

# Package ‘triangulr’

November 20, 2020

**Title** High-Performance Triangular Distribution Functions

**Version** 1.1.0

**Description** A collection of high-performance functions for the triangular distribution that consists of the probability density function, cumulative distribution function, quantile function, random variate generator, moment generating function, characteristic function, and expected shortfall function. References: Samuel Kotz, Johan Ren Van Dorp (2004) <doi:10.1142/5720>, Melissa E. O'Neill (2014) <<https://www.cs.hmc.edu/tr/hmc-cs-2014-0905.pdf>>, David Blackman and Sebastiano Vigna (2018) <arXiv:1805.01407>, and Acerbi, Carlo and Tasche, Dirk. (2002) <doi:10.1111/1468-0300.00091>.

**License** MIT + file LICENSE

**URL** <https://github.com/irkaal/triangulr/>  
<https://irkaal.github.io/triangulr/>

**BugReports** <https://github.com/irkaal/triangulr/issues>

**Depends** R (>= 3.3)

**Encoding** UTF-8

**LazyData** true

**ByteCompile** true

**RoxygenNote** 7.1.1

**Collate** 'RcppExports.R' 'Triangular.R' 'ctri.R' 'dtri.R' 'error.R'  
'estri.R' 'mgtri.R' 'ptri.R' 'qtri.R' 'rtri.R'

**LinkingTo** Rcpp, dqrng

**Imports** Rcpp (>= 1.0.5), rlang (>= 0.4.8), vctrs (>= 0.3.5), dqrng (>= 0.2.1)

**Suggests** testthat (>= 3.0.0)

**NeedsCompilation** yes

**Author** Alvin Nursalim [aut, cre]

**Maintainer** Alvin Nursalim <[alvin.nursalim@outlook.com](mailto:alvin.nursalim@outlook.com)>

**Repository** CRAN

**Date/Publication** 2020-11-20 07:30:02 UTC

## R topics documented:

Triangular . . . . . 2

**Index** . . . . . 6

Triangular *The Triangular Distribution*

### Description

These functions provide information about the triangular distribution on the interval from `min` to `max` with mode equal to `mode`. `ctri` gives the characteristic function, `dtri` gives the density function, `estri` gives the expected shortfall, `mgtri` gives the moment generating function, `ptri` gives the distribution function, `qtri` gives the quantile function, and `rtri` gives the random variate generator.

### Usage

```
ctri(t, min = 0, max = 1, mode = 0.5)
dtri(x, min = 0, max = 1, mode = 0.5, log = FALSE)
estri(p, min = 0, max = 1, mode = 0.5, lower_tail = TRUE, log_p = FALSE)
mgtri(t, min = 0, max = 1, mode = 0.5)
ptri(q, min = 0, max = 1, mode = 0.5, lower_tail = TRUE, log_p = FALSE)
qtri(p, min = 0, max = 1, mode = 0.5, lower_tail = TRUE, log_p = FALSE)
rtri(n, min = 0, max = 1, mode = 0.5, dqrng = FALSE)
```

### Arguments

<code>t</code>	Vector of dummy variables.
<code>min</code>	Lower limit of the distribution. Must have $\text{min} < \text{max}$ .
<code>max</code>	Upper limit of the distribution. Must have $\text{max} > \text{min}$ .
<code>mode</code>	The mode of the distribution. Must have $\text{mode} \geq \text{min}$ and $\text{mode} \leq \text{max}$ .
<code>x, q</code>	Vector of quantiles.
<code>log, log_p</code>	Logical; if TRUE, probabilities <code>p</code> are given as $\log(p)$ .
<code>p</code>	Vector of probabilities.
<code>lower_tail</code>	Logical; if TRUE (default), probabilities <code>p</code> are $P[X \leq x]$ , otherwise, $P[X > x]$ .
<code>n</code>	Number of observations. Must have length of one.
<code>dqrng</code>	Logical; if FALSE (default), <code>runic</code> will be used to generate random numbers instead of <code>dqrunic</code> from the <code>dqrng</code> package.

**Details**

If `min`, `max`, or `mode` are not specified they assume the default values of 0, 1, and 0.5 respectively.

The triangular distribution has density

$$0$$

for  $x < \text{min}$  or  $x > \text{max}$

$$f(x) = \frac{2(x - \text{min})}{(\text{max} - \text{min})(\text{mode} - \text{min})}$$

for  $\text{min} \leq x < \text{mode}$ , and

$$f(x) = \frac{2(\text{max} - x)}{(\text{max} - \text{min})(\text{max} - \text{mode})}$$

for  $\text{mode} < x \leq \text{max}$ .

`rtri` will not generate either of the extreme values unless `max - min` is small compared to `min`, and in particular not for the default arguments.

**Value**

`ctri` gives the characteristic function, `dtri` gives the density function, `estri` gives the expected shortfall, `mgtri` gives the moment generating function, `ptri` gives the distribution function, `qtri` gives the quantile function, and `rtri` gives the random variate generator.

The numerical arguments other than `n` with values of size one are recycled to the length of `t` for `ctri` and `mgtri`, the length of `x` for `dtri`, the length of `p` for `estri` and `qtri`, the length of `q` for `ptri`, and `n` for `rtri`. This determines the length of the result.

The logical arguments `log`, `lower_tail`, and `log_p` must be of length one each.

**Note**

The characteristics of output from pseudo-random number generators (such as precision and periodicity) vary widely. See [.Random.seed](#) for more information on R's random number generation algorithms.

**See Also**

[RNG](#) about random number generation in R.

[Distributions](#) for other standard distributions.

**Examples**

```
# min, max, and mode with lengths equal to the length of x
x <- c(0, 0.5, 1)
d <- dtri(x,
  min = c(0, 0, 0),
  max = c(1, 1, 1),
  mode = c(0.5, 0.5, 0.5))
# min and max will be recycled to the length of x
```

```

rec_d <- dtri(x,
             min = 0,
             max = 1,
             mode = c(0.5, 0.5, 0.5))
all.equal(d, rec_d)

# min, max, and mode with lengths equal to the length of x
n <- 3
set.seed(1)
r <- rtri(n,
         min = c(0, 0, 0),
         max = c(1, 1, 1),
         mode = c(0.5, 0.5, 0.5))
# min and max will be recycled to the length of n
set.seed(1)
rec_r <- rtri(n,
             min = 0,
             max = 1,
             mode = c(0.5, 0.5, 0.5))
all.equal(r, rec_r)

dqrng::dqset.seed(1)
r <- rtri(
  n,
  min = 0,
  max = 1,
  mode = 0.5,
  dqrng = TRUE
)

# Log quantiles
x <- c(0, 0.5, 1)
log_d <- dtri(x, log = TRUE)
d <- dtri(x, log = FALSE)
all.equal(log(d), log_d)

# Upper tail probabilities
q <- c(0, 0.5, 1)
upper_p <- ptri(q, lower_tail = FALSE)
p <- ptri(q, lower_tail = TRUE)
all.equal(upper_p, 1 - p)

# Log probabilities
q <- c(0, 0.5, 1)
log_p <- ptri(q, log_p = TRUE)
p <- ptri(q, log_p = FALSE)
all.equal(upper_p, 1 - p)

# The quantile function
p <- c(0, 0.5, 1)
upper_q <- ptri(1 - p, lower_tail = FALSE)
q <- ptri(p, lower_tail = TRUE)
all.equal(upper_q, q)

```

```
p <- c(0, 0.5, 1)
log_q <- qtri(log(p), log_p = TRUE)
q <- qtri(p, log_p = FALSE)
all.equal(log_q, q)

# Moment generating function
t <- c(1, 2, 3)
mgtri(t)

# Characteristic function
t <- c(1, 2, 3)
ctri(t)

# Expected Shortfall
p <- c(0.1, 0.5, 1)
estri(p)
```

# Index

.Random.seed, 3  
ctri (Triangular), 2  
Distributions, 3  
dqrng, 2  
dqrnif, 2  
dtri (Triangular), 2  
estri (Triangular), 2  
mgtri (Triangular), 2  
ptri (Triangular), 2  
qtri (Triangular), 2  
RNG, 3  
rtri (Triangular), 2  
runif, 2  
Triangular, 2