

# Package ‘tggd’

October 14, 2022

**Type** Package

**Title** The Standard Distribution Functions for the Truncated Generalised Gamma Distribution

**Version** 0.1.1

**Date** 2015-12-03

**Author** Aaron Robotham, Steven Murray

**Maintainer** Aaron Robotham <aaron.robotham@uwa.edu.au>

**Description** Density, distribution function, quantile function and random generation for the Truncated Generalised Gamma Distribution (also in  $\log_{10}(x)$  and  $\ln(x)$  space).

**License** GPL-3

**LazyData** TRUE

**Depends** R ( $\geq 3.00$ ), gsl

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-12-03 13:05:16

## R topics documented:

tggd	1
tggd_ln	3
tggd_log	5

<b>Index</b>	<b>8</b>
--------------	----------

---

tggd *The Truncated Generalised Gamma Distribution*

---

## Description

Density, distribution function, quantile function and random generation for the Truncated Generalised Gamma Distribution in linear space.

**Usage**

```
dtggd(x, scale=1e14, a=-1, b=1, xmin=1e10, log = FALSE)
ptggd(q, scale=1e14, a=-1, b=1, xmin=1e10, lower.tail = TRUE, log.p = FALSE)
qtggd(p, scale=1e14, a=-1, b=1, xmin=1e10, lower.tail = TRUE, log.p = FALSE,
res.approx=1e-2)
rtggd(n, scale=1e14, a=-1, b=1, xmin=1e10, res.approx=1e-2)
```

**Arguments**

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations. If length(n) > 1, the length is taken to be the number required.
scale	Vector of values for scale, which controls the transition regime between the power law slope and the exponential cut-off of the TGGD. This is analogous to the scale parameter for the standard Gamma distribution (see <a href="#">GammaDist</a> ).
a	Vector of values for a, which controls the power law slope of the TGGD.
b	Vector of values for b, which controls the exponential cutoff of the TGGD.
xmin	Vector of values for xmin, which controls the lower limit at which to truncate the TGGD.
res.approx	The resolution used to create the inverted CDF required to map probability integrals back onto quantiles.
log, log.p	Logical; if TRUE, probabilities/densities p are returned as log(p).
lower.tail	Logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].

**Details**

This distribution function is described in detail in Murray, Robotham and Power 2016. The PDF is given by:

$$f(x; a, b, s, m) = \frac{b(\frac{x}{s})^a \exp(-(\frac{x}{s})^b)}{s\Gamma(\frac{a+1}{b}, (\frac{m}{s})^b)}$$

where, from the argument list above, we use x=x, a=a, b=b, s=scale and m=xmin.  $\Gamma$  is the upper incomplete Gamma function as defined by the gsl [gamma\\_inc](#) function, using the same argument ordering, where `gamma_inc(a, x) == pgamma(x, a, lower=FALSE)*gamma(x)` for a>0. [gamma\\_inc](#) is used because it allows for the computation of upper incomplete integrals in cases where a<=0.

**Value**

dtggd gives the density, ptggd gives the distribution function, qtggd gives the quantile function, and rtggd generates random deviates.

Invalid arguments will result in return value NaN, with a warning.

The length of the result is determined by n for rtggd, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than n are recycled to the length of the result. Only the first elements of the logical arguments are used.

**Author(s)**

Aaron Robotham, Steven Murray

**References**

Murray, Robotham and Power (2016)

**See Also**

[GammaDist](#) regarding the Gamma distribution. [RNG](#) about random number generation in R. [Distributions](#) for other standard distributions.

**Examples**

```
r <- rtggd(100,a=-2)
hist(log10(r))

##Ideally the output below should equal 1, in practice it will be very close:
qtggd(ptggd(r))/r

#These should be the same:
integrate(dtggd,lower=1e10,upper=1e11,a=-1.5,b=0.7,xmin=1e10)
ptggd(1e11,a=-1.5,b=0.7,xmin=1e10)

#This should be very close to 1 (for a true PDF):
ptggd(1e18,a=-1.5,b=0.7,xmin=1e10)

#To show the link to the log10 (called log) and ln variants (and the slight inaccuracies)
#these outputs should be a sequence from 0 to 1 (by=0.1):
ptggd(10^qtggd_log(seq(0,1,by=0.1)))
ptggd(exp(qtggd_ln(seq(0,1,by=0.1))))
```

---

tggd\_ln

*The Truncated Generalised Gamma Distribution*


---

**Description**

Density, distribution function, quantile function and random generation for the Truncated Generalised Gamma Distribution in natural log (ln) space. Specifically, if  $\exp(x)$  is drawn from a TGGD distribution (in real space), these functions give the distribution of  $x$ , using the same parameter values.

**Usage**

```
dtggd_ln(x, scale=log(1e14), a=-1, b=1, xmin=log(1e10), log = FALSE)
ptggd_ln(q, scale=log(1e14), a=-1, b=1, xmin=log(1e10), lower.tail = TRUE, log.p = FALSE)
qtggd_ln(p, scale=log(1e14), a=-1, b=1, xmin=log(1e10), lower.tail = TRUE, log.p = FALSE,
res.approx=1e-2)
rtggd_ln(n, scale=log(1e14), a=-1, b=1, xmin=log(1e10), res.approx=1e-2)
```

**Arguments**

<code>x, q</code>	Vector of quantiles.
<code>p</code>	Vector of probabilities.
<code>n</code>	Number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>scale</code>	Vector of values for scale, which controls the transition regime between the power law slope and the exponential cut-off of the TGGD. This is analogous to the scale parameter for the standard Gamma distribution (see <a href="#">GammaDist</a> ).
<code>a</code>	Vector of values for a, which controls the power law slope of the TGGD.
<code>b</code>	Vector of values for b, which controls the exponential cutoff of the TGGD.
<code>xmin</code>	Vector of values for xmin, which controls the lower limit at which to truncate the TGGD.
<code>res.approx</code>	The resolution used to create the inverted CDF required to map probability integrals back onto quantiles.
<code>log, log.p</code>	Logical; if TRUE, probabilities/densities p are returned as log(p).
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

**Details**

This distribution function is described in detail in Murray, Robotham and Power 2016.

$$f(x; a, b, s, m) = \frac{b \exp((x - s)^{(a+1)}) \exp(-\exp(b(x - s)))}{s \Gamma(\frac{a+1}{b}, \exp(m - s)^b)}$$

where, from the argument list above, we use  $x=x$ ,  $a=a$ ,  $b=b$ ,  $s=scale$  and  $m=xmin$ .  $\Gamma$  is the upper incomplete Gamma function as defined by the gsl [gamma\\_inc](#) function, using the same argument ordering, where `gamma_inc(a, x) == pgamma(x, a, lower=FALSE) * gamma(x)` for  $a > 0$ . [gamma\\_inc](#) is used because it allows for the computation of upper incomplete integrals in cases where  $a \leq 0$ .

**Value**

`dtggd_In` gives the density, `ptggd_In` gives the distribution function, `qtggd_In` gives the quantile function, and `rtggd_In` generates random deviates.

Invalid arguments will result in return value NaN, with a warning.

The length of the result is determined by `n` for `rtggd_In`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

**Note**

The intended application of the log-space version of the TGGD is to provide the correct distribution when variates are drawn from a real-space TGGD, but there are priors on their uncertainty which operate in logspace (eg. a lognormal distribution). The likelihood of a given set of parameters is incorrect in such a case if the real-space version is used without an adjustment to the Jacobian.

**Author(s)**

Aaron Robotham, Steven Murray

**References**

Murray, Robotham and Power (2016)

**See Also**

[GammaDist](#) regarding the Gamma distribution. [RNG](#) about random number generation in R. [Distributions](#) for other standard distributions.

**Examples**

```
r <- rtggd_ln(100,a=-2)
hist(r)

##Ideally the output below should equal 0, in practice it will be very close:
qtggd_ln(ptggd_ln(r))-r

#These should be the same:
integrate(dtggd_ln,lower=log(1e10),upper=log(1e11),a=-1.5,b=0.7,xmin=log(1e10))
ptggd_ln(log(1e11),a=-1.5,b=0.7,xmin=log(1e10))

#This should be very close to 1 (for a true PDF):
ptggd_ln(log(1e18),a=-1.5,b=0.7,xmin=log(1e10))

#To show the link to the linear and log10 (called log) variants (and the slight
#inaccuracies) these outputs should be a sequence from 0 to 1 (by=0.1):
ptggd_log(log10(qtggd(seq(0,1,by=0.1))))
ptggd_log(qtggd_ln(seq(0,1,by=0.1))/log(10))
```

---

tggd\_log

*The Truncated Generalised Gamma Distribution*


---

**Description**

Density, distribution function, quantile function and random generation for the Truncated Generalised Gamma Distribution in log base 10 (log10) space. Specifically, if  $10^x$  is drawn from a TGGD distribution (in real space), these functions give the distribution of  $x$ , using the same parameter values.

**Usage**

```
dtggd_log(x, scale=14, a=-1, b=1, xmin=10, log = FALSE)
ptggd_log(q, scale=14, a=-1, b=1, xmin=10, lower.tail = TRUE, log.p = FALSE)
qtggd_log(p, scale=14, a=-1, b=1, xmin=10, lower.tail = TRUE, log.p = FALSE,
res.approx=1e-2)
rtggd_log(n, scale=14, a=-1, b=1, xmin=10, res.approx=1e-2)
```

**Arguments**

<code>x, q</code>	Vector of quantiles.
<code>p</code>	Vector of probabilities.
<code>n</code>	Number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>scale</code>	Vector of values for scale, which controls the transition regime between the power law slope and the exponential cut-off of the TGGD. This is analogous to the scale parameter for the standard Gamma distribution (see <a href="#">GammaDist</a> ).
<code>a</code>	Vector of values for a, which controls the power law slope of the TGGD.
<code>b</code>	Vector of values for b, which controls the exponential cutoff of the TGGD.
<code>xmin</code>	Vector of values for xmin, which controls the lower limit at which to truncate the TGGD.
<code>res.approx</code>	The resolution used to create the inverted CDF required to map probability integrals back onto quantiles.
<code>log, log.p</code>	Logical; if TRUE, probabilities/densities p are returned as log(p).
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

**Details**

This distribution function is described in detail in Murray, Robotham and Power 2016. The PDF is given by:

$$f(x; a, b, s, m) = \frac{\ln(10) \cdot b \cdot (10^{(x-s)})^{(a+1)} \exp(-10^{(b(x-s))})}{s \Gamma\left(\frac{a+1}{b}, (10^{(m-s)})^b\right)}$$

where, from the argument list above, we use  $x=x$ ,  $a=a$ ,  $b=b$ ,  $s=scale$  and  $m=xmin$ .  $\Gamma$  is the upper incomplete Gamma function as defined by the gsl [gamma\\_inc](#) function, using the same argument ordering, where `gamma_inc(a, x) == pgamma(x, a, lower=FALSE) * gamma(x)` for  $a > 0$ . [gamma\\_inc](#) is used because it allows for the computation of upper incomplete integrals in cases where  $a \leq 0$ .

**Value**

`dtggd_log` gives the density, `ptggd_log` gives the distribution function, `qtggd_log` gives the quantile function, and `rtggd_log` generates random deviates.

Invalid arguments will result in return value NaN, with a warning.

The length of the result is determined by `n` for `rtggd_log`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

**Note**

The intended application of the log-space version of the TGGD is to provide the correct distribution when variates are drawn from a real-space TGGD, but there are priors on their uncertainty which operate in logspace (eg. a lognormal distribution). The likelihood of a given set of parameters is incorrect in such a case if the real-space version is used without an adjustment to the Jacobian.

**Author(s)**

Aaron Robotham, Steven Murray

**References**

Murray, Robotham and Power (2016)

**See Also**

[GammaDist](#) regarding the Gamma distribution. [RNG](#) about random number generation in R. [Distributions](#) for other standard distributions.

**Examples**

```
r <- rtggd_log(100,a=-2)
hist(r)

##Ideally the output below should equal 0, in practice it will be very close:
qtggd_log(ptggd_log(r))-r

#These should be the same:
integrate(dtggd_log,lower=10,upper=11,a=-1.5,b=0.7,xmin=10)
ptggd_log(11,a=-1.5,b=0.7,xmin=10)

#This should be very close to 1 (for a true PDF):
ptggd_log(18,a=-1.5,b=0.7,xmin=10)

#To show the link to the linear and ln variants (and the slight inaccuracies) these
#outputs should be a sequence from 0 to 1 (by=0.1):
ptggd_ln(log(qtggd(seq(0,1,by=0.1))))
ptggd_ln(qtggd_log(seq(0,1,by=0.1))*log(10))
```

# Index

\* **~distribution**

tggd, 1  
tggd\_ln, 3  
tggd\_log, 5

\* **~tggd\_ln**

tggd\_ln, 3

\* **~tggd\_log**

tggd\_log, 5

\* **~tggd**

tggd, 1

Distributions, 3, 5, 7

dtggd (tggd), 1

dtggd\_ln (tggd\_ln), 3

dtggd\_log (tggd\_log), 5

gamma\_inc, 2, 4, 6

GammaDist, 2–7

ptggd (tggd), 1

ptggd\_ln (tggd\_ln), 3

ptggd\_log (tggd\_log), 5

qtggd (tggd), 1

qtggd\_ln (tggd\_ln), 3

qtggd\_log (tggd\_log), 5

RNG, 3, 5, 7

rtggd (tggd), 1

rtggd\_ln (tggd\_ln), 3

rtggd\_log (tggd\_log), 5

tggd, 1

tggd\_ln, 3

tggd\_log, 5