# Package ‘ggamma’

December 15, 2019

**Title** Generalized Gamma Probability Distribution

**Version** 1.0.1

**Description** Density, distribution function, quantile function and random generation for the Generalized Gamma proposed in Stacy, E. W. (1962) <doi:10.1214/aoms/1177704481>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**URL** https://mjsaldanha.com/posts/ggamma

**BugReports** https://github.com/matheushjs/ggamma/issues

**RoxygenNote** 7.0.1

**Depends** R (>= 3.1.0)

**Suggests** testthat, flexsurv

**NeedsCompilation** no

**Author** Matheus H. J. Saldanha [aut, cre],
Adriano K. Suzuki [aut]

**Maintainer** Matheus H. J. Saldanha <mhjsaldanha@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-12-15 14:40:02 UTC

## R topics documented:

G.Gamma *Generalized Gamma Probability Distribution*

## Description

Fast implementation of density, distribution function, quantile function and random generation for the Generalized Gamma probability distribution.

## Usage

```
dggamma(x, a, b, k, log = F)

pggamma(q, a, b, k, lower.tail = TRUE, log.p = FALSE)

qggamma(p, a, b, k, lower.tail = TRUE, log.p = FALSE)

rggamma(n, a, b, k)
```

## Arguments

| | |
|---|---|
| `x, q` | vector of quantiles. |
| `a, b, k` | Parameters of the distribution, all of which must be positive. |
| `log, log.p` | logical; if TRUE, probabilities p are given as log(p). |
| `lower.tail` | logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$. |
| `p` | vector of probabilities. |
| `n` | number of observations. If `length(n) > 1`, the length is taken to be the number required. |

## Details

The generalized gamma distribution proposed by Stacy (1962) has parameters $a, d, p$, but here we adopt the reparametrization

$$a = a$$

$$b = p$$

$$k = \frac{d}{p}$$

as is used by the R package *flexsurv*.

Probability density function

$$f(x) = \frac{bx^{bk-1}\exp[-(x/a)^b]}{a^{bk}\Gamma(k)}$$

Cumulative density function

$$F(x) = \frac{\gamma(k, (x/a)^b)}{\Gamma(k)}$$

The above function can be written in terms of a $Gamma(\alpha, \beta)$. Let $T \sim Gamma(k, 1)$ and its cumulative distribution be denoted as $F_T(t)$, then the cumulative density function of the generalized gamma distribution can be written as

$$F(x) = F_T((x/a)^b)$$

which allows us to write the quantile function of the generalized gamma in terms of the gamma one ($Q_T(u)$ is the quantile function of $T$)

$$Q(u) = (Q_T(u) \cdot a)^{1/b}$$

from which random numbers can be drawn.

## References

Stacy, E. W. (1962). A generalization of the gamma distribution. The Annals of mathematical statistics, 33(3), 1187-1192.

## Examples

```
x = seq(0.001, 5, length=1000);
plot(x, dggamma(x, 3, 1.8, 0.5), col=2, type="l", lwd=4, ylim=c(0, 1));
lines(x, pggamma(x, 3, 1.8, 0.5), col=4, type="l", lwd=4, ylim=c(0, 1));
legend("right", c("PDF", "CDF"), col=c(2, 4), lwd=4);

r = rgamma(n = 100, 2, 2);
lik = function(params) -sum(dggamma(r, params[1], params[2], params[3], log=TRUE));
optPar = optim(lik, par=c(1, 1, 1), method="L-BFGS", lower=0.00001, upper=Inf)$par;
x = seq(0.001, 5, length=1000);
plot(x, dgamma(x, 2, 2), type="l", col=2, lwd=4, ylim=c(0, 1));
lines(x, dggamma(x, optPar[1], optPar[2], optPar[3]), col=4, lwd=4);
legend("topright", c("Gamma(shape=2, rate=2)", "MLE Gen. Gamma"), col=c(2, 4), lwd=4);
```

---

# `ggamma` *Generalized Gamma Probability Distribution*

---

## Description

Density, distribution function, quantile function and random generation for the Generalized Gamma lifetime distributions.

## Details

This package follows naming convention that is consistent with base R, where density (or probability mass) functions, distribution functions, quantile functions and random generation functions names are followed by `d`, `p`, `q`, and `r` prefixes.

Behaviour of the functions is consistent with base R, where for not valid parameters values `NaN`'s are returned, while for values beyond function support `0`'s are returned (e.g. for non-integers in discrete distributions, or for negative values in functions with non-negative support).

C++ was not used, as the R code proved itself most efficient. See the package website page for more details.

# Index