

Package: triangle (via r-universe)

September 11, 2024

Title Distribution Functions and Parameter Estimates for the Triangle Distribution

Version 1.0.1

Description Provides the `r`, `q`, `p`, and `d` distribution functions for the triangle distribution. Also includes maximum likelihood estimation of parameters.

License GPL (>= 2)

URL <https://bertcarnell.github.io/triangle/>,
<https://bertcarnell.github.io/triangle/>

BugReports <https://github.com/bertcarnell/triangle/issues>

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 2.14.1)

Collate 'cdfe.R' 'dtriangle.R' 'fit-plots.R' 'ltriangle.r'
'mle-utils.R' 'mle.R' 'mom.R' 'ptriangle.r' 'qtriangle.R'
'rtriangle.r'

Imports assertthat, stats4, methods

Suggests testthat, knitr, rmarkdown, MASS, Rmpfr

VignetteBuilder knitr

Repository <https://bertcarnell.r-universe.dev>

RemoteUrl <https://github.com/bertcarnell/triangle>

RemoteRef HEAD

RemoteSha c4225c895bdd9699b5c820ce039d43f59273ab79

Contents

<code>compare_triangle_fit</code>	2
<code>ltriangle</code>	3
<code>qqtriangle</code>	4

standard_triangle_mle	5
summary.triangle_mle	6
triangle	7
triangle_cdf	9
triangle_mle	10
triangle_mom	11

Index	12
--------------	-----------

compare_triangle_fit *Compare multiple triangle distributions fits*

Description

Compare multiple triangle distributions fits

Usage

```
compare_triangle_fit(
  y,
  cols = c("red", "blue", "green"),
  main = "Triangle Fit Comparison",
  ...
)
```

Arguments

y	the triangle distributed sample
cols	the colors of the CDF-based estimates, the maximum likelihood estimates, and the method of moments estimates
main	the plot title
...	other parameters passed to plot.ecdf

Examples

```
set.seed(10304)
xtest <- rtriangle(100, 1, 5, 2)
compare_triangle_fit(xtest)
```

`ltriangle`*The Log-Triangle Distribution*

Description

These functions provide information about the triangle distribution on the logarithmic interval from a to b with a maximum at c . `dltriangle` gives the density, `pltriangle` gives the distribution function, `qltriangle` gives the quantile function, and `rltriangle` generates n random deviates.

Usage

```
rltriangle(  
  n = 1,  
  a = 1,  
  b = 100,  
  c = 10^((log10(a) + log10(b))/2),  
  logbase = 10  
)
```

```
dltriangle(x, a = 1, b = 100, c = 10^((log10(a) + log10(b))/2), logbase = 10)
```

```
pltriangle(q, a = 1, b = 100, c = 10^((log10(a) + log10(b))/2), logbase = 10)
```

```
qltriangle(p, a = 1, b = 100, c = 10^((log10(a) + log10(b))/2), logbase = 10)
```

Arguments

<code>n</code>	number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required.
<code>a</code>	lower limit of the distribution.
<code>b</code>	upper limit of the distribution.
<code>c</code>	mode of the distribution.
<code>logbase</code>	the base of the logarithmic scale to use (default to 10)
<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.

Details

All probabilities are lower tailed probabilities. a , b , and c may be appropriate length vectors except in the case of `rtriangle`.

Value

`dltriangle` gives the density, `pltriangle` gives the distribution function, `qltriangle` gives the quantile function, and `rltriangle` generates random deviates. Invalid arguments will result in return value `NaN` or `NA`.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

.[Random.seed](#) about random number generation, [runif](#), etc for other distributions.

Examples

```
tri <- rtriangle(100000, 1, 100, 10)
hist(log10(tri), breaks=100, main="Triangle Distribution", xlab="x")
dtriangle(10, 1, 100, 10) # 2/(log10(b)-log10(a)) = 1
qltriangle(pltriangle(10)) # 10
```

qqtriangle

Quantile-Quantile Plot for Triangle Distributed Data

Description

Quantile-Quantile Plot for Triangle Distributed Data

Usage

```
qqtriangle(
  y,
  a,
  b,
  c,
  main = "Triangle Q-Q Plot",
  xlab = "Theoretical Quantiles",
  ylab = "Sample Quantiles",
  ...
)
```

Arguments

y	the triangle distributed sample
a	the theoretical distribution triangle minimum parameter
b	the theoretical distribution triangle maximum parameter
c	the theoretical distribution triangle mode parameter
main	the plot title
xlab	the x-axis label
ylab	the y-axis label
...	other parameters passed to qqplot

Value

a list of x-y coordinates on the plot

Examples

```
set.seed(10304)
xtest <- rtriangle(100, 1, 5, 2)
theta <- coef(triangle_mle(xtest))
qqtriangle(xtest, theta[1], theta[2], theta[3])
```

standard_triangle_mle *Maximum likelihood estimate of the standard triangle distribution mode*

Description

Maximum likelihood estimate of the standard triangle distribution mode

Usage

```
standard_triangle_mle(x, debug = FALSE)
```

Arguments

x	sample from a triangle distribution
debug	if TRUE then the function will check the input parameters and print calculation information

Value

an object of S3 class `triangle_mle` containing a list with the call, coefficients, variance co-variance matrix, minimum negative log likelihood, number of observations, and the sample

References

Samuel Kotz and Johan Rene van Dorp. Beyond Beta [doi:10.1142/5720](https://doi.org/10.1142/5720)

Examples

```
xtest <- c(0.1, 0.25, 0.3, 0.4, 0.45, 0.6, 0.75, 0.8)
standard_triangle_mle(xtest)

xtest <- rtriangle(20, 0, 1, 0.63)
standard_triangle_mle(xtest)
```

summary.triangle_mle *Utility Methods for S3 class triangle_mle*

Description

Utility Methods for S3 class triangle_mle

Usage

```
## S3 method for class 'triangle_mle'
summary(object, ...)

## S3 method for class 'triangle_mle'
print(x, ...)

## S3 method for class 'triangle_mle'
coef(object, ...)

## S3 method for class 'triangle_mle'
logLik(object, ...)

## S3 method for class 'triangle_mle'
AIC(object, ..., k = 2)

## S3 method for class 'triangle_mle'
BIC(object, ...)

## S3 method for class 'triangle_mle'
vcov(object, ...)

## S3 method for class 'triangle_mle'
profile(fitted, ...)

## S3 method for class 'triangle_mle'
confint(object, parm, level = 0.95, ...)
```

Arguments

object	class triangle_mle from a call to triangle_mle()
...	not used except for print (other arguments passed to printCoefmat)
x	the triangle_mle object
k	the penalty per parameter to be used; the default k = 2
fitted	an object of class triangle_mle
parm	parameters to be given confidence intervals passed to stats4::confint
level	confidence interval level passed to stats4::confint

Value

an object of class `summary.mle`
`print.triangle_mle`: x invisibly
`coef.triangle_mle`: a vector of coefficients
`logLik.triangle_mle`: an object of class `logLik`
`AIC.triangle_mle`: the AIC
`BIC.triangle_mle`: the BIC
`vcov.triangle_mle`: the variance co-variance matrix
`profile.triangle_mle`: an object of class `profile.mle`
`confint.triangle_mle`: a matrix of parameter confidence intervals

Examples

```
set.seed(1234)
x <- rtriangle(100, 0, 1, 0.5)
mle1 <- triangle_mle(x)
summary(mle1)
print(mle1)
coef(mle1)
logLik(mle1)
AIC(mle1)
BIC(mle1)
vcov(mle1)
## Not run:
prof <- profile(mle1)
stats4::plot(prof)
confint(mle1, 1:3, level = 0.95)

## End(Not run)
```

triangle

The Triangle Distribution

Description

These functions provide information about the triangle distribution on the interval from a to b with a maximum at c . `dtriangle` gives the density, `ptriangle` gives the distribution function, `qtriangle` gives the quantile function, and `rtriangle` generates n random deviates.

Usage

```
dtriangle(x, a = 0, b = 1, c = (a + b)/2)
```

```
ptriangle(q, a = 0, b = 1, c = (a + b)/2)
```

```
qtriangle(p, a = 0, b = 1, c = (a + b)/2)
```

```
rtriangle(n = 1, a = 0, b = 1, c = (a + b)/2)
```

Arguments

x, q	vector of quantiles.
a	lower limit of the distribution.
b	upper limit of the distribution.
c	mode of the distribution.
p	vector of probabilities.
n	number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required.

Details

All probabilities are lower tailed probabilities. a, b, and c may be appropriate length vectors except in the case of `rtriangle`. `rtriangle` is derived from a draw from `runif`. The triangle distribution has density:

$$f(x) = \frac{2(x - a)}{(b - a)(c - a)}$$

for $a \leq x < c$.

$$f(x) = \frac{2(b - x)}{(b - a)(b - c)}$$

for $c \leq x \leq b$. $f(x) = 0$ elsewhere. The mean and variance are:

$$E(x) = \frac{(a + b + c)}{3}$$

$$V(x) = \frac{1}{18}(a^2 + b^2 + c^2 - ab - ac - bc)$$

Value

`dtriangle` gives the density, `ptriangle` gives the distribution function, `qtriangle` gives the quantile function, and `rtriangle` generates random deviates. Invalid arguments will result in return value `NaN` or `NA`.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

[.Random.seed](#) about random number generation, [runif](#), etc for other distributions.

Examples

```
## view the distribution
tri <- rtriangle(100000, 1, 5, 3)
hist(tri, breaks=100, main="Triangle Distribution", xlab="x")
mean(tri) # 1/3*(1 + 5 + 3) = 3
var(tri) # 1/18*(1^2 + 3^2 + 5^2 - 1*5 - 1*3 - 5*3) = 0.666667
dtriangle(0.5, 0, 1, 0.5) # 2/(b-a) = 2
qtriangle(ptriangle(0.7)) # 0.7
```

triangle_cdfc	<i>Triangle parameter estimates using a non-linear fit of the empirical CDF</i>
---------------	---

Description

Triangle parameter estimates using a non-linear fit of the empirical CDF

Usage

```
triangle_cdfc(x, control = stats::nls.control(maxiter = 100, warnOnly = TRUE))
```

Arguments

x	the triangle distributed sample
control	an object created by stats::nls.control

Value

an object of class nls

Examples

```
set.seed(10304)
xtest <- rtriangle(100, 1, 5, 2)
cdfc <- triangle_cdfc(xtest)
print(cdfc)
summary(cdfc)
coef(cdfc)
## Not run:
  confint(cdfc)

## End(Not run)
```

triangle_mle	<i>Maximum likelihood estimate of the triangle distribution parameters</i>
--------------	--

Description

Maximum likelihood estimate of the triangle distribution parameters

Usage

```
triangle_mle(x, debug = FALSE, maxiter = 100)
```

Arguments

x	sample from a triangle distribution
debug	if TRUE then the function will check the input parameters and print calculation information
maxiter	the maximum number of cycles of optimization between maximizing a and b given c and maximizing c given a and b

Value

an object of S3 class `triangle_mle` containing a list with the call, coefficients, variance co-variance matrix, minimum negative log likelihood, details of the optimization number of observations, and the sample

References

Samuel Kotz and Johan Rene van Dorp. Beyond Beta [doi:10.1142/5720](https://doi.org/10.1142/5720)

Examples

```
xtest <- c(0.1, 0.25, 0.3, 0.4, 0.45, 0.6, 0.75, 0.8)
triangle_mle(xtest)
```

```
xtest <- rtriangle(20, 1, 5, 3.5)
triangle_mle(xtest)
```

triangle_mom	<i>Triangle distribution method of moments estimate</i>
--------------	---

Description

Triangle distribution method of moments estimate

Usage

```
triangle_mom(x)
```

Arguments

x triangle distribution sample

Value

a named vector of the parameter estimates

Examples

```
set.seed(1204)
x <- rtriangle(20, 0, 2, 1.5)
triangle_mom(x)
```

Index

* **distribution**

ltriangle, 3
triangle, 7
.Random.seed, 4, 8

AIC.triangle_mle
(summary.triangle_mle), 6

BIC.triangle_mle
(summary.triangle_mle), 6

coef.triangle_mle
(summary.triangle_mle), 6
compare_triangle_fit, 2
confint.triangle_mle
(summary.triangle_mle), 6

dltriangle (ltriangle), 3
dtriangle (triangle), 7

logLik.triangle_mle
(summary.triangle_mle), 6
ltriangle, 3

pltriangle (ltriangle), 3
print.triangle_mle
(summary.triangle_mle), 6
profile.triangle_mle
(summary.triangle_mle), 6
ptriangle (triangle), 7

qltriangle (ltriangle), 3
qqtriangle, 4
qtriangle (triangle), 7

rltriangle (ltriangle), 3
rtriangle (triangle), 7
runif, 4, 8

standard_triangle_mle, 5
summary.triangle_mle, 6

triangle, 7
triangle_cdf, 9
triangle_mle, 10
triangle_mom, 11

vcov.triangle_mle
(summary.triangle_mle), 6