Package: matrixTests (via r-universe)

July 2, 2024

Title Fast Statistical Hypothesis Tests on Rows and Columns of Matrices

Version 0.2.3

Maintainer Karolis Koncevičius <karolis.koncevicius@gmail.com>

Description Functions to perform fast statistical hypothesis tests on rows/columns of matrices. The main goals are: 1) speed via vectorization, 2) output that is detailed and easy to use, 3) compatibility with tests implemented in R (like those available in the 'stats' package).

Depends R (>= 3.2.2)

Imports matrixStats

Suggests PMCMRplus, car, cosinor, cosinor2, moments, nortest

License GPL-2

Encoding UTF-8

LazyData true

URL https://github.com/karoliskoncevicius/matrixTests

BugReports https://github.com/karoliskoncevicius/matrixTests/issues

RoxygenNote 7.2.3

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

RemoteUrl https://github.com/karoliskoncevicius/matrixTests

RemoteRef HEAD

RemoteSha 2159ebd5622ec5a4b026569f84d746590547a64e

Contents

`andersondarling` *Anderson-Darling test*

Description

Performs Anderson-Darling goodness of fit test for normality.

Usage

```
row_andersondarling(x)

col_andersondarling(x)
```

Arguments

`x`	numeric matrix.

Details

`row_andersondarling(x)` - Anderson-Darling test on rows. `col_andersondarling(x)` - Anderson-Darling test on columns.

Results should be the same as running `nortest::ad.test(x)` on every row (or column) of `x`

Value

a data.frame where each row contains the results of Anderson-Darling test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs - number of observations
2. statistic - test statistic
3. pvalue - p-value

Author(s)

Karolis Koncevičius

See Also

`shapiro.test()`

Examples

```
col_andersondarling(iris[,1:4])
row_andersondarling(t(iris[,1:4]))
```

---

bartlett *Bartlett test*

Description

Performs the Bartlett's test of homogeneity of variances on each row/column of the input matrix.

Usage

```
row_bartlett(x, g)

col_bartlett(x, g)
```

Arguments

x	numeric matrix.
g	a vector specifying group membership for each observation of x.

Details

NA values are always ommited. If values are missing for a whole group - that group is discarded. Groups with only one observation are also discarded.

`row_bartlett(x, g)` - Bartlet's test on rows. `col_bartlett(x, g)` - Bartlet's test on columns.

Results should be the same as as running `bartlett.test(x, g)` on every row (or column) of `x`.

Value

a data.frame where each row contains the results of the bartlett test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs.tot - total number of observations
2. obs.groups - number of groups
3. var.pooled - pooled variance estimate
4. df - degrees of freedom
5. statistic - chi-squared statistic
6. pvalue - p-value

Author(s)

Karolis Koncevičius

See Also

`bartlett.test()`

Examples

```
col_bartlett(iris[,1:4], iris$Species)
row_bartlett(t(iris[,1:4]), iris$Species)
```

---

cortest *Correlation*

Description

Performs a correlation test on each row/column of a the input matrix.

Usage

```
row_cor_pearson(x, y, alternative = "two.sided", conf.level = 0.95)

col_cor_pearson(x, y, alternative = "two.sided", conf.level = 0.95)
```

Arguments

`x`	numeric matrix.
`y`	numeric matrix for the second group of observations.
`alternative`	alternative hypothesis to use for each row/column of x. A single string or a vector with value for each observation. Must be one of "two.sided" (default), "greater" or "less".
`conf.level`	confidence levels used for the confidence intervals. A single number or a numeric vector with value for each observation. All values must be in the range of [0;1] or NA.

Details

Functions to perform various correlation tests for rows/columns of matrices. Main arguments and results were intentionally matched to the `cor.test()` function from default stats package.

`row_cor_pearson(x, y)` - test for Pearson correlation on rows. `col_cor_pearson(x, y)` - test for Pearson correlation on columns.

Results should be the same as running `cor.test(x, y, method="pearson")` on every row (or column) of `x` and `y`.

Value

a data.frame where each row contains the results of a correlation test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs.paired - number of paired observations (present in x and y)
2. cor - estimated correlation coefficient
3. df - degrees of freedom
4. statistic - t statistic
5. pvalue - p-value
6. conf.low - lower confidence interval
7. conf.high - higher confidence interval
8. alternative - chosen alternative hypothesis
9. cor.null - correlation of the null hypothesis (=0)
10. conf.level - chosen confidence level

Note

For a marked increase in computation speed turn off the calculation of confidence interval by setting `conf.level` to NA.

Author(s)

Karolis Koncevičius

See Also

`cor.test()`

Examples

```
X <- iris[iris$Species=="setosa",1:4]
Y <- iris[iris$Species=="virginica",1:4]
col_cor_pearson(X, Y)
row_cor_pearson(t(X), t(Y))
```

---

cosinor *Cosinor*

Description

Performs a Cosinor test for periodicity on each row/column of the input matrix.

Usage

```
row_cosinor(x, t, period = 24)

col_cosinor(x, t, period = 24)
```

Arguments

x	numeric matrix.
t	a vector specifying time variable for each observation of x.
period	oscillation period in the units of t (default = 24, suitable when inspecting diurnal rhythms with hourly data).

Details

row_cosinor - cosinor test on rows. col_cosinor - cosinor test on columns.

Value

a data.frame where each row contains the results of a cosinor test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs - total number of observations
2. mesor - "Midline Estimating Statistic Of Rhythm" - the average value around which the variable oscillates
3. amplitude - difference between mesor and the peak of the rhythm
4. acrophase - time when rhythm reaches its peak
5. rsquared - R-squared
6. df.model - model terms degrees of freedom
7. df.residual - residual degrees of freedom
8. statistic - F statistic for the omnibus test against intercept-only model
9. pvalue - p-value
10. period - the period used within the model

Author(s)

Karolis Koncevičius

See Also

cosinor.lm

Examples

```
wave <- sin(2*pi*1:24/24) + rnorm(24)
row_cosinor(wave, 1:24, 24)
```

`fligner` *Fligner-Killeen test*

Description

Performs the Fligner-Killeen test of homogeneity of variances (with median centering of the groups) on each row/column of the input matrix.

Usage

```
row_flignerkilleen(x, g)

col_flignerkilleen(x, g)
```

Arguments

x	numeric matrix.
g	a vector specifying group membership for each observation of x.

Details

NA values are always ommited. If values are missing for a whole group - that group is discarded. Groups with only one observation are also discarded.

`row_flignerkilleen(x, g)` - Fligner-Killeen test on rows. `col_flignerkilleen(x, g)` - Fligner-Killeen test on columns.

Results should be the same as as running `fligner.test(x, g)` on every row (or column) of `x`.

Value

a data.frame where each row contains the results of the Fligner-Killeen test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs.tot - total number of observations
2. obs.groups - number of groups
3. df - degrees of freedom
4. statistic - squared statistic
5. pvalue - p-value

Author(s)

Karolis Koncevičius

See Also

`fligner.test()`

Examples

```
col_flignerkilleen(iris[,1:4], iris$Species)
row_flignerkilleen(t(iris[,1:4]), iris$Species)
```

`fvar` *F Variance test*

Description

Performs the F test of equality of variances for two normal populations on each row/column of the two input matrices.

Usage

```
row_f_var(x, y, null = 1, alternative = "two.sided", conf.level = 0.95)

col_f_var(x, y, null = 1, alternative = "two.sided", conf.level = 0.95)
```

Arguments

`x`	numeric matrix.
`y`	numeric matrix for the second group of observations.
`null`	- hypothesized 'x' and 'y' variance ratio. A single number or numeric vector with values for each observation.
`alternative`	alternative hypothesis to use for each row/column of x. A single string or a vector with values for each observation. Values must be one of "two.sided" (default), "greater" or "less".
`conf.level`	confidence levels used for the confidence intervals. A single number or a numeric vector with values for each observation. All values must be in the range of [0:1] or NA.

Details

NA values are always ommited.

`row_f_var(x, y)` - F-test for variance on rows. `col_f_var(x, y)` - F-test for variance on columns.

Results should be the same as as running `var.test(x, y)` on every row (or column) of `x` and `y`.

Value

a data.frame where each row contains the results of the F variance test performed on the corresponding row/column of x and y.

Each row contains the following information (in order):
1. obs.x - number of x observations
2. obs.y - number of y observations

3. obs.tot - total number of observations
4. var.x - variance of x
5. var.y - variance of y
6. var.ratio - x/y variance ratio
7. df.num - numerator degrees of freedom
8. df.denom - denominator degrees of freedom
9. statistic - F statistic
10 pvalue - p-value
11. conf.low - lower bound of the confidence interval
12. conf.high - higher bound of the confidence interval
13. ratio.null - variance ratio of the null hypothesis
14. alternative - chosen alternative hypothesis
15. conf.level - chosen confidence level

Note

For a marked increase in computation speed turn off the calculation of confidence interval by setting `conf.level` to NA.

Author(s)

Karolis Koncevičius

See Also

`var.test()`

Examples

```
X <- iris[iris$Species=="setosa",1:4]
Y <- iris[iris$Species=="virginica",1:4]
col_f_var(X, Y)
```

---

jarquebera *Jarque-Bera test*

Description

Performs Jarque-Bera goodness of fit test for normality.

Usage

```
row_jarquebera(x)

col_jarquebera(x)
```

Arguments

x	numeric matrix.

Details

`row_jarquebera(x)` - Jarque-Bera test on rows. `col_jarquebera(x)` - Jarque-Bera test on columns.

Results should be the same as running `moments::jarque.test(x)` on every row (or column) of `x`

Value

a data.frame where each row contains the results of Jarque-Bera test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs - number of observations
2. skewness - skewness
3. kurtosis - kurtosis
4. df - degrees of freedom
5. statistic - chi-squared statistic
6. pvalue - p-value

Author(s)

Karolis Koncevičius

See Also

`shapiro.test()`

Examples

```
col_jarquebera(iris[,1:4])
row_jarquebera(t(iris[,1:4]))
```

---

kolmogorov *Kolmogorov-Smirnov test*

Description

Performs a Kolmogorov-Smirnov test on each row/column of the input matrix.

Usage

```
row_kolmogorovsmirnov_twosample(x, y, alternative = "two.sided", exact = NA)

col_kolmogorovsmirnov_twosample(x, y, alternative = "two.sided", exact = NA)
```

Arguments

x	numeric matrix.
y	numeric matrix for the second group of observations.
alternative	alternative hypothesis to use for each row/column of x. A single string or a vector with values for each observation. Values must be one of "two.sided" (default), "greater" or "less".
exact	logical or NA (default) indicator whether an exact p-value should be computed (see Details). A single value or a logical vector with values for each observation.

Details

Function to perform two sample Kolmogorov-Smirnov test on rows/columns of matrices. Main arguments and results were intentionally matched to the `ks.test()` function from default stats package.

Results should be the same as running `ks.test(x, y)` on every row (or column) of `x` and `y`.

By default if 'exact' argument is set to 'NA', exact p-values are computed if the product of 'x' and 'y' sample sizes is less than 10000. Otherwise, asymptotic distributions are used.

Alternative hypothesis setting specifies null and alternative hypotheses. The possible values of 'two sided', 'less', and 'greater'. 'two sided' sets the null hypothesis for the distributions of 'x' being equal to the distribution 'y'. 'less' sets the null hypothesis for the distribution of x not being less than the distribution of y. 'greater' sets the null hypothesis for the distribution of x not being greater than the distribution of y. See `help(ks.test)` for more details.

Value

a data.frame where each row contains the results of a Kolmogorov-Smirnov test performed on the corresponding row/column of x and y. Each row contains the following information (in order):
1. obs.x - number of x observations
2. obs.y - number of y observations
3. obs.tot - total number of observations
5. statistic - Wilcoxon test statistic
6. pvalue - p-value
8. alternative - chosen alternative hypothesis
9. exact - indicates if exact p-value was computed

Author(s)

Karolis Koncevičius

See Also

`ks.test()`

Examples

```
X <- iris[iris$Species=="setosa", 1:4]
Y <- iris[iris$Species=="virginica", 1:4]
col_kolmogorovsmirnov_twosample(X, Y)

# same column using different alternative hypotheses
col_kolmogorovsmirnov_twosample(X[,c(1,1,1)], Y[,c(1,1,1)], alternative=c("t", "g", "l"))
```

---

kruskalwallis *Kruskal-Wallis rank sum test*

Description

Performs a Kruskal-Wallis rank sum test on each row/column of the input matrix.

Usage

```
row_kruskalwallis(x, g)

col_kruskalwallis(x, g)
```

Arguments

x	numeric matrix.
g	a vector specifying group membership for each observation of x.

Details

`row_kruskalwallis(x, g)` - Kruskal Wallis test on rows. `col_kruskalwallis(x, g)` - Kruskal Wallis test on columns.

Results should be the same as running `kruskal.test(x, g)` on every row (or column) of `x`

Value

a data.frame where each row contains the results of a Kruskal-Wallis test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs.tot - total number of observations
2. obs.groups - number of groups
4. df - degrees of freedom
5. statistic - chi-squared statistic
6. pvalue - p.value

Author(s)

Karolis Koncevičius

See Also

`kruskal.test()`

Examples

```
col_kruskalwallis(iris[,1:4], iris$Species)
row_kruskalwallis(t(iris[,1:4]), iris$Species)
```

`levene` *Levene test*

Description

Levene's test and Brown-Forsythe test for equality of variances between groups on each row/column of the input matrix.

Usage

```
row_levene(x, g)

col_levene(x, g)

row_brownforsythe(x, g)

col_brownforsythe(x, g)
```

Arguments

`x`	numeric matrix.
`g`	a vector specifying group membership for each observation of x.

Details

NA values are always ommited. If values are missing for a whole group - that group is discarded.

`row_levene(x, g)` - Levene's test on rows. `col_levene(x, g)` - Levene's test on columns.

`row_brownforsythe(x, g)` - Brown-Forsythe test on rows. `col_brownforsythe(x, g)` - Brown-Forsythe test on columns.

Value

a data.frame where each row contains the results of the Levene's test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs.tot - total number of observations
2. obs.groups - number of groups

3. df.between - between group (treatment) degrees of freedom
4. df.within - within group (residual) degrees of freedom
5. statistic - F statistic
6. pvalue - p.value

Note

Difference between Levene's test and Brown-Forsythe test is that the Brown-Forsythe test uses the median instead of the mean in computing the spread within each group. Many software implementations use the name "Levene's test" for both variants.

Author(s)

Karolis Koncevičius

See Also

leveneTest

Examples

```
col_levene(iris[,1:4], iris$Species)
row_brownforsythe(t(iris[,1:4]), iris$Species)
```

---

oneway *Oneway ANOVA*

Description

Performs an analysis of variance tests on each row/column of the input matrix.

Usage

```
row_oneway_equalvar(x, g)

col_oneway_equalvar(x, g)

row_oneway_welch(x, g)

col_oneway_welch(x, g)
```

Arguments

x	numeric matrix.
g	a vector specifying group membership for each observation of x.

Details

Functions to perform ONEWAY ANOVA analysis for rows/columns of matrices.

`row_oneway_equalvar(x, g)` - oneway ANOVA on rows. `col_oneway_equalvar(x, g)` - oneway ANOVA on columns.

Results should be the same as running `aov(x ~ g)` on every row (or column) of `x`

`row_oneway_welch(x, g)` - oneway ANOVA with Welch correction on rows. `col_oneway_welch(x, g)` - oneway ANOVA with Welch correction on columns.

Results should be the same as running `oneway.test(x, g, var.equal=FALSE)` on every row (or column) of `x`

Value

a data.frame where each row contains the results of an oneway anova test performed on the corresponding row/column of x. The columns will vary depending on the type of test performed.

They will contain a subset of the following information:
1. obs.tot - total number of observations
2. obs.groups - number of groups
3. sumsq.between - between group (treatment) sum of squares
4. sumsq.within - within group (residual) sum of squares
5. meansq.between - between group mean squares
6. meansq.within - within group mean squares
7. df.between - between group (treatment) degrees of freedom
8. df.within - within group (residual) degrees of freedom
9. statistic - F statistic
10. pvalue - p.value

Author(s)

Karolis Koncevičius

See Also

`aov()`, `oneway.test()`

Examples

```
col_oneway_welch(iris[,1:4], iris$Species)
row_oneway_equalvar(t(iris[,1:4]), iris$Species)
```

ttest *t-test*

Description

Performs a t-test on each row/column of the input matrix.

Usage

```
row_t_equalvar(x, y, null = 0, alternative = "two.sided", conf.level = 0.95)

col_t_equalvar(x, y, null = 0, alternative = "two.sided", conf.level = 0.95)

row_t_welch(x, y, null = 0, alternative = "two.sided", conf.level = 0.95)

col_t_welch(x, y, null = 0, alternative = "two.sided", conf.level = 0.95)

row_t_onesample(x, null = 0, alternative = "two.sided", conf.level = 0.95)

col_t_onesample(x, null = 0, alternative = "two.sided", conf.level = 0.95)

row_t_paired(x, y, null = 0, alternative = "two.sided", conf.level = 0.95)

col_t_paired(x, y, null = 0, alternative = "two.sided", conf.level = 0.95)
```

Arguments

x	numeric matrix.
y	numeric matrix for the second group of observations.
null	true values of the means for the null hypothesis. A single number or numeric vector with values for each observation.
alternative	alternative hypothesis to use for each row/column of x. A single string or a vector with values for each observation. Values must be one of "two.sided" (default), "greater" or "less".
conf.level	confidence levels used for the confidence intervals. A single number or a numeric vector with values for each observation. All values must be in the range of [0:1] or NA.

Details

Functions to perform one sample and two sample t-tests for rows/columns of matrices. Main arguments and results were intentionally matched to the `t.test()` function from default stats package. Other arguments were split into separate functions:

`row_t_onesample(x)` - one sample t-test on rows. `col_t_onesample(x)` - one sample t-test on columns.

Results should be the same as running `t.test(x)` on every row (or column) of `x`.

`row_t_equalvar(x, y)` - two sample equal variance t-test on rows. `col_t_equalvar(x, y)` - two sample equal variance t-test on columns.

Results should be the same as running `t.test(x, y, var.equal=TRUE)` on every row (or column) of `x` and `y`.

`row_t_welch(x, y)` - two sample t-test with Welch correction on rows. `col_t_welch(x, y)` - two sample t-test with Welch correction on columns.

Results should be the same as running `t.test(x, y)` on every row (or column) of `x` and `y`.

`row_t_paired(x, y)` - two sample paired t-test on rows. `col_t_paired(x, y)` - two sample paired t-test on columns.

Results should be the same as running `t.test(x, y, paired=TRUE)` on every row (or column) of `x` and `y`.

Value

a data.frame where each row contains the results of a t.test performed on the corresponding row/column of x. The columns will vary depending on the type of test performed.

They will contain a subset of the following information:
1. obs.x - number of x observations
2. obs.y - number of y observations
3. obs.tot - total number of observations
4. obs.paired - number of paired observations (present in x and y)
5. mean.x - mean estiamte of x
6. mean.y - mean estiamte of y
7. mean.diff - mean estiamte of x-y difference
8. var.x - variance estiamte of x
9. var.y - variance estiamte of y
10. var.diff - variance estiamte of x-y difference
11. var.pooled - pooled variance estimate of x and y
12. stderr - standard error
13. df - degrees of freedom
14. statistic - t statistic
15. pvalue - p-value
16. conf.low - lower bound of the confidence interval
17. conf.high - higher bound of the confidence interval
18. mean.null - mean of the null hypothesis
19. alternative - chosen alternative hypothesis
20. conf.level - chosen confidence level

Note

For a marked increase in computation speed turn off the calculation of confidence interval by setting `conf.level` to NA.

Author(s)

Karolis Koncevičius

See Also

`t.test()`

Examples

```
X <- iris[iris$Species=="setosa",1:4]
Y <- iris[iris$Species=="virginica",1:4]
col_t_welch(X, Y)

# same row using different confidence levels
col_t_equalvar(X[,c(1,1,1)], Y[,c(1,1,1)], conf.level=c(0.9, 0.95, 0.99))
```

---

waerden *Van der Waerden test*

Description

Performs van der Waerden test on each row/column of the input matrix.

Usage

```
row_waerden(x, g)

col_waerden(x, g)
```

Arguments

x	numeric matrix.
g	a vector specifying group membership for each observation of x.

Details

`row_waerden(x, g)` - van der Waerden test on rows. `col_waerden(x, g)` - van det Waerden test on columns.

Value

a data.frame where each row contains the results of van det Waerden test performed on the corresponding row/column of x.

Each row contains the following information (in order):
1. obs.tot - total number of observations
2. obs.groups - number of groups
3. df - degrees of freedome
4. statistic - van det Waerden chi-squared statistic
5. pvalue - p.value

Author(s)

Karolis Koncevičius

See Also

vanWaerdenTest, row_oneway_equalvar, row_kruskalwallis

Examples

```
col_waerden(iris[,1:4], iris$Species)
row_waerden(t(iris[,1:4]), iris$Species)
```

---

wilcoxon *Wilcoxon test*

Description

Performs a Wilcoxon test on each row/column of the input matrix.

Usage

```
row_wilcoxon_twosample(
  x,
  y,
  null = 0,
  alternative = "two.sided",
  exact = NA,
  correct = TRUE
)

col_wilcoxon_twosample(
  x,
  y,
  null = 0,
  alternative = "two.sided",
  exact = NA,
  correct = TRUE
)

row_wilcoxon_onesample(
  x,
  null = 0,
  alternative = "two.sided",
  exact = NA,
  correct = TRUE
)
```

```
col_wilcoxon_onesample(
  x,
  null = 0,
  alternative = "two.sided",
  exact = NA,
  correct = TRUE
)

row_wilcoxon_paired(
  x,
  y,
  null = 0,
  alternative = "two.sided",
  exact = NA,
  correct = TRUE
)

col_wilcoxon_paired(
  x,
  y,
  null = 0,
  alternative = "two.sided",
  exact = NA,
  correct = TRUE
)
```

Arguments

x	numeric matrix.
y	numeric matrix for the second group of observations.
null	true values of the location shift for the null hypothesis. A single number or numeric vector with values for each observation.
alternative	alternative hypothesis to use for each row/column of x. A single string or a vector with values for each observation. Values must be one of "two.sided" (default), "greater" or "less".
exact	logical or NA (default) indicator whether an exact p-value should be computed (see Details). A single value or a logical vector with values for each observation.
correct	logical indicator whether continuity correction should be applied in the cases where p-values are obtained using normal approximation. A single value or logical vector with values for each observation.

Details

Functions to perform one sample and two sample Wilcoxon tests on rows/columns of matrices. Main arguments and results were intentionally matched to the `wilcox.test()` function from default stats package. Other arguments were split into separate functions:

`row_wilcoxon_onesample(x)` - one sample Wilcoxon test on rows. `col_wilcoxon_onesample(x)` - one sample Wilcoxon test on columns.

Results should be the same as running `wilcox.test(x)` on every row (or column) of `x`.

`row_wilcoxon_twosample(x, y)` - two sample Wilcoxon test on rows. `col_wilcoxon_twosample(x, y)` - two sample Wilcoxon test on columns.

Results should be the same as running `wilcox.test(x, y)` on every row (or column) of `x` and `y`.

`row_wilcoxon_paired(x, y)` - two sample paired Wilcoxon test on rows. `col_wilcoxon_paired(x, y)` - two sample paired Wilcoxon test on columns.

Results should be the same as running `wilcox.test(x, y, paired=TRUE)` on every row (or column) of `x` and `y`.

By default if 'exact' argument is set to 'NA', exact p-values are computed only if both 'x' and 'y' contain less than 50 values and there are no ties. Single sample and paired tests have additional requirement of not having zeroe values (values equal to null hypothesis location argument 'mu'). Otherwise, a normal approximation is used. Be wary of using 'exact=TRUE' on large sample sizes as computations can take a very long time.

'correct' argument controls the continuity correction of p-values but only when exact p-values cannot be computed and normal approximation is used. For cases where exact p-values are returned 'correct' is switched to FALSE.

Value

a data.frame where each row contains the results of a wilcoxon test performed on the corresponding row/column of x. The columns will vary depending on the type of test performed.

They will contain a subset of the following information:
1. obs.x - number of x observations
2. obs.y - number of y observations
3. obs.tot - total number of observations
4. obs.paired - number of paired observations (present in x and y)
5. statistic - Wilcoxon test statistic
6. pvalue - p-value
7. location.null - location shift of the null hypothesis
8. alternative - chosen alternative hypothesis
9. exact - indicates if exact p-value was computed
10. correct - indicates if continuity correction was performed

Note

Confidence interval and pseudo-median calculations are not implemented.

Author(s)

Karolis Koncevičius

See Also

`wilcox.test()`

Examples

```
X <- iris[iris$Species=="setosa",1:4]
Y <- iris[iris$Species=="virginica",1:4]
col_wilcoxon_twosample(X, Y)

# same row using different alternative hypotheses
col_wilcoxon_twosample(X[,c(1,1,1)], Y[,c(1,1,1)], alternative=c("t", "g", "l"))
```

Index