Package 'outlying'

May 9, 2026

Title Outliers Detection

Version 0.0.1

Description Provides functions for detecting outliers in datasets using statistical methods. The package supports identification of anomalous observations in numerical data and is intended for use in data cleaning, exploratory data analysis, and preprocessing workflows.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Joon-Keat Lai [aut, cre, cph]

Maintainer Joon-Keat Lai <p10911004@gmail.com>

Repository CRAN

Date/Publication 2026-01-23 20:30:08 UTC

Contents

`Grubbs_test` *Grubbs' test*

Description

Iteratively search for all possible outliers in a numeric vector.

Usage

```
Grubbs_test(
  x,
  alpha = 0.05,
  min_n = 7L,
  iteration = -1L,
  max_out = 0.2,
  use_median = FALSE,
  sensitivity = 2,
  verbose = FALSE
)
```

Arguments

x	A numeric vector.
alpha	Default: 0.05 (two-tailed, thus 0.025 for each side).
min_n	A positive integer (default: 7). The minimum observations required for the test.
iteration	How many iterations of the test should be proceeded (default: -1; means unlimited)? Each iteration will only recognize one outlier. For example, `iteration = 3` means the test will find no more than 3 outliers.
max_out	The maximum proportion (ranged from 0 to 1) of outliers to be detected in the dataset (default: 0.2, which means the data contain no more than 20% of outliers data points). If too many outliers, simply discarding them using this approach might be inappropriate.
use_median	Use the median or the mean value as the center (default: FALSE).
sensitivity	An integer value range from 1 to 3. The higher the value, the more sensitive of the test to outliers (default: 2).
verbose	Should the output includes statistics result (default: FALSE)?

Value

By default (verbose = FALSE), return a logical named vector indicating the outlying elements. If verbose = TRUE, return a list which contains statistic values.

References

Grubbs, F. E. (1969). Procedures for Detecting Outlying Observations in Samples. Technometrics, 11(1), 1–21. https://doi.org/10.1080/00401706.1969.10490657

Examples

```
set.seed(1)
#--------------------------------------------------------------------------
Grubbs_test(c(0, 0, 7, 0, 0, 1, 0))
#>     0     0     7     0     0     1     0
#> FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
#--------------------------------------------------------------------------
```

```
x <- c(round(rnorm(3, 0, 1), 2), -5, 3)
Grubbs_test(x, min_n = 5, max_out = 0.4)
#> -0.63  0.18 -0.84    -5     3
#> FALSE FALSE FALSE  TRUE  TRUE
#-------------------------------------------------------------------------------
x <- round(c(rnorm(10, 0, 1), 5))
Grubbs_test(x)
#>     2     0    -1     0     1     1     0     2     0    -1     5
#> FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
```

Index