

# Package ‘outliers’

July 22, 2025

**Version** 0.15

**Date** 2022-03-26

**Title** Tests for Outliers

**Author** Lukasz Komsta <lukasz.komsta@umlub.pl>

**Maintainer** Lukasz Komsta <lukasz.komsta@umlub.pl>

**Depends** R (>= 2.0)

**Description** A collection of some tests commonly used for identifying outliers.

**License** GPL (>= 2)

**URL** <https://www.R-project.org>, <http://www.komsta.net/>

**Repository** CRAN

**NeedsCompilation** no

**Date/Publication** 2022-03-26 09:00:02 UTC

## Contents

chisq.out.test . . . . .	2
cochran.test . . . . .	3
dixon.test . . . . .	4
grubbs.test . . . . .	6
outlier . . . . .	7
qcochran . . . . .	8
qdixon . . . . .	9
qgrubbs . . . . .	10
qtable . . . . .	11
rm.outlier . . . . .	12
scores . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

chisq.out.test	<i>Chi-squared test for outlier</i>
----------------	-------------------------------------

---

### Description

Performs a chisquared test for detection of one outlier in a vector.

### Usage

```
chisq.out.test(x, variance=var(x), opposite = FALSE)
```

### Arguments

x	a numeric vector for data values.
variance	known variance of population. if not given, estimator from sample is taken, but there is not so much sense in such test (it is similar to z-scores)
opposite	a logical indicating whether you want to check not the value with largest difference from the mean, but opposite (lowest, if most suspicious is highest etc.)

### Details

This function performs a simple test for one outlier, based on chisquared distribution of squared differences between data and sample mean. It assumes known variance of population. It is rather not recommended today for routine use, because several more powerful tests are implemented (see other functions mentioned below). It was discussed by Dixon (1950) for the first time, as one of the tests taken into account by him.

### Value

A list with class `htest` containing the following components:

statistic	the value of chisquared-statistic.
p.value	the p-value for the test.
alternative	a character string describing the alternative hypothesis.
method	a character string indicating what type of test was performed.
data.name	name of the data argument.

### Note

This test is known to reject only extreme outliers, if no known variance is specified.

### Author(s)

Lukasz Komsta

## References

Dixon, W.J. (1950). Analysis of extreme values. Ann. Math. Stat. 21, 4, 488-506.

## See Also

[dixon.test](#), [grubbs.test](#)

## Examples

```
set.seed(1234)
x = rnorm(10)
chisq.out.test(x)
chisq.out.test(x,opposite=TRUE)
```

---

cochran.test	<i>Test for outlying or inlying variance</i>
--------------	--

---

## Description

This test is useful to check if largest variance in several groups of data is "outlying" and this group should be rejected. Alternatively, if one group has very small variance, we can test for "inlying" variance.

## Usage

```
cochran.test(object, data, inlying = FALSE)
```

## Arguments

object	A vector of variances or formula.
data	If object is a vector, data should be another vector, giving number of data in each corresponding group. If object is a formula, data should be a dataframe.
inlying	Test smallest variance instead of largest.

## Details

The corresponding p-value is calculated using `pcochran` function.

## Value

A list with class `htest` containing the following components:

statistic	the value of Cochran-statistic.
p.value	the p-value for the test.
alternative	a character string describing the alternative hypothesis.
method	a character string indicating what type of test was performed.
data.name	name of the data argument.
estimate	vector of variance estimates

**Author(s)**

Lukasz Komsta

**References**

Snedecor, G.W., Cochran, W.G. (1980). Statistical Methods (seventh edition). Iowa State University Press, Ames, Iowa.

**See Also**

[qcochran](#)

**Examples**

```
set.seed(1234)
x=rnorm(100)
d=data.frame(x=x,group=rep(1:10,10))
cochran.test(x~group,d)
cochran.test(x~group,d,inlying=TRUE)
x=runif(5)
cochran.test(x,rep(5,5))
cochran.test(x,rep(100,5))
```

---

dixon.test

*Dixon tests for outlier*


---

**Description**

Performs several variants of Dixon test for detecting outlier in data sample.

**Usage**

```
dixon.test(x, type = 0, opposite = FALSE, two.sided = TRUE)
```

**Arguments**

x	a numeric vector for data values.
opposite	a logical indicating whether you want to check not the value with largest difference from the mean, but opposite (lowest, if most suspicious is highest etc.)
type	an integer specifying the variant of test to be performed. Possible values are compliant with these given by Dixon (1950): 10, 11, 12, 20, 21. If this value is set to zero, a variant of the test is chosen according to sample size (10 for 3-7, 11 for 8-10, 21 for 11-13, 22 for 14 and more). The lowest or highest value is selected automatically, and can be reversed used opposite parameter.
two.sided	treat test as two-sided (default).

## Details

The p-value is calculating by interpolation using [qdixon](#) and [qtable](#). According to Dixon (1951) conclusions, the critical values can be obtained numerically only for  $n=3$ . Other critical values are obtained by simulations, taken from original Dixon's paper, and regarding corrections given by Rorabacher (1991).

## Value

A list with class `htest` containing the following components:

<code>statistic</code>	the value of Dixon Q-statistic.
<code>p.value</code>	the p-value for the test.
<code>alternative</code>	a character string describing the alternative hypothesis.
<code>method</code>	a character string indicating what type of test was performed.
<code>data.name</code>	name of the data argument.

## Author(s)

Lukasz Komsta

## References

- Dixon, W.J. (1950). Analysis of extreme values. *Ann. Math. Stat.* 21, 4, 488-506.
- Dixon, W.J. (1951). Ratios involving extreme values. *Ann. Math. Stat.* 22, 1, 68-78.
- Rorabacher, D.B. (1991). Statistical Treatment for Rejection of Deviant Values: Critical Values of Dixon Q Parameter and Related Subrange Ratios at the 95 percent Confidence Level. *Anal. Chem.* 83, 2, 139-146.

## See Also

[chisq.out.test](#), [grubbs.test](#)

## Examples

```
set.seed(1234)
x = rnorm(10)
dixon.test(x)
dixon.test(x,opposite=TRUE)
dixon.test(x,type=10)
```

grubbs.test

*Grubbs tests for one or two outliers in data sample***Description**

Performs Grubbs' test for one outlier, two outliers on one tail, or two outliers on opposite tails, in small sample.

**Usage**

```
grubbs.test(x, type = 10, opposite = FALSE, two.sided = FALSE)
```

**Arguments**

<code>x</code>	a numeric vector for data values.
<code>opposite</code>	a logical indicating whether you want to check not the value with largest difference from the mean, but opposite (lowest, if most suspicious is highest etc.)
<code>type</code>	Integer value indicating test variant. 10 is a test for one outlier (side is detected automatically and can be reversed by <code>opposite</code> parameter). 11 is a test for two outliers on opposite tails, 20 is test for two outliers in one tail.
<code>two.sided</code>	Logical value indicating if there is a need to treat this test as two-sided.

**Details**

The function can perform three tests given and discussed by Grubbs (1950).

First test (10) is used to detect if the sample dataset contains one outlier, statistically different than the other values. Test is based by calculating score of this outlier  $G$  (outlier minus mean and divided by  $sd$ ) and comparing it to appropriate critical values. Alternative method is calculating ratio of variances of two datasets - full dataset and dataset without outlier. The obtained value called  $U$  is bound with  $G$  by simple formula.

Second test (11) is used to check if lowest and highest value are two outliers on opposite tails of sample. It is based on calculation of ratio of range to standard deviation of the sample.

Third test (20) calculates ratio of variance of full sample and sample without two extreme observations. It is used to detect if dataset contains two outliers on the same tail.

The p-values are calculated using [qgrubbs](#) function.

**Value**

<code>statistic</code>	the value statistic. For type 10 it is difference between outlier and the mean divided by standard deviation, and for type 20 it is sample range divided by standard deviation. Additional value $U$ is ratio of sample variances with and without suspicious outlier. According to Grubbs (1950) these values for type 10 are bound by simple formula and only one of them can be used, but function gives both. For type 20 the $G$ is the same as $U$ .
<code>p.value</code>	the p-value for the test.

alternative	a character string describing the alternative hypothesis.
method	a character string indicating what type of test was performed.
data.name	name of the data argument.

**Author(s)**

Lukasz Komsta

**References**

Grubbs, F.E. (1950). Sample Criteria for testing outlying observations. Ann. Math. Stat. 21, 1, 27-58.

**See Also**

[dixon.test](#), [chisq.out.test](#)

**Examples**

```
set.seed(1234)
x = rnorm(10)
grubbs.test(x)
grubbs.test(x, type=20)
grubbs.test(x, type=11)
```

---

outlier

---

*Find value with largest difference from the mean*


---

**Description**

Finds value with largest difference between it and sample mean, which can be an outlier.

**Usage**

```
outlier(x, opposite = FALSE, logical = FALSE)
```

**Arguments**

x	a data sample, vector in most cases. If argument is a dataframe, then outlier is calculated for each column by <code>sapply</code> . The same behavior is applied by <code>apply</code> when the matrix is given.
opposite	if set to TRUE, gives opposite value (if largest value has maximum difference from the mean, it gives smallest and vice versa)
logical	if set to TRUE, gives vector of logical values, and possible outlier position is marked by TRUE

**Value**

A vector of value(s) with largest difference from the mean.

**Author(s)**

Lukasz Komsta, corrections by Markus Graube

**See Also**

[rm.outlier](#)

**Examples**

```
set.seed(1234)
y=rnorm(100)
outlier(y)
outlier(y,opposite=TRUE)
dim(y) <- c(20,5)
outlier(y)
outlier(y,opposite=TRUE)
```

---

qcochran

---

*Critical values and p-values for Cochran outlying variance test*


---

**Description**

This functions calculates quantiles (critical values) and reversively p-values for Cochran test for outlying variance.

**Usage**

```
qcochran(p, n, k)
pcochran(q, n, k)
```

**Arguments**

p	vector of probabilities.
q	vector of quantiles.
n	number of values in each group (if not equal, use arithmetic mean).
k	number of groups.

**Value**

Vector of p-values or critical values.

**Author(s)**

Lukasz Komsta



**References**

Snedecor, G.W., Cochran, W.G. (1980). Statistical Methods (seventh edition). Iowa State University Press, Ames, Iowa.

**See Also**

[cochran.test](#)

**Examples**

```
qcochran(0.05, 5, 5)
pcochran(0.293, 5, 5)
```

---

qdixon	<i>critical values and p-values for Dixon tests</i>
--------	---

---

**Description**

Approximated quantiles (critical values) and distribution function (giving p-values) for Dixon tests for outliers.

**Usage**

```
qdixon(p, n, type = 10, rev = FALSE)
pdixon(q, n, type = 10)
```

**Arguments**

p	vector of probabilities.
q	vector of quantiles.
n	length of sample.
type	integer value: 10, 11, 12, 20, or 21. For description see <a href="#">dixon.test</a> .
rev	function qdixon with this parameter set to TRUE acts as pdixon.

**Details**

This function is based on tabularized Dixon distribution, given by Dixon (1950) and corrected by Rorabacher (1991). Continuity is reached due to smart interpolation using [qtable](#) function. By now, numerical procedure to obtain these values for  $n > 3$  is not known.

**Value**

Critical value or p-value (vector).

**Author(s)**

Lukasz Komsta

## References

- Dixon, W.J. (1950). Analysis of extreme values. *Ann. Math. Stat.* 21, 4, 488-506.
- Dixon, W.J. (1951). Ratios involving extreme values. *Ann. Math. Stat.* 22, 1, 68-78.
- Rorabacher, D.B. (1991). Statistical Treatment for Rejection of Deviant Values: Critical Values of Dixon Q Parameter and Related Subrange Ratios at the 95 percent Confidence Level. *Anal. Chem.* 83, 2, 139-146.

## See Also

[qtable](#), [dixon.test](#)

---

qgrubbs	<i>Calculate critical values and p-values for Grubbs tests</i>
---------	--

---

## Description

This function is designed to calculate critical values for Grubbs tests for outliers detecting and to approximate p-values reversively.

## Usage

```
qgrubbs(p, n, type = 10, rev = FALSE)
pgrubbs(q, n, type = 10)
```

## Arguments

p	vector of probabilities.
q	vector of quantiles.
n	sample size.
type	Integer value indicating test variant. 10 is a test for one outlier (side is detected automatically and can be reversed by opposite parameter). 11 is a test for two outliers on opposite tails, 20 is test for two outliers in one tail.
rev	if set to TRUE, function qgrubbs acts as pgrubbs.

## Details

The critical values for test for one outlier is calculated according to approximations given by Pearson and Sekar (1936). The formula is simply reversed to obtain p-value.

The values for two outliers test (on opposite sides) are calculated according to David, Hartley, and Pearson (1954). Their formula cannot be rearranged to obtain p-value, thus such values are obtained by [uniroot](#).

For test checking presence of two outliers at one tail, the tabularized distribution (Grubbs, 1950) is used, and approximations of p-values are interpolated using [qtable](#).

**Value**

A vector of quantiles or p-values.

**Author(s)**

Lukasz Komsta

**References**

Grubbs, F.E. (1950). Sample Criteria for testing outlying observations. *Ann. Math. Stat.* 21, 1, 27-58.

Pearson, E.S., Sekar, C.C. (1936). The efficiency of statistical tools and a criterion for the rejection of outlying observations. *Biometrika*, 28, 3, 308-320.

David, H.A, Hartley, H.O., Pearson, E.S. (1954). The distribution of the ratio, in a single normal sample, of range to standard deviation. *Biometrika*, 41, 3, 482-493.

**See Also**

[grubbs.test](#)

---

qtable

*Interpolate tabularized distribution*

---

**Description**

This function calculates critical values or p-values which cannot be obtained numerically, and only tabularized version is available.

**Usage**

```
qtable(p, probs, quants)
```

**Arguments**

p	vector of probabilities.
probs	vector of given probabilities.
quants	vector of given corresponding quantiles.

**Details**

This function is internal routine used to obtain Grubbs and Dixon critical values. It fits linear or cubical regression to closests values of its argument, then uses obtained function to obtain quantile by interpolation.

**Value**

A vector of interpolated values

**Note**

You can simply do "reverse" interpolation (p-value calculating) by reversing probabilities and quantiles (2 and 3 argument).

**Author(s)**

Lukasz Komsta

---

rm.outlier

*Remove the value(s) most differing from the mean*

---

**Description**

If the outlier is detected and confirmed by statistical tests, this function can remove it or replace by sample mean or median.

**Usage**

```
rm.outlier(x, fill = FALSE, median = FALSE, opposite = FALSE)
```

**Arguments**

x	a dataset, most frequently a vector. If argument is a dataframe, then outlier is removed from each column by sapply. The same behavior is applied by apply when the matrix is given.
fill	If set to TRUE, the median or mean is placed instead of outlier. Otherwise, the outlier(s) is/are simply removed.
median	If set to TRUE, median is used instead of mean in outlier replacement.
opposite	if set to TRUE, gives opposite value (if largest value has maximum difference from the mean, it gives smallest and vice versa)

**Value**

A dataset of the same type as argument, with outlier(s) removed or replacement by appropriate means or medians.

**Author(s)**

Lukasz Komsta

**See Also**

[outlier](#)

**Examples**

```

set.seed(1234)
y=rnorm(100)
outlier(y)
outlier(y,opposite=TRUE)
rm.outlier(y)
rm.outlier(y,opposite=TRUE)
dim(y) <- c(20,5)
outlier(y)
outlier(y,logical=TRUE)
outlier(y,logical=TRUE,opposite=TRUE)
rm.outlier(y)
rm.outlier(y,opposite=TRUE)

```

---

scores	<i>Calculate scores of the sample</i>
--------	---------------------------------------

---

**Description**

This function calculates normal, t, chi-squared, IQR and MAD scores of given data.

**Usage**

```
scores(x, type = c("z", "t", "chisq", "iqr", "mad"), prob = NA, lim = NA)
```

**Arguments**

x	a vector of data.
type	"z" calculates normal scores (differences between each value and the mean divided by sd), "t" calculates t-Student scores (transformed by $(z \cdot \sqrt{n-2}) / \sqrt{z^2 - 1 - t^2}$ formula, "chisq" gives chi-squared scores (squares of differences between values and mean divided by variance. For the "iqr" type, all values lower than first and greater than third quartile is considered, and difference between them and nearest quartile divided by IQR are calculated. For the values between these quartiles, scores are always equal to zero. "mad" gives differences between each value and median, divided by median absolute deviation.
prob	If set, the corresponding p-values instead of scores are given. If value is set to 1, p-value are returned. Otherwise, a logical vector is formed, indicating which values are exceeding specified probability. In "z" and "mad" types, there is also possibility to set this value to zero, and then scores are confirmed to $(n-1)/\sqrt{n}$ value, according to Shiffler (1998). The "iqr" type does not support probabilities, but "lim" value can be specified.
lim	This value can be set for "iqr" type of scores, to form logical vector, which values has this limit exceeded.

**Value**

A vector of scores, probabilities, or logical vector.

**Author(s)**

Lukasz Komsta, corrections by Alan Richter

**References**

Schiffler, R.E (1998). Maximum Z scores and outliers. Am. Stat. 42, 1, 79-80.

**See Also**

[mad](#), [IQR](#), [grubbs.test](#),

**Examples**

```
set.seed(1234)
x = rnorm(10)
scores(x)
scores(x,prob=1)
scores(x,prob=0.5)
scores(x,prob=0.1)
scores(x,prob=0.93)
scores(x,type="iqr")
scores(x,type="mad")
scores(x,prob=0)
```

# Index

## \* **distribution**

qcochran, [8](#)

qgrubbs, [10](#)

qtable, [11](#)

## \* **htest**

chisq.out.test, [2](#)

cochran.test, [3](#)

dixon.test, [4](#)

grubbs.test, [6](#)

outlier, [7](#)

qdixon, [9](#)

rm.outlier, [12](#)

## \* **models**

scores, [13](#)

chisq.out.test, [2](#), [5](#), [7](#)

cochran.test, [3](#), [9](#)

dixon.test, [3](#), [4](#), [7](#), [10](#)

grubbs.test, [3](#), [5](#), [6](#), [11](#), [14](#)

IQR, [14](#)

mad, [14](#)

outlier, [7](#), [12](#)

pcochran (qcochran), [8](#)

pdixon (qdixon), [9](#)

pgrubbs (qgrubbs), [10](#)

qcochran, [4](#), [8](#)

qdixon, [5](#), [9](#)

qgrubbs, [6](#), [10](#)

qtable, [5](#), [9](#), [10](#), [11](#)

rm.outlier, [8](#), [12](#)

scores, [13](#)

uniroot, [10](#)