

# Package ‘gbeta’

November 19, 2020

**Type** Package

**Title** Generalized Beta and Beta Prime Distributions

**Version** 0.1.0

**Author** Stéphane Laurent

**Maintainer** Stéphane Laurent <laurent\_step@outlook.fr>

**Description** Density, distribution function, quantile function, and random generation for the generalized Beta and Beta prime distributions. The family of generalized Beta distributions is conjugate for the Bayesian binomial model, and the generalized Beta prime distribution is the posterior distribution of the relative risk in the Bayesian 'two Poisson samples' model when a Gamma prior is assigned to the Poisson rate of the reference group and a Beta prime prior is assigned to the relative risk. References: Laurent (2012) <doi:10.1214/11-BJPS139>, Hamza & Vallois (2016) <doi:10.1016/j.spl.2016.03.014>, Chen & Novick (1984) <doi:10.3102/10769986009002163>.

**License** GPL (>= 2)

**Imports** Rcpp (>= 1.0.5), gsl, Runuran

**LinkingTo** Rcpp, RcppNumerical, RcppEigen

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**URL** <https://github.com/stla/gbeta>

**BugReports** <https://github.com/stla/gbeta/issues>

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2020-11-19 09:00:02 UTC

## R topics documented:

GBeta	2
GBetaP	3

---

GBeta                                      *Generalized Beta distribution*

---

### Description

Density, distribution function, quantile function, and random generation for the generalized Beta distribution.

### Usage

```

dgbeta(u, c, d, kappa, tau, log = FALSE)

pgbeta(q, c, d, kappa, tau)

rgbeta(n, c, d, kappa, tau, method = "mixture")

qgbeta(p, c, d, kappa, tau)

```

### Arguments

u	numeric vector
c, d, kappa, tau	parameters; they must be strictly positive numbers, except kappa which can take any value
log	logical, whether to return the log-density
q	numeric vector of quantiles
n	positive integer, the desired number of simulations
method	the method of random generation, "mixture" or "arou"; only a positive kappa is allowed for the "mixture" method, but this method is faster
p	numeric vector of probabilities

### References

- Marwa Hamza & Pierre Vallois. *On Kummer's distributions of type two and generalized Beta distributions*. *Statistics & Probability Letters* 118 (2016), pp. 60-69. <doi:10.1016/j.spl.2016.03.014>
- James J. Chen & Melvin R. Novick. *Bayesian Analysis for Binomial Models with Generalized Beta Prior Distributions*. *Journal of Educational Statistics* 9, No. 2 (1984), pp. 163-175. <doi:10.3102/10769986009002163>

### Examples

```

library(gbeta)
curve(dgbeta(x, 4, 12, 10, 0.01), axes = FALSE, lwd = 2)
axis(1)

```

---

GBetaP                      *Generalized Beta prime distribution*

---

**Description**

Density, distribution function, quantile function, and random generation for the generalized Beta prime distribution.

**Usage**

```
dgbetap(x, c, d, kappa, tau, scale = 1, log = FALSE)
```

```
pgbetap(q, c, d, kappa, tau, scale = 1)
```

```
rgbetap(n, c, d, kappa, tau, scale = 1, method = "mixture")
```

```
qgbetap(p, c, d, kappa, tau, scale = 1)
```

**Arguments**

x	numeric vector
c, d, kappa, tau	parameters; they must be strictly positive numbers, except kappa which can take any value
scale	scale parameter, a strictly positive number
log	logical, whether to return the log-density
q	numeric vector of quantiles
n	positive integer, the desired number of simulations
method	the method of random generation, "mixture" or "arou"; only a positive kappa is allowed for the "mixture" method, but this method is faster
p	numeric vector of probabilities

**References**

- Stéphane Laurent. *Some Poisson mixtures distributions with a hyperscale parameter*. Brazilian Journal of Probability and Statistics 26, No. 3 (2012), pp. 265-278. <doi:10.1214/11-BJPS139>
- Myriam Chabot. *Sur l'estimation du rapport de deux paramètres d'intensité poissonniens et l'estimation par fonctions de masse prédictives*. Master thesis. Université de Scherbrooke, 2016.

**Examples**

```
library(gbeta)
curve(dgbetap(x, 4, 12, 10, 0.01), to = 10, axes = FALSE, lwd = 2)
axis(1)
```

# Index

[dgbeta \(GBeta\)](#), [2](#)  
[dgbetap \(GBetaP\)](#), [3](#)

[GBeta](#), [2](#)  
[GBetaP](#), [3](#)

[pgbeta \(GBeta\)](#), [2](#)  
[pgbetap \(GBetaP\)](#), [3](#)

[qgbeta \(GBeta\)](#), [2](#)  
[qgbetap \(GBetaP\)](#), [3](#)

[rgbeta \(GBeta\)](#), [2](#)  
[rgbetap \(GBetaP\)](#), [3](#)