

# Package ‘dixonTest’

July 22, 2025

**Type** Package  
**Title** Dixon's Ratio Test for Outlier Detection  
**Version** 1.0.4  
**Date** 2022-08-22  
**Description** For outlier detection in small and normally distributed samples the ratio test of Dixon (Q-test) can be used. Density, distribution function, quantile function and random generation for Dixon's ratio statistics are provided as wrapper functions. The core applies McBane's Fortran functions <[doi:10.18637/jss.v016.i03](https://doi.org/10.18637/jss.v016.i03)> that use Gaussian quadrature for a numerical solution.  
**License** GPL-3  
**ByteCompile** yes  
**NeedsCompilation** yes  
**Encoding** UTF-8  
**Classification/MSR-2010** 62F03, 62E17, 62Q05  
**RoxigenNote** 7.2.0  
**Author** Thorsten Pohlert [aut, cre] (ORCID: <<https://orcid.org/0000-0003-3855-3025>>), George C. McBane [ctb]  
**Maintainer** Thorsten Pohlert <[thorsten.pohlert@gmx.de](mailto:thorsten.pohlert@gmx.de)>  
**Repository** CRAN  
**Date/Publication** 2022-08-22 19:40:02 UTC

## Contents

Dixon . . . . .	2
dixonTest . . . . .	3
Index	6

Dixon

*Dixon distribution***Description**

Density, distribution function, quantile function and random generation for Dixon's ratio statistics  $r_{j,i-1}$  for outlier detection.

**Usage**

```
qdixon(p, n, i = 1, j = 1, log.p = FALSE, lower.tail = TRUE)

pdixon(q, n, i = 1, j = 1, lower.tail = TRUE, log.p = FALSE)

ddixon(x, n, i = 1, j = 1, log = FALSE)

rdixon(n, i = 1, j = 1)
```

**Arguments**

p	vector of probabilities.
n	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required
i	number of observations $\leq x_i$
j	number of observations $\geq x_j$
log.p	logical; if TRUE probabilities p are given as <code>log(p)</code>
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .
q	vector of quantiles
x	vector of quantiles.
log	logical; if TRUE (default), probabilities p are given as <code>log(p)</code> .

**Details**

According to McBane (2006) the density of the statistics  $r_{j,i-1}$  of Dixon can be yield if  $x$  and  $v$  are integrated over the range  $(-\infty < x < \infty, 0 \leq v < \infty)$

$$\begin{aligned}
 f(r) = & \frac{n!}{(i-1)!(n-j-i-1)!(j-1)!} \\
 & \times \int_{-\infty}^{\infty} \int_0^{\infty} \left[ \int_{-\infty}^{x-v} \phi(t) dt \right]^{i-1} \left[ \int_{x-v}^{x-rv} \phi(t) dt \right]^{n-j-i-1} \\
 & \times \left[ \int_{x-rv}^x \phi(t) dt \right]^{j-1} \phi(x-v) \phi(x-rv) \phi(x) v \, dv \, dx
 \end{aligned}$$

where  $v$  is the Jacobian and  $\phi(\cdot)$  is the density of the standard normal distribution. McBane (2006) has proposed a numerical solution using Gaussian quadratures (Gauss-Hermite quadrature and half-range Hermite quadrature) and coded a library in Fortran. These R functions are wrapper functions to use the respective Fortran code.

**Value**

ddixon gives the density function, pdixon gives the distribution function, qdixon gives the quantile function and rdixon generates random deviates.

**Source**

The R code is a wrapper to the Fortran code released under GPL  $\geq 2$  in the electronic supplement of McBane (2006). The original files are 'rfuncs.f', 'utility.f' and 'dixonr.fi'. They were slightly modified to comply with current CRAN policy and the R manual 'Writing R Extensions'.

**Note**

The file 'slowTest/d-p-q-r-tests.R.out.save' that is included in this package contains some results for the assessment of the numerical accuracy.

The slight numerical differences between McBane's original Fortran output (see files 'slowTests/test[1,2,4].ref.output' and this implementation are related to different floating point rounding algorithms between R (see 'round to even' in [round](#)) and Fortran's write(\*, 'F6.3') statement.

**References**

Dixon, W. J. (1950) Analysis of extreme values. *Ann. Math. Stat.* **21**, 488–506. doi:[10.1214/aoms/1177729747](https://doi.org/10.1214/aoms/1177729747).

Dean, R. B., Dixon, W. J. (1951) Simplified statistics for small numbers of observation. *Anal. Chem.* **23**, 636–638. doi:[10.1021/ac60052a025](https://doi.org/10.1021/ac60052a025).

McBane, G. C. (2006) Programs to compute distribution functions and critical values for extreme value ratios for outlier detection. *J. Stat. Soft.* **16**. doi:[10.18637/jss.v016.i03](https://doi.org/10.18637/jss.v016.i03).

**Examples**

```
set.seed(123)
n <- 20
Rdixon <- rdixon(n, i = 3, j = 2)
Rdixon
pdixon(Rdixon, n = n, i = 3, j = 2)
ddixon(Rdixon, n = n, i = 3, j = 2)
```

---

dixonTest

*Dixons Outlier Test (Q-Test)*


---

**Description**

Performs Dixons single outlier test.

**Usage**

```
dixonTest(x, alternative = c("two.sided", "greater", "less"), refined = FALSE)
```

## Arguments

x	a numeric vector of data
alternative	the alternative hypothesis. Defaults to "two.sided"
refined	logical indicator, whether the refined version or the Q-test shall be performed. Defaults to FALSE

## Details

Let  $X$  denote an identically and independently distributed normal variate. Further, let the increasingly ordered realizations denote  $x_1 \leq x_2 \leq \dots \leq x_n$ . Dixon (1950) proposed the following ratio statistic to detect an outlier (two sided):

$$r_{j,i-1} = \max \left\{ \frac{x_n - x_{n-j}}{x_n - x_i}, \frac{x_{1+j} - x_1}{x_{n-i} - x_1} \right\}$$

The null hypothesis, no outlier, is tested against the alternative, at least one observation is an outlier (two sided). The subscript  $j$  on the  $r$  symbol indicates the number of outliers that are suspected at the upper end of the data set, and the subscript  $i$  indicates the number of outliers suspected at the lower end. For  $r_{10}$  it is also common to use the statistic  $Q$ .

The statistic for a single maximum outlier is:

$$r_{j,i-1} = (x_n - x_{n-j}) / (x_n - x_i)$$

The null hypothesis is tested against the alternative, the maximum observation is an outlier.

For testing a single minimum outlier, the test statistic is:

$$r_{j,i-1} = (x_{1+j} - x_1) / (x_{n-i} - x_1)$$

The null hypothesis is tested against the alternative, the minimum observation is an outlier.

Apart from the earlier Dixons Q-test (i.e.  $r_{10}$ ), a refined version that was later proposed by Dixon can be performed with this function, where the statistic  $r_{j,i-1}$  depends on the sample size as follows:

$$\begin{array}{ll} r_{10}: & 3 \leq n \leq 7 \\ r_{11}: & 8 \leq n \leq 10 \\ r_{21}: & 11 \leq n \leq 13 \\ r_{22}: & 14 \leq n \leq 30 \end{array}$$

The p-value is computed with the function `pdixon`.

## References

- Dixon, W. J. (1950) Analysis of extreme values. *Ann. Math. Stat.* **21**, 488–506. doi:10.1214/aoms/1177729747.
- Dean, R. B., Dixon, W. J. (1951) Simplified statistics for small numbers of observation. *Anal. Chem.* **23**, 636–638. doi:10.1021/ac60052a025.
- McBane, G. C. (2006) Programs to compute distribution functions and critical values for extreme value ratios for outlier detection. *J. Stat. Soft.* **16**. doi:10.18637/jss.v016.i03.

**Examples**

```
## example from Dean and Dixon 1951, Anal. Chem., 23, 636-639.  
x <- c(40.02, 40.12, 40.16, 40.18, 40.18, 40.20)  
dixonTest(x, alternative = "two.sided")  
  
## example from the dataplot manual of NIST  
x <- c(568, 570, 570, 570, 572, 578, 584, 596)  
dixonTest(x, alternative = "greater", refined = TRUE)
```

# Index

- \* **distribution**

- Dixon, [2](#)

- \* **htest**

- dixonTest, [3](#)

- \* **outliers**

- dixonTest, [3](#)

ddixon (Dixon), [2](#)

Dixon, [2](#)

dixonTest, [3](#)

pdixon, [4](#)

pdixon (Dixon), [2](#)

qdixon (Dixon), [2](#)

rdixon (Dixon), [2](#)

round, [3](#)