Package 'testsuite'

June 2, 2023

Title Right tests in rigth order

Version 0.0.1.0000

Description The testsuite package (try to) organizes frequentist
tests to check the validaty of hypothesis (homoscedasticity, normality,
or whatever) in the right order, and execute the tests to compare means,
or linear models.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports agricolae,
broom,
crayon,
foreach,
normtest,
lawstat,
lmtest,
WRS2

Suggests dplyr,
parallel,
testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

`catego` *Compute pairwise tests and categorize groups from the results of the pairwise function*

Description

Compute pairwise tests and categorize groups from the results of the pairwise function

Usage

```
catego(x, ...)

## S3 method for class 'numeric'
catego(
  x,
  g,
  test_fnct = "t.test",
  p.adjust.method = p.adjust.methods,
  conf.level = 0.95,
  ...
)

## S3 method for class 'formula'
catego(
  formula,
  test_fnct = "t.test",
  p.adjust.method = p.adjust.methods,
  conf.level = 0.95,
  data,
  ...
)
```

Arguments

`x`	the values : a vector of values, off the samples to be compared, of a formula.
`...`	other parameters to give to `pairwise.table()`.

g	the groups : the vector of factors, same length than parameter x
test_fnct	the fonction used to calculate p-value. Actually 't.test', 'wilcox.test', 'yuen','var.test' are supported.
p.adjust.method	the method to correct the p-value. See p.ajust.methods to see the whole list.
formula	the formula giving x~g where x are the quantitatives measures and g is a factor of the same length.
data	the dataframe where x and g are columns names.

Value

A list :

- a dataframe $cat with a named category for each group,
- an object of class pairwise.htest. If a group are between two category, and are name with the two letters.

Author(s)

Julien Bousquet (2021)

Examples

```
#catego(x=iris$Sepal.Length, g=iris$Species, p.adjust.method='bonferroni')
```

---

cleanFactor *Clean factors*

Description

cleanFactor() check that the factor given in argument allows the application of statistical test.

Usage

```
cleanFactor(x, g)
```

Arguments

x	a vector of values, corresponding to g.
g	a vector of categories, which must be checked.

Value

Give comprehensive error messages. Return a list with x and g, cleaned if possible.

Author(s)

Julien Bousquet (2023)

Examples

```
cleanFactor( 1:6, factor(rep(c('a', 'b', NA), times=2) ))
```

`device.print` *Plot data.frame and titles*

Description

To help print graphics in m.test()

Usage

```
device.print(x, y, info = NULL, title = NULL)
```

Arguments

`x`	abscisse position of print
`y`	ordinate position of print
`info`	the information to print, a data.frame or a character string
`title`	a character string title printed in bald font. If only title is given, print in bald and italic

Author(s)

Julien Bousquet (2021)

`ind.median` *Indexes around median.*

Description

Compute index of a proportion of values of group around the median

Usage

```
ind.median(group, prop)
```

Arguments

`group`	a vector with a calculable median
`prop`	a proportion between 0 and 1 of values to find around the median. The median is always returned among the indexes, and the proportion is symetricaly define around it.

Details

ind.median() is used in wilcoxon.cut.test() when prop>0. The value of the median is always excluded.

Value

The function return the vector of indexes of prop values around the median of the group

Author(s)

Julien Bousquet (2021)

`intro` *Write an intro text*

Description

Write an intro text

Usage

```
intro(x)
```

Arguments

x	a string to write

Author(s)

Julien Bousquet (2023)

`lm.compare` *Compare two linear models*

Description

Execute the tests of `lmtest` package to compare two linear model, fitted with `lm()`.

Usage

```
lm.compare(
  reg1,
  reg2,
  nested = c(TRUE, FALSE),
  code = TRUE,
  verbose = TRUE,
  conf.level = 0.95
)
```

Arguments

reg1	,reg2 two models to compare.

`lm.test` *Execute the tests of* `lmtest` *package to validate a fit of linear model with* `lm()`.

Description

Execute the tests of `lmtest` package to validate a fit of linear model with `lm()`.

Usage

```
lm.test(reg, order.by = NULL, code = TRUE, verbose = TRUE, conf.level = 0.95)
```

Arguments

reg	the result of a linear model, obtained with `lm()`.
order.by	a vector z. The observations in the model are ordered by the size of z. If set to NULL (the default), the observations are assumed to be ordered (e.g., a time series).
code	boolean. If TRUE, write the code to replay, in the response.
verbose	boolean. If TRUE, write a short remainder comments about the interpretation of each test.

`lms.to.table` *Make a table to summary coefficients of a list of linear models*

Description

Make a table to summary coefficients of a list of linear models

Usage

```
lms.to.table(L)
```

Arguments

x	vector of names of the columns to test in data, or dataframe of the vectors to test, or a name of data column

Details

The different linear models must be computed separetely, and then pass to the argument in a list format, using function list().

Value

a dataframe with all the terms of the different linear models and their coefficient. The minus sign - is set for absence of the term in the model

Author(s)

Julien Bousquet (2021)

Examples

```
# Simple example
```

`mean.compare` *Automatic mean/median comparison*

Description

`mean.compare()` do all the possible tests of comparison of mean, even not mathematically corrects and even if hypothesis are wrong. Execute tests to verify hypothesis. The results are summarised in a graph. The user need to choose the good path, which is be done automatically by `mean.compare()`. By default, find the path the more to the left.

Usage

```
## S3 method for class 'compare'
mean(
  x,
  g,
  pval = 0.05,
  verbose = FALSE,
  return = TRUE,
  paired = FALSE,
  pval_ks = 0.01,
  maxcat = 15,
  plot = TRUE,
  silent = TRUE,
  boot = FALSE,
  iter = 100,
  conf = 0.95,
  code = FALSE,
  debug. = FALSE
)
```

Arguments

x	the values : a vector of values, with the samples to be compared.
g	groups : a vector of less (than maxcat) factors
pval	the usual level on confidence, 0.05 by default.
verbose	FALSE by defaul. Increase information.
return	to be completed
paired	FALSE by default, but can be passed to TRUE if each value of x in categories are paired and in the same order.
pval_ks	The p-value for repartitions tests like Kolmogorov-Smirnov or Shapiro : 0.01 by default.
maxcat	The maximum number of categories : 50 by default.
plot	TRUE to plot the graph with all the results.
silent	TRUE to keep the function silent : no warning
iter	100 by default, number of iterations in the bootstrap.
conf	Confidence of the bootstrap, 0.95 by default.
code	if TRUE, return the code that makes the different tests. FALSE by default
bootstrap	FALSE by default, dont do bootstrap. TRUE to make iter sampling

Author(s)

Julien Bousquet (2021)

Examples

```
#mean.compare(x=iris$Petal.Length, g=iris$Species)
```

---

pairwise *Pairwise comparisons tests*

Description

`pairwise()` compute p-value for all the possible combinaison of groups. The p-values are corrected by the `pairwise.table()` function of package stats.

Usage

```
pairwise(x, test_fnct = "t.test", p.adjust.method = p.adjust.methods, ...)

## S3 method for class 'numeric'
pairwise(x, g, test_fnct = "t.test", p.adjust.method = p.adjust.methods, ...)

## S3 method for class 'formula'
pairwise(
```

```
  formula,
  test_fnct = "t.test",
  p.adjust.method = p.adjust.methods,
  data,
  ...
)
```

Arguments

x	the values : a vector of values, off the samples to be compared, of a formula.
test_fnct	the fonction used to calculate p-value. Actually 't.test', 'wilcox.test', 'yuen','var.test' are supported.
p.adjust.method	the method to correct the p-value. See p.ajust.methods to see the whole list.
...	other parameters to give to pairwise.table().
g	the groups : the vector of factors, same length than parameter x
formula	the formula giving x~g where x are the quantitatives measures and g is a factor of the same length.
data	the dataframe where x and g are columns names.

Value

A list of class "pairwise.htest". Find the table of p-values in field $p.value.

Author(s)

Julien Bousquet (2021)

Examples

```
# Classical Student test :
#pairwise(x=iris$Sepal.Length, g=iris$Species, test_fnct='t.test')

# Robust Yuen's test of WRS2 package :
#pairwise(x=iris$Sepal.Length,g=iris$Species, test_fnct='yuen')

# Also compare variance :
#pairwise(x=iris$Sepal.Length,g=iris$Species, test_fnct='var.test')
```

part *Write a part title*

Description

Write a part title

Usage

```
part(x)
```

Arguments

x	a string to write

Author(s)

Julien Bousquet (2023)

star *A function that transforms pvalues in stars ***

Description

A function that transforms pvalues in stars ***

Usage

```
star(p.val, probs = c(0.001, 0.01, 0.05, 0.1))
```

Arguments

p.val	numeric, the value of the pvalue.
probs	a vector of the four probabilities of changes in increasing order.

Value

A character string with stars, dot or blank space.

Author(s)

Julien Bousquet (2021)

Examples

```
star(0.015)
#for a vector of p-values
pvals <- c(0.01, 0.05, 0.015, 0.5)
sapply(pvals, star)
```

---

star.lm *Get stars for a linear model*

Description

Get stars for a linear model

Usage

```
## S3 method for class 'lm'
star(p.val, ...)
```

Arguments

p.val	a linear model

Details

Use the function glance from package broom to get the p.value of the linear model, and then convert it into stars with function star().

---

warns *Write a warning about the result of a test*

Description

Use color to emphasis the result of the test.

Usage

```
warns(name, signe = c("<", ">"), pval, conf.level = 0.95)
```

Arguments

name	the name of the test.
signe	one character '>' or '<' giving the expected
pval	a numeric, between 0 and 1 : the resulting p-value of a test.
conf.level	level of confidence for decision. position of pval. expected way for the p-value to be

Author(s)

Julien Bousquet (2023)

`wilcoxon.cut.test` *Compare two halves of a population by Wilcoxon test.*

Description

Test the global efficiency of an index (group) on a mesure. This function is use in B. Jamet – (2022) to show the efficiency of different banks informations product by ONG on reals financial information (see examples).

Usage

```
wilcoxon.cut.test(
  x,
  group,
  data = NULL,
  prop = 0,
  verbose = FALSE,
  bootstrap = 0,
  seed = NULL
)
```

Arguments

`x`	vector of names of the columns to test in data, or dataframe of the vectors to test, or a name of data column
`group`	a vector of values, or the name or the number of the column variable in data to use to cut x : a vector with 2 values or a quantitative vector, with same length than x.
`data`	data.frame with the data.
`prop`	proportion of values excluded around the median of group. Make sense only if group is numerical, not factorial.
`verbose`	if TRUE, always print the wilcox.test(), if verbose=F print the result only if p-value < 0.05
`bootstrap`	0 if no bootstrap, number of bootstrap, only pertient on qualitative groups, not on a 2-values group vector.
`seed`	NULL, an integer random seed for reproductibility of bootstrap.

Details

`wilcoxon.cut.test()` take the `group` vector and cut it in two halves around the median. This group vector may be numerical or factorial with 2 levels. A proportion prop= of data around the median may be possibly excluded. The two remaining samples of data in `x`, associated for lower an upper parts of the group are passed to the test of Wilcoxon to compare their median. This allow us to validate the efficiency of the index `group` on the measured `x` tendency.

Value

a dataframe with wilcoxon p-values, or with confidence intervall of p-values, if bootstraped. Note that a p value is always positive. Here, the minus sign in front of a negative p-value means that the index (in group) is in the opposite order with x.

Author(s)

Julien Bousquet (2021)

Examples

```
# Call with names of columns of dataframe
#wilcoxon.cut.test(x=c('Sepal.Width', 'Petal.Length'), group='Sepal.Length', data=iris)
```

Index