

Package ‘autotestR’

August 21, 2026

Title Automated Functions for Basic Statistical Tests

Version 1.2.16

Description Provides simple and intuitive functions for basic statistical analyses.

Methods include the t-test (Student 1908 <[doi:10.1093/biomet/6.1.1](https://doi.org/10.1093/biomet/6.1.1)>),
the Mann-Whitney U test (Mann and Whitney 1947 <[doi:10.1214/aoms/1177730491](https://doi.org/10.1214/aoms/1177730491)>),
Pearson's correlation (Pearson 1895 <[doi:10.1098/rspl.1895.0041](https://doi.org/10.1098/rspl.1895.0041)>),
and analysis of variance (Fisher 1925, <[doi:10.1007/978-1-4612-4380-9_5](https://doi.org/10.1007/978-1-4612-4380-9_5)>).
Functions are compatible with 'ggplot2' and 'dplyr'.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

RoxygenNote 7.3.2

Imports car, crayon, dplyr, FSA, ggplot2, ggdist, ggExtra, nortest,
purrr, RColorBrewer, scales, stats, tibble, tidyr, tidyselect,
utils, vcd

NeedsCompilation no

Maintainer Luiz Garcia <luiz.cardoso@ufpr.br>

Author Luiz Garcia [aut, cre] (ORCID: <<https://orcid.org/0000-0002-9616-0927>>)

Repository CRAN

Date/Publication 2026-08-21 05:45:17 UTC

Contents

autotestR	2
pre.test	3
test.anova	3
test.chi	4
test.correlation	6
test.fisher	8
test.interaction	9
test.kruskal	11
test.t	12

test.tmulti	13
test.tpaired	15
test.u	16

Index	18
--------------	-----------

autotestR	<i>autotestR</i>
-----------	------------------

Description

designed to simplify the execution of the main statistical tests commonly used in the life sciences. It provides user-friendly functions that automatically generate plots and clear explanations, making statistical analysis more accessible for researchers and students.

Main features

- t test (independent and paired)
- Mann–Whitney test (Wilcoxon rank-sum)
- Multiple group comparison (t test or Mann–Whitney)
- Chi-squared test and Fisher’s exact test
- One-way ANOVA with Tukey HSD post hoc test
- Kruskal–Wallis test with Dunn post hoc test
- Pearson, Spearman, and Kendall correlation tests with automatic plots
- Diagnostic function that suggests the most appropriate statistical test
- Intuitive plots fully integrated into the functions

Basic usage:

```
library(autotestR)
```

Independent t test

```
group1 <- rnorm(30, 10, 2) group2 <- rnorm(30, 12, 2) test.t(group1, group2)
```

Chi-squared test

```
var1 <- sample(c("A", "B"), 100, replace = TRUE) var2 <- sample(c("Yes", "No"), 100, replace = TRUE) test.chi(var1, var2)
```

Multiple test (t test or Mann–Whitney)

```
df <- data.frame( control = rnorm(30, 10), treatment = rnorm(30, 12), test1 = rnorm(30, 11), test2 = rnorm(30, 15) ) test.tmulti(df)
```

ANOVA with post hoc test

```
g1 <- rnorm(20, 5) g2 <- rnorm(20, 7) g3 <- rnorm(20, 6) test.anova(g1, g2, g3)
```

Correlation test

```
x <- rnorm(30) y <- x + rnorm(30, 0, 1) test.correlation(x, y)
```

Author(s)

Maintainer: Luiz Garcia <luiz.cardoso@ufpr.br> ([ORCID](#))

```
pre.test
```

```
pre.test() — Automatic statistical test suggestion
```

Description

Automatically identifies whether the input data are numeric or categorical and suggests the most appropriate statistical test.

Usage

```
pre.test(..., alpha = 0.05, help = FALSE, verbose = TRUE)
```

Arguments

...	Two or more vectors (numeric or categorical), or a data.frame with ≥ 2 columns
alpha	Significance level. Default = 0.05
help	Logical. If TRUE, shows detailed help
verbose	Logical. If TRUE, prints informative messages

Value

Invisible list with normality results, homogeneity or contingency table, and test recommendation

```
test.anova
```

```
ANOVA test with automated assumption checking
```

Description

Performs ANOVA (and Tukey HSD) if data meet normality and homogeneity assumptions. Otherwise, automatically recommends Kruskal-Wallis/Dunn.

Usage

```
test.anova(
  ...,
  title = "ANOVA/Tukey HSD",
  xlab = "Group",
  ylab = "Value",
  style = c("boxplot", "violin", "mono", "halfeye"),
  help = FALSE,
  verbose = TRUE
)
```

Arguments

...	Vectors or a data.frame with ≥ 2 columns.
title	Plot title.
xlab	X-axis label.
ylab	Y-axis label.
style	Aesthetic style of the generated plot.
help	If TRUE, shows help.
verbose	If TRUE, shows detailed messages.

Value

An aov object or a recommendation message.

Examples

```
df <- data.frame(
  control = rnorm(30, 10),
  treatment = rnorm(30, 12),
  test = rnorm(30, 11)
)
test.anova(df)
```

test.chi

Chi-square test

Description

Applies the Pearson chi-square test or Fisher's exact test to assess association between two categorical variables.

Usage

```
test.chi(  
  x,  
  y = NULL,  
  title = "Chi-square Test",  
  xlab = NULL,  
  ylab = "Proportion",  
  style = c("stacked", "barplot", "mosaic", "pie"),  
  show_table = TRUE,  
  help = FALSE,  
  verbose = TRUE  
)
```

Arguments

x	Categorical vector or data frame with two columns (group 1 and group 2).
y	Categorical vector (group 2). Required if x is a vector.
title	Plot title (string). Default: "Chi-square Test".
xlab	X-axis label in the plot (string). Default: NULL (uses variable name).
ylab	Y-axis label in the plot (string). Default: "Proportion".
style	Plot style generated by the function.
show_table	Logical. If TRUE, prints the contingency table to the console. Default: TRUE.
help	Logical. If TRUE, displays a detailed explanation of the function. Default: FALSE.
verbose	Logical. If TRUE, prints messages about the test and expected frequencies. Default: TRUE.

Value

Test result and contingency table.

Examples

```
data <- data.frame(  
  control = c(rep("healthy", 50), rep("sick", 150)),  
  treatment = c(rep("healthy", 100), rep("sick", 100))  
)  
test.chi(data)
```

test.correlation

*Correlation Analysis with Automatic Method Selection***Description**

Performs correlation analysis between two numeric variables using Pearson, Spearman, or Kendall methods. When `method = "auto"`, the function automatically selects the most appropriate method based on normality, presence of ties, and proportion of outliers.

Usage

```
test.correlation(
  x,
  y = NULL,
  method = "auto",
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  style = c("simple", "inference", "structure", "density", "distribution"),
  plot_normality = FALSE,
  help = FALSE,
  verbose = TRUE
)

## S3 method for class 'test.correlation'
print(x, ...)
```

Arguments

<code>x</code>	Numeric vector or data frame with exactly two numeric columns.
<code>y</code>	Numeric vector (optional if <code>x</code> is a data frame).
<code>method</code>	Correlation method: "auto" (default), "pearson", "spearman", or "kendall".
<code>main</code>	Plot title. If NULL, an automatic title is generated. If FALSE, no title is displayed.
<code>xlab</code>	Label for the x-axis. If NULL, a label is inferred from the input. If FALSE, no label is shown.
<code>ylab</code>	Label for the y-axis. If NULL, a label is inferred from the input. If FALSE, no label is shown.
<code>style</code>	Plot style: "simple" (default), "inference", "structure", "density", or "distribution".
<code>plot_normality</code>	Logical. If TRUE, displays QQ-plots for normality assessment.
<code>help</code>	Logical. If TRUE, displays a detailed usage guide.
<code>verbose</code>	Logical. If TRUE, prints information about method selection and diagnostics.
<code>...</code>	Additional arguments passed to other print methods (currently ignored)

Details

In addition to hypothesis testing, the function provides diagnostic information, effect size interpretation, and publication-ready visualizations.

Method selection in automatic mode follows these rules:

- If more than 5\ Kendall's tau is used.
- If both variables are normally distributed and no ties are present, Pearson's correlation is used.
- Otherwise, Spearman's rank correlation is applied.

Normality is assessed using Shapiro-Wilk ($n \leq 50$), Anderson-Darling ($50 < n \leq 300$), or Kolmogorov-Smirnov ($n > 300$) tests.

Value

An object of class "test.correlation" containing:

- call: Matched function call.
- data: Input data (if $n \leq 5000$).
- n: Sample size.
- method: Selected correlation method.
- estimate: Correlation coefficient.
- effect: Effect size information (r, rho, or tau, and r^2 when applicable).
- conf.int: Confidence interval.
- p.value: P-value of the test.
- interpretation: Qualitative interpretation of effect size.
- diagnostics: Normality, ties, and outlier diagnostics.
- decision: Automatic selection rationale.
- htest: Underlying stats::cor.test object.
- plot: ggplot2 object.

See Also

[cor.test](#), [ggplot](#)

Examples

```
# Pearson (approximately normal data)
set.seed(123)
x <- rnorm(50, sd = 0.1)
y <- x + rnorm(50, sd = 0.1)
test.correlation(x, y)

# Spearman (non-normal data)
set.seed(123)
x <- runif(300)
y <- log(x + 0.1) + rnorm(300, sd = 0.5)
```

```
test.correlation(x, y)

# Kendall (ties and outliers)
set.seed(123)
x <- runif(1000, 1, 100)
y <- sin(x) * 30 + rnorm(1000, 0, 10)
x[sample(1:500, 50)] <- 50
y[sample(1:500, 50)] <- 0
x_out <- runif(100, 10, 20)
y_out <- runif(100, 80, 120)
x <- c(x, x_out)
y <- c(y, y_out)
test.correlation(x, y)
```

test.fisher	<i>Fisher's Exact Test</i>
-------------	----------------------------

Description

Performs Fisher's Exact Test using two categorical vectors or a data frame with two columns, constructing a contingency table and optionally generating graphical visualizations.

Usage

```
test.fisher(
  x,
  y = NULL,
  title = "Fisher's Exact Test",
  xlab = NULL,
  ylab = "Proportion",
  style = c("stacked", "barplot", "mosaic", "pie"),
  show_table = TRUE,
  help = FALSE,
  verbose = TRUE
)
```

Arguments

x	Categorical vector or data frame with two columns.
y	Categorical vector (required if x_var is a vector).
title	Plot title (string). Default: "Fisher's Exact Test"
xlab	Name of the x-axis in the plot (string). Default: NULL (uses variable name)
ylab	Name of the y-axis in the plot (string). Default: "Proportion"
style	Plot style. Controls the visualization type
show_table	Logical. If TRUE, prints the contingency table to the console. Default: TRUE
help	Logical. If TRUE, shows detailed function explanation. Default: FALSE
verbose	Logical. If TRUE, prints detailed test messages. Default: TRUE

Value

Invisible object containing the Fisher test result.

Examples

```
data <- data.frame(control = c("healthy", "healthy", "sick", "sick", "sick"),
                   treatment = c("healthy", "healthy", "healthy", "healthy", "sick"))
test.fisher(data)
```

test.interaction	<i>Test for interaction between a continuous predictor and a grouping variable</i>
------------------	--

Description

Fits a linear model of the form $y \sim x * by$ to evaluate whether the association between a continuous predictor and an outcome differs across groups. Optionally produces a publication-ready visualization of group-specific regression lines.

Usage

```
test.interaction(
  x,
  y,
  by,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  plot = TRUE,
  style = c("clean", "CI", "facet"),
  conf.level = 0.95,
  verbose = TRUE,
  help = FALSE
)

## S3 method for class 'test.interaction'
print(x, ...)
```

Arguments

x	Numeric vector representing the continuous predictor.
y	Numeric vector representing the continuous outcome.
by	Grouping variable defining the interaction. Must be coercible to a factor with at least two levels.
title	Optional title title for the plot.
xlab	Optional x-axis label.

<code>ylab</code>	Optional y-axis label.
<code>plot</code>	Logical. Should a plot be generated?
<code>style</code>	Plot style. One of "clean", "CI" or "facet".
<code>conf.level</code>	Confidence level for the interaction interval (default: 0.95).
<code>verbose</code>	Logical. If TRUE, prints detailed messages. Default: TRUE.
<code>help</code>	Logical. If TRUE, shows a detailed explanation of the function. Default: FALSE.
<code>...</code>	Additional arguments passed to other print methods (currently ignored)

Details

The interaction coefficient (β) represents the difference in regression slopes between groups, conditional on the reference level of `by`. The sign and magnitude of this coefficient depend on the chosen reference group.

Confidence intervals are emphasized as the primary inferential quantity.

Value

An object of class "test.interaction" containing:

- `model`: the fitted linear model,
- `interaction`: estimated interaction effects with confidence intervals,
- `plot`: a ggplot object (if `plot = TRUE`).

Examples

```
# Simple example: different trends between groups

set.seed(123)

n <- 60

marker <- rnorm(n, 10, 2)
group <- rep(c("Control", "Treatment"), each = n/2)

# Same intercept, different slopes
response <- 2 +
  ifelse(group == "Control", 0.5, 1.2) * marker +
  rnorm(n, 0, 1)

test.interaction(marker, response, group)
```

test.kruskal	<i>Kruskal-Wallis Test with Dunn Post-hoc</i>
--------------	---

Description

Performs the Kruskal-Wallis rank-sum test for comparing three or more independent groups, followed by Dunn's post-hoc test with multiple comparison adjustment.

Usage

```
test.kruskal(
  ...,
  title = "Kruskal-Wallis + Dunn",
  xlab = "Group",
  ylab = "Value",
  style = c("boxplot", "violin", "mono", "halfeye"),
  adjust = c("bonferroni", "holm", "BH"),
  help = FALSE,
  verbose = TRUE
)
```

Arguments

...	Numeric vectors representing groups, or a data frame with two or more columns (each column is treated as a group).
title	Character. Plot title.
xlab	Character. X-axis label.
ylab	Character. Y-axis label.
style	Character. Plot style. One of: "boxplot", "violin", "mono", or "halfeye".
adjust	Character. Method for p-value adjustment in Dunn's test. One of "bonferroni", "holm", or "BH".
help	Logical. If TRUE, displays a short help message and exits.
verbose	Logical. If TRUE, prints formatted statistical results to the console.

Details

This function is a non-parametric alternative to one-way ANOVA and is recommended when normality or homoscedasticity assumptions are violated.

Value

Invisibly returns a list with the following components:

type Test type.

H Kruskal-Wallis H statistic.

df Degrees of freedom.
p Global test p-value.
epsilon_sq Epsilon-squared effect size.
epsilon_ci Bootstrap confidence interval for effect size.
means_sd Group means and standard deviations.
dunn Dunn post-hoc results.
significant_pairs Significant pairwise comparisons.
data Long-format data used in the analysis.

Examples

```
set.seed(123)

n <- 25

df <- data.frame(
  control    = rexp(n, rate = 1),
  treatment1 = rexp(n, rate = 0.6),
  treatment2 = rgamma(n, shape = 2, scale = 1)
)

test.kruskal(df)
```

test.t

Student's t-test (with automatic diagnostics)

Description

Performs Student's t-test to compare means between two independent groups, with automatic checks for normality and homogeneity of variances. If assumptions are violated, the Mann-Whitney test is automatically applied (without generating a plot).

Usage

```
test.t(
  ...,
  title = TRUE,
  title_text = "t-test",
  xlab = "Group",
  ylab = "Value",
  style = c("boxplot", "violin", "mono", "halfeye"),
  help = FALSE,
  verbose = TRUE
)
```

Arguments

<code>...</code>	Two numeric vectors or a data frame with exactly two columns.
<code>title</code>	Logical. If true, return a plot entitled.
<code>title_text</code>	Plot title (string). Default: "t-test".
<code>xlab</code>	X-axis label in the plot (string). Default: "Group".
<code>ylab</code>	Y-axis label in the plot (string). Default: "Value".
<code>style</code>	Plot aesthetic generated by the function.
<code>help</code>	Logical. If TRUE, shows a detailed explanation of the function. Default: FALSE.
<code>verbose</code>	Logical. If TRUE, prints detailed messages. Default: TRUE.

Value

Invisible list with summary, test `test_result`, method and (optionally) plot.

Examples

```
set.seed(123)
df <- data.frame(
  control = rnorm(30, 10),
  treatment = rnorm(30, 15)
)
test.t(df)
```

test.tmulti

Multiple pairwise comparisons with diagnostics

Description

Performs multiple pairwise comparisons using Student's t-test or Mann-Whitney test, with automatic diagnostics, effect sizes, confidence intervals, multiple testing correction and visualization.

Usage

```
test.tmulti(
  ...,
  comparisons = NULL,
  title = "Multiple comparisons (t / MW)",
  xlab = "",
  ylab = "Value",
  style = c("boxplot", "violin", "mono", "halfeye"),
  p_adjust = c("none", "holm", "BH", "bonferroni"),
  help = FALSE,
  verbose = TRUE
)
```

Arguments

<code>...</code>	Numeric vectors or a data.frame with groups in columns.
<code>comparisons</code>	List of character vectors specifying pairwise comparisons (e.g. <code>list(c("A","B"), c("B","C"))</code>). If NULL, all pairwise combinations are used.
<code>title</code>	Plot title.
<code>xlab</code>	X-axis label.
<code>ylab</code>	Y-axis label.
<code>style</code>	Plot style. One of "boxplot", "violin", "mono", or "halfeye".
<code>p_adjust</code>	Method for multiple testing correction. One of "none", "holm", "BH", "bonferroni".
<code>help</code>	Logical. If TRUE, prints usage examples.
<code>verbose</code>	Logical. If TRUE, prints results and plots.

Details

Normality is assessed using Shapiro-Wilk tests and homogeneity of variances using Levene's test. If assumptions are met, a pooled-variance t-test is used. Otherwise, the Mann-Whitney test is applied with bootstrap confidence intervals.

Effect sizes:

- Cohen's d for t-tests
- Rank-biserial correlation for Mann-Whitney

Value

A list with:

results A tibble with test results.

plot A ggplot object.

data_long Long-format data used for plotting.

Examples

```
df <- data.frame(
  control = rnorm(30, 10),
  treatment = rnorm(30, 12),
  test1 = rnorm(30, 11),
  test2 = rnorm(30, 15)
)

test.tmulti(
  df,
  comparisons = list(
    c("control", "treatment"),
    c("treatment", "test1")
  )
)
```

test.tpaired	<i>Paired t-test with advanced visualizations</i>
--------------	---

Description

Performs a paired t-test between two numeric vectors (e.g., before vs after) or between two numeric columns of a data frame. Includes four visualization styles (boxplot, violin, mono, and half-eye).

Usage

```
test.tpaired(  
  ...,  
  title = "Paired t-test",  
  xlab = "",  
  ylab = "Value",  
  style = c("boxplot", "violin", "mono", "halfeye"),  
  connect = TRUE,  
  help = FALSE,  
  verbose = TRUE  
)
```

Arguments

...	Two numeric vectors of equal length, or a data frame with exactly two numeric columns.
title	Plot title.
xlab	X-axis label.
ylab	Y-axis label.
style	Plot style: • 1 Premium boxplot • 2 Violin + minimal boxplot • 3 mono • 4 Half-eye (ggdist)
connect	Logical. If TRUE, connects paired observations.
help	If TRUE, displays detailed help.
verbose	If TRUE, prints progress messages.

Value

- An invisible list containing:
- summary_table** Group means and standard deviations
 - test_result** t-test result object (stats::t.test)
 - data** Data frame used for plotting
 - plot** ggplot2 object

Examples

```
before <- c(13, 12, 15, 14)
after  <- c(9, 11, 10, 10)
test.tpaired(before, after)
```

test.u	<i>Mann-Whitney U Test</i>
--------	----------------------------

Description

Performs the Mann-Whitney (Wilcoxon rank-sum) test for comparing two independent groups, with statistical summary and graphical visualization.

Usage

```
test.u(
  ...,
  title = "Mann-Whitney Test",
  xlab = "Group",
  ylab = "Value",
  style = c("boxplot", "violin", "mono", "halfeye"),
  help = FALSE,
  verbose = TRUE
)
```

Arguments

...	Two numeric vectors or a data.frame with two numeric columns.
title	Plot title. Default: "Mann-Whitney Test".
xlab	Label for x-axis. Default: "Group".
ylab	Label for y-axis. Default: "Value".
style	Plot aesthetic style.
help	Logical. If TRUE, prints a detailed explanation. Default: FALSE.
verbose	Logical. If TRUE, prints detailed messages. Default: TRUE.

Value

Invisible list with:

- summary** Group-wise statistical summary
- test** Test result (htest object)
- plot** ggplot2 visualization object

Examples

```
x <- c(1, 3, 5, 6)
y <- c(7, 8, 9, 12)
data <- data.frame(groupA = x, groupB = y)
test.u(data)
```

Index

autotestR, [2](#)
autotestR-package (autotestR), [2](#)

cor.test, [7](#)

ggplot, [7](#)

pre.test, [3](#)
print.test.correlation
 (test.correlation), [6](#)
print.test.interaction
 (test.interaction), [9](#)

test.anova, [3](#)
test.chi, [4](#)
test.correlation, [6](#)
test.fisher, [8](#)
test.interaction, [9](#)
test.kruskal, [11](#)
test.t, [12](#)
test.tmulti, [13](#)
test.tpaired, [15](#)
test.u, [16](#)