ", format.pval(x$p.value, digits = max(1L,
     digits - 3L))), xlab = "Distribution of Z-scores", ylab = paste(sep = "",
     "Frequency (out of ", x$n.iter, ")"), p.col = "red", p.lty = 2,
     p.lwd = 2, ...)
```

Arguments

x	Value of class 'htest_boot'
breaks	The number of breaks to use when plotting the distribution, the default is calculated from n.iter of the call to nestedRanksTest
col	Fill color for histogram bars, passed to hist.
border	Border color for histogram bars, passed to hist
digits	Number of digits to use for statistic and p-value, the default is taken from the "digits" option
main	Main title, passed to hist
xlab	X-axis label, passed to hist
ylab	Y-axis label, passed to hist
p.col	Observed value line colour, passed to abline
p.lty	Observed value line type, passed to abline
p.lwd	Observed value line width, passed to abline
...	Additional arguments passed to hist and abline for plotting

Details

If there is no null distribution included in the class, e.g., because the options `lightweight = TRUE` or `n.iter = 1` were given to `nestedRanksTest`, this function produces an error.

Value

None

See Also

[print.htest_boot](#) for printing test results, [hist](#) and [abline](#) for plotting options, and [nestedRanksTest](#) for one test that returns an object of class 'htest_boot'

Examples

```

require(graphics)

data(woodpecker_multiyear)

## n.iter set to 1000 to shorten completion time
res.a <- nestedRanksTest(Distance ~ Year | Granary, n.iter = 1000,
                        data = woodpecker_multiyear,
                        subset = Species == "agrifolia")
res.l <- nestedRanksTest(Distance ~ Year | Granary, n.iter = 1000,
                        data = woodpecker_multiyear,
                        subset = Species == "lobata")

opa = par(mfrow = c(2, 1))
## Defaults
plot(res.l)
## Modify colours, line type and main title
plot(res.a, main = "Quercus agrifolia", col = "lightgreen",
      p.col = "brown4", p.lty = 1)
par(opa)

```

```
print.htest_boot      Print result held in htest_boot object
```

Description

print.htest_boot prints a return value held in a list of class 'htest_boot', as returned by e.g., [nestedRanksTest](#). Class 'htest_boot' extends class 'htest' by including group weights, the number of bootstrap iterations, and the complete null distribution. The latter is not printed by this function; it may be visualised with [plot.htest_boot](#).

Usage

```
## S3 method for class 'htest_boot'
print(x, digits = getOption("digits"), prefix = "\t",
      ...)
```

Arguments

x	Value of class 'htest_boot' as returned by e.g., nestedRanksTest .
digits	Number of digits or significant digits to use in output. Similar to other print methods, this method pays attention to the "digits" option.
prefix	String, passed to print.htest
...	Additional arguments passed to print.htest .

Value

The value of x is returned invisibly.

See Also

[plot.htest_boot](#) for a graphical plot of test results, [print.htest](#) for the print method of the base class, and [nestedRanksTest](#) for one test that returns an object of class 'htest_boot'

Examples

```
data(woodpecker_multiyear)
## n.iter set to 1000 to shorten completion time
res <- nestedRanksTest(Distance ~ Year | Granary, n.iter = 1000,
                      data = woodpecker_multiyear,
                      subset = Species == "agrifolia")

class(res)
print(res)
```

woodpecker_multiyear *Distances acorns of two oak species were carried by acorn woodpeckers in two different years.*

Description

A dataset containing distances acorns of two oak species were carried by acorn woodpeckers (*Melanerpes formicivorus*) to their granaries, in two different years for each oak species. Data were collected in oak savanna habitat in central California. Acorn woodpeckers store acorns in central granaries, and different woodpecker social groups maintain different granaries. The variables are as follows:

- Species, the species of oak for the observed acorn ("lobata" for *Quercus lobata*, "agrifolia" for *Quercus agrifolia*)
- Year, the year of observation (2002 and 2004 for *Quercus lobata*, 2006 and 2007 for *Quercus agrifolia*)
- Granary, the woodpecker granary from which the acorn was collected
- Distance, distance in metres from the acorn source tree to the granary

Format

Data frame with 534 rows and 4 variables.

Author(s)

Douglas G. Scofield <douglasgscofield@gmail.com>

Source

Dataset originates from the lab of Victoria L. Sork <vlsork@ucla.edu> and is used with permission.

References

Thompson, P. G., Smouse, P. E., Scofield, D. G. and Sork, V. L. (2014) What seeds tell us about birds: a multi-year analysis of acorn woodpecker foraging movements. *Movement Ecology* 2:12.
<http://www.movementecologyjournal.com/content/2/1/12>

Index

- *Topic **datasets**
 - woodpecker_multiyear, 9
- *Topic **hplot**
 - plot.htest_boot, 6
- *Topic **htest**
 - nestedRanksTest, 2
- *Topic **nonparametric**
 - nestedRanksTest, 2

abline, 6, 7

hist, 6, 7

model.frame, 3

nestedRanksTest, 2, 2, 5–9

nestedRanksTest-package, 2

nestedRanksTest_weights, 5

nestedRanksTest_Z, 5

plot.htest_boot, 4, 6, 8, 9

print.htest, 8, 9

print.htest_boot, 4, 7, 8

wilcox.test, 2, 4, 6

woodpecker_multiyear, 9