

Package: finitization (via r-universe)

June 9, 2026

Type Package

Title Moment-Preserving Finitization for Power-Series Distributions

Description Implements the finitization of discrete probability distributions, a technique that approximates a distribution by preserving a finite number of moments. These finitized distributions enable faster random variate generation than inverse transform sampling and are useful in simulation and statistical modeling.

Version 0.1.0

Maintainer Bogdan Oancea <bogdan.oancea@gmail.com>

License GPL (>= 2)

Encoding UTF-8

Depends R (>= 4.0.0)

NeedsCompilation yes

SystemRequirements GMP, CLN, GiNaC, pkg-config, make

OS_type unix

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

URL <https://github.com/bogdanoancea/finitization>

BugReports <https://github.com/bogdanoancea/finitization/issues>

Imports Rcpp, rootSolve

Suggests testthat (>= 3.0.0), knitr, rmarkdown, extraDistr, truncdist, ggplot2

Config/testthat/edition 3

VignetteBuilder knitr

LinkingTo Rcpp

Collate 'RcppExports.R' 'utils.R' 'binom.R' 'finitization.R' 'get_n.R' 'log.R' 'negbinom.R' 'pois.R' 'zzz.R'

Config/pak/sysreqs libcln-dev libginac-dev libgmp3-dev pkg-config

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

Date/Publication 2026-03-05 09:44:47 UTC

RemoteUrl <https://github.com/bogdanoancea/finitization>

RemoteRef master

RemoteSha 7ab9719f6f5e4c4534ac0e5d0e54d03dc55321d5

Contents

dbinom	2
dlog	3
dnegbinom	4
dpois	5
getBinomialMFPS	5
getLogarithmicMFPS	6
getNegativeBinomialMFPS	7
getPoissonMFPS	7
pbinom	8
plog	9
pnegbinom	10
ppois	11
printFinitizedBinomialDensity	12
printFinitizedLogarithmicDensity	13
printFinitizedNegativeBinomialDensity	13
printFinitizedPoissonDensity	14
qbinom	15
qlog	16
qnegbinom	16
qpois	17
rbinom	18
rlog	19
rnegbinom	19
rpois	20
Index	21

dbinom	<i>The density for the finitized Binomial distribution.</i>
--------	---

Description

dbinom(*n*, *p*, *N*, *val*, *log*) computes the finitized Binomial density for each value in *val*.

Usage

```
dbinom(n, p, N, val = NULL, log = FALSE)
```

Arguments

n	The finitization order. An integer > 0.
p	The success probability for each trial (must satisfy $0 \leq p \leq 1$).
N	The number of trials.
val	A vector of values at which the density is computed. If NULL, a data frame containing all possible values (from 0 to n) and the corresponding densities is returned.
log	Logical; if TRUE, the (natural) logarithm of the probabilities is returned.

Value

A data.frame with two columns: val, which contains the values, and prob, which contains the corresponding density (or log density if log = TRUE).

Examples

```
library(finitization)
dbinom(4, 0.5, 4)
dbinom(4, 0.5, 4, log = TRUE)
```

dlog

The density for the Logarithmic distribution.

Description

dlog(n, theta, val, log) computes the finitized Logarithmic density for each value in val.

Usage

```
dlog(n, theta, val = NULL, log = FALSE)
```

Arguments

n	The finitization order. It should be an integer > 0.
theta	The parameter of the finitized Logarithmic distribution.
val	A vector with the values of the variable for which the probability density is computed. If NULL, a data frame containing all possible values, i.e. $0 \dots n$, and the corresponding probabilities is returned.
log	Logical; if TRUE, the (natural) logarithm of the computed probabilities is returned.

Value

A data.frame object with two columns: val containing the values and prob containing the corresponding densities (or their logarithms if log = TRUE).

Examples

```
library(finitization)
dlog(4, 0.1, c(0,1,3))
dlog(4, 0.1, c(0,1,3), log = TRUE)
```

dnegbinom

The density for the finitized Negative Binomial distribution.

Description

dnegbinom(n, q, k, val, log) computes the finitized Negative Binomial density for val.

Usage

```
dnegbinom(n, q, k, val = NULL, log = FALSE)
```

Arguments

n	The finitization order. It should be an integer > 0 .
q	The parameter of the finitized Negative Binomial distribution - the success probability for each trial ($q \in [0, 1]$).
k	The number of failures until the experiment is stopped, $k > 0$.
val	A vector with the values of the variable for which the probability density is computed. If NULL, a data frame containing all possible values, i.e. $0 \dots n$, and the corresponding probabilities is returned.
log	Logical; if TRUE, the (natural) logarithm of the computed densities is returned.

Value

A data.frame object with two columns: val containing the values and prob containing the corresponding densities (or their logarithms if log = TRUE).

Examples

```
library(finitization)
dnegbinom(4, 0.12, 4)
dnegbinom(4, 0.12, 4, log = TRUE)
```

dpois	<i>The density for the finitized Poisson distribution.</i>
-------	--

Description

dpois(n, theta, val, log) computes the finitized Poisson density for each value in val.

Usage

```
dpois(n, theta, val = NULL, log = FALSE)
```

Arguments

n	The finitization order. It should be an integer > 0.
theta	The parameter of the finitized Poisson distribution.
val	A vector with the values of the variable for which the probability density is computed. If NULL, a data frame containing all possible values (0, 1, ..., n) and the corresponding probabilities is returned.
log	Logical; if TRUE, the (natural) logarithm of the computed probabilities is returned.

Value

A data.frame object with two columns: val containing the values and prob containing the corresponding densities (or their logarithms if log = TRUE).

Examples

```
library(finitization)
dpois(4, 0.5, c(0,1,3))
dpois(4, 0.5, c(0,1,3), log = TRUE)
```

getBinomialMFPS	<i>Maximum feasible parameter space for the finitized Binomial distribution.</i>
-----------------	--

Description

getBinomialMFPS(n, N) computes and returns the maximum feasible parameter space for the finitized Binomial distribution.

Usage

```
getBinomialMFPS(n, N)
```

Arguments

n The finitization order. An integer > 0 .
N The number of trials.

Value

A vector of two elements, where the first element is the lower limit and the second element is the upper limit of the maximum feasible parameter space.

Examples

```
library(finitization)
getBinomialMFPS(2, 4)
```

getLogarithmicMFPS *Maximum feasible parameter space for the finitized Logarithmic distribution.*

Description

getLogarithmicMFPS(n) computes and returns the maximum feasible parameter space for the finitized Logarithmic distribution.

Usage

```
getLogarithmicMFPS(n)
```

Arguments

n The finitization order. It should be an integer > 0 .

Value

A vector with two elements where the first element is the lower limit and the second is the upper limit.

Examples

```
library(finitization)
getLogarithmicMFPS(1)
```

`getNegativeBinomialMFPS`*Maximum feasible parameter space for the finitized Negative Binomial distribution.*

Description

`getNegativeBinomialMFPS(n, k)` computes and returns the maximum feasible parameter space for the finitized Negative Binomial distribution with parameter k .

Usage

```
getNegativeBinomialMFPS(n, k)
```

Arguments

n The finitization order. It should be an integer > 0 .
 k The number of failures until the experiment is stopped, $k > 0$.

Value

A vector with two elements where the first element is the lower limit of the maximum feasible parameter space and the second is the upper limit.

Examples

```
library(finitization)
getNegativeBinomialMFPS(2, 4)
```

`getPoissonMFPS`*Maximum feasible parameter space for the finitized Poisson distribution.*

Description

`getPoissonMFPS(n)` computes and returns the maximum feasible parameter space for the finitized Poisson distribution.

Usage

```
getPoissonMFPS(n)
```

Arguments

n The finitization order. It should be an integer > 0 .

Value

A vector with two elements where the first element is the lower limit of the maximum feasible parameter space and the second is the upper limit.

Examples

```
library(finitization)
getPoissonMFPS(4)
```

pbinom	<i>The cumulative distribution function (CDF) for the finitized Binomial distribution.</i>
--------	--

Description

pbinom(n, p, N, val, lower.tail, log.p) computes the CDF for the finitized Binomial distribution at specified value(s).

Usage

```
pbinom(n, p, N, val = NULL, lower.tail = TRUE, log.p = FALSE)
```

Arguments

n	The finitization order. A positive integer (> 0).
p	The success probability for each trial ($0 \leq p \leq 1$).
N	The number of trials.
val	A numeric vector of values at which the CDF is computed. If NULL (default), the function returns the full CDF over all possible outcomes (0, 1, ..., n).
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq x)$; if FALSE, probabilities are $P(X > x)$.
log.p	Logical; if TRUE, returns cumulative probabilities on the log scale.

Value

If val is provided, a data.frame with two columns:

val The input values.

cdf The corresponding cumulative probabilities (or log-probabilities if log.p = TRUE).

If val is NULL, a data.frame with all possible values (0, ..., n) and their cumulative probabilities is returned.

Examples

```

library(finitization)
# CDF evaluated at specific points
pbinom(4, 0.5, 4, val = c(0, 2, 4))

# Full CDF over all values (0 to 4)
pbinom(4, 0.5, 4)

# CDF upper tail (P(X > x))
pbinom(4, 0.5, 4, lower.tail = FALSE)

# CDF on log scale
pbinom(4, 0.5, 4, log.p = TRUE)

```

plog	<i>The cumulative distribution function (CDF) for the finitized Logarithmic distribution.</i>
------	---

Description

plog(n, theta, val, log.p, lower.tail) computes the CDF for the finitized Logarithmic distribution at specified value(s).

Usage

```
plog(n, theta, val = NULL, log.p = FALSE, lower.tail = TRUE)
```

Arguments

n	The finitization order. A positive integer (> 0).
theta	The parameter of the finitized Logarithmic distribution.
val	A numeric vector of values at which to compute the CDF. If NULL (default), the function returns the full CDF over all values from 0 to n.
log.p	Logical; if TRUE, returns cumulative probabilities on the log scale.
lower.tail	Logical; if TRUE (default), probabilities are computed as $P(X \leq x)$; if FALSE, as $P(X > x)$.

Value

A data.frame with the following columns:

val The input values (either the user-supplied val or 0:n).

cdf The corresponding cumulative probabilities (or log-probabilities if log.p = TRUE).

Examples

```

library(finitization)
# Evaluate CDF at specific points
plog(4, 0.1, val = c(0, 2, 4))

# Full CDF from 0 to 4
plog(4, 0.1)

# Upper-tail probabilities
plog(4, 0.1, val = c(0, 2, 4), lower.tail = FALSE)

# Log-scale CDF
plog(4, 0.1, log.p = TRUE)

```

pnegbinom	<i>The cumulative distribution function (CDF) for the finitized Negative Binomial distribution.</i>
-----------	---

Description

pnegbinom(n, q, k, val, log.p, lower.tail) computes the CDF for the finitized Negative Binomial distribution at specified value(s).

Usage

```
pnegbinom(n, q, k, val = NULL, log.p = FALSE, lower.tail = TRUE)
```

Arguments

n	The finitization order. A positive integer (> 0).
q	The success probability for each trial ($0 \leq q \leq 1$).
k	The target number of failures before stopping; a positive integer.
val	A numeric vector of values at which to compute the CDF. If NULL (default), the function returns the full CDF over all possible outcomes (0, 1, ..., n).
log.p	Logical; if TRUE, returns cumulative probabilities on the log scale.
lower.tail	Logical; if TRUE (default), probabilities are computed as $P(X \leq x)$; if FALSE, as $P(X > x)$.

Value

A data.frame with two columns:

val The input values (either specified val or the full sequence from 0 to n).

cdf The corresponding cumulative probabilities (or log-probabilities if log.p = TRUE).

Examples

```

library(finitization)
# Compute CDF at specific values
pnegbinom(4, 0.12, 4, val = c(0, 2, 4))

# Compute full CDF over range 0 to 4
pnegbinom(4, 0.12, 4)

# Upper-tail probabilities
pnegbinom(4, 0.12, 4, val = c(0, 2, 4), lower.tail = FALSE)

# Log-scale cumulative probabilities
pnegbinom(4, 0.12, 4, log.p = TRUE)

```

ppois	<i>The cumulative distribution function (CDF) for the finitized Poisson distribution.</i>
-------	---

Description

ppois(n, theta, val, log.p, lower.tail) computes the CDF for the finitized Poisson distribution at specified value(s).

Usage

```
ppois(n, theta, val = NULL, log.p = FALSE, lower.tail = TRUE)
```

Arguments

n	The finitization order. A positive integer (> 0).
theta	The mean parameter of the finitized Poisson distribution.
val	A numeric vector of values at which to compute the CDF. If NULL (default), the function returns the full CDF over the entire support (0, 1, ..., n).
log.p	Logical; if TRUE, returns cumulative probabilities on the log scale.
lower.tail	Logical; if TRUE (default), probabilities are calculated as $P(X \leq x)$. If FALSE, returns upper-tail probabilities $P(X > x)$.

Value

A data.frame with columns:

val The evaluated values (input val or the full range 0:n).

cdf The corresponding cumulative probabilities (or log-probabilities if log.p = TRUE).

Examples

```
# Compute CDF at specific values
ppois(n = 4, theta = 0.5, val = c(0, 2, 4))

# Compute full CDF over range 0 to 4
ppois(n = 4, theta = 0.5)

# Upper-tail probabilities
ppois(n = 4, theta = 0.5, val = c(0, 2, 4), lower.tail = FALSE)

# Log-scale cumulative probabilities
ppois(n = 4, theta = 0.5, log.p = TRUE)
```

```
printFinitizedBinomialDensity
```

The string representation of the probability density function for the finitized Binomial distribution.

Description

printFinitizedBinomialDensity(n, N, val, latex) computes and prints the string representation of the probability density function for the finitized Binomial distribution.

Usage

```
printFinitizedBinomialDensity(n, N, val = NULL, latex = FALSE)
```

Arguments

n	The finitization order. An integer > 0.
N	The number of trials.
val	The value(s) at which the density is printed. If NULL, prints for all values from 0 to n.
latex	Logical; if TRUE, the output is formatted in LaTeX.

Value

Silently returns a character vector with the string representation(s) of the pdf.

Examples

```
library(finitization)
printFinitizedBinomialDensity(4, 4)
```

```
printFinitizedLogarithmicDensity
```

The string representation of the probability density function for the finitized Logarithmic distribution.

Description

`printFinitizedLogarithmicDensity(n, val, latex)` computes and prints the string representation of the pdf.

Usage

```
printFinitizedLogarithmicDensity(n, val = NULL, latex = FALSE)
```

Arguments

<code>n</code>	The finitization order. It should be an integer > 0.
<code>val</code>	The value for which the pdf is printed. If NULL, the pdf for all values (0 ... n) is printed.
<code>latex</code>	Logical; if TRUE the output is formatted in LaTeX.

Value

A character vector containing the string representation(s) of the pdf.

Examples

```
library(finitization)
printFinitizedLogarithmicDensity(4)
```

```
printFinitizedNegativeBinomialDensity
```

The string representation of the probability density function for the finitized Negative Binomial distribution.

Description

`printFinitizedNegativeBinomialDensity(n,k, val, latex)` computes and prints the string representation of the probability density function for the finitized Negative Binomial distribution with parameter k.

Usage

```
printFinitizedNegativeBinomialDensity(n, k, val = NULL, latex = FALSE)
```

Arguments

n	The finitization order. It should be an integer > 0.
k	The number of failures until the experiment is stopped, $k > 0$.
val	The value of the variable for which the probability density function is printed. If NULL, this function computes the string representation of the pdf for all possible values, i.e. $\{0 \dots n\}$.
latex	If TRUE, a string representation of the pdf formatted in Latex format is printed, otherwise this function prints the string representation of the pdf as an R expression.

Value

This function silently returns a vector of type `character` with the string representation of the pdf(s). The length of the vector is the same with the length of the parameter `val`, i.e. one element for each value in `val`.

Examples

```
library(finitization)
printFinitizedNegativeBinomialDensity(4, 4)
```

```
printFinitizedPoissonDensity
```

The string representation of the probability density function for the finitized Poisson distribution.

Description

`printFinitizedPoissonDensity(n, val, latex)` computes and prints the string representation of the probability density function for the finitized Poisson distribution.

Usage

```
printFinitizedPoissonDensity(n, val = NULL, latex = FALSE)
```

Arguments

n	The finitization order. It should be an integer > 0.
val	The value of the variable for which the probability density function is printed. If NULL, this function computes the string representation of the pdf for all possible values, i.e. $\{0 \dots n\}$.
latex	If TRUE, a string representation of the pdf formatted in Latex format is printed, otherwise this function prints the string representation of the pdf as an R expression.

Value

This function silently returns a vector of type `character` with the string representation of the pdf(s). The length of the vector is the same with the length of the parameter `val`, i.e. one element for each value in `val`.

Examples

```
library(finitization)
printFinitizedPoissonDensity(4)
```

qbinom

The quantile function for the finitized Binomial distribution.

Description

`qbinom(n, p, N, prob, lower.tail = TRUE, log.p = FALSE)` computes the quantile function for the finitized Binomial distribution.

Usage

```
qbinom(n, p, N, prob, lower.tail = TRUE, log.p = FALSE)
```

Arguments

<code>n</code>	The finitization order. An integer > 0 .
<code>p</code>	The success probability for each trial ($0 \leq p \leq 1$).
<code>N</code>	The number of trials.
<code>prob</code>	A vector of probabilities (or log-probabilities if <code>log.p = TRUE</code>) for which the quantiles are computed. Each probability must be between 0 and 1.
<code>lower.tail</code>	Logical; if <code>TRUE</code> (default) the input probabilities are interpreted as $P(X \leq x)$; if <code>FALSE</code> , they are interpreted as $P(X > x)$.
<code>log.p</code>	Logical; if <code>TRUE</code> , the probabilities in <code>prob</code> are assumed to be given on the logarithmic scale.

Value

A numeric vector of quantiles corresponding to the input probabilities. For each probability, the quantile is defined as the smallest integer $\lfloor x \rfloor$ (from 0 to `n`) such that the cumulative probability is at least the provided value.

Examples

```
library(finitization)
qbinom(4, 0.5, 4, prob = c(0.1, 0.5, 0.9))
qbinom(4, 0.5, 4, prob = log(c(0.1, 0.5, 0.9)), lower.tail = TRUE, log.p = TRUE)
qbinom(4, 0.5, 4, prob = c(0.1, 0.5, 0.9), lower.tail = FALSE)
```

qlog *The quantile function for the finitized Logarithmic distribution.*

Description

qlog(n, theta, p, lower.tail, log.p) computes the quantile function for the finitized Logarithmic distribution.

Usage

```
qlog(n, theta, p, lower.tail = TRUE, log.p = FALSE)
```

Arguments

n	The finitization order. An integer > 0.
theta	The parameter of the finitized Logarithmic distribution.
p	A numeric vector of probabilities. For lower.tail = TRUE (default), these are interpreted as $P(X \leq x)$; if lower.tail = FALSE they are interpreted as $P(X > x)$.
lower.tail	Logical; if TRUE (default) the input probabilities are lower-tail probabilities, otherwise upper-tail probabilities.
log.p	Logical; if TRUE, the probabilities in p are assumed to be given on the logarithmic scale.

Value

A numeric vector of quantiles corresponding to the input probabilities.

Examples

```
library(finitization)
qlog(4, 0.1, p = c(0.1, 0.1, 0.9))
qlog(4, 0.1, p = c(0.1, 0.1, 0.9), lower.tail = FALSE)
```

qnegbinom *The quantile function for the finitized Negative Binomial distribution.*

Description

qnegbinomial(n, q, k, p, lower.tail = TRUE, log.p = FALSE) computes the quantile(s) corresponding to the given probability(ies) for the finitized Negative Binomial distribution.

Usage

```
qnegbinom(n, q, k, p, lower.tail = TRUE, log.p = FALSE)
```

Arguments

n	The finitization order. An integer > 0.
q	The success probability for each trial ($q \in [0, 1]$).
k	The number of failures until the experiment is stopped, $k > 0$.
p	A numeric vector of probabilities. For <code>lower.tail = TRUE</code> (default), these are interpreted as $P(X \leq x)$; if <code>lower.tail = FALSE</code> , they represent $P(X > x)$.
<code>lower.tail</code>	Logical; if TRUE (default) the input probabilities are lower-tail probabilities, otherwise they are upper-tail probabilities.
<code>log.p</code>	Logical; if TRUE, the probabilities in p are assumed to be on the logarithmic scale.

Value

A numeric vector of quantiles corresponding to the input probabilities. Each quantile is the smallest integer for which the cumulative probability is at least the given probability.

Examples

```
library(finitization)
qnegbinom(n = 4, q = 0.12, k = 4, p = c(0.1, 0.5, 0.9))
qnegbinom(n = 4, q = 0.12, k = 4, p = c(0.1, 0.5, 0.9), lower.tail = FALSE)
```

qpois

The quantile function for the finitized Poisson distribution.

Description

`qpois(n, theta, p, lower.tail, log.p)` computes the quantile(s) corresponding to the given probability(ies) for the finitized Poisson distribution.

Usage

```
qpois(n, theta, p, lower.tail = TRUE, log.p = FALSE)
```

Arguments

n	The finitization order. An integer > 0.
theta	The parameter of the finitized Poisson distribution.
p	A numeric vector of probabilities. For <code>lower.tail = TRUE</code> (default), these are interpreted as $P(X \leq x)$; if <code>lower.tail = FALSE</code> they are interpreted as $P(X > x)$.
<code>lower.tail</code>	Logical; if TRUE (default) the input probabilities are lower-tail probabilities; otherwise, they are upper-tail.
<code>log.p</code>	Logical; if TRUE the probabilities in p are assumed to be on the logarithmic scale.

Value

A numeric vector of quantiles corresponding to the input probabilities. Each quantile is defined as the smallest integer x for which the cumulative probability is at least the given probability.

Examples

```
# For a finitized Poisson distribution with n = 4 and theta = 0.5:
qpois(n = 4, theta = 0.5, p = c(0.1, 0.5, 0.9))
qpois(n = 4, theta = 0.5, p = c(0.1, 0.5, 0.9), lower.tail = FALSE)
```

rbinom

Random values generation for the finitized Binomial distribution.

Description

rbinom(n, p, N, no) generates random values according to the finitized Binomial distribution.

Usage

```
rbinom(n, p, N, no)
```

Arguments

n	The finitization order. An integer > 1.
p	The success probability for each trial ($0 \leq p \leq 1$).
N	The number of trials.
no	The number of random values to be generated.

Value

An integer vector of length no, with random values drawn from the finitized Binomial distribution.

Examples

```
library(finitization)
rbinom(2, 0.5, 2, 10)
```

rlog	<i>Random values generation for the finitized Logarithmic distribution.</i>
------	---

Description

rlog(n, theta, no) generates random values according to the finitized Logarithmic distribution.

Usage

```
rlog(n, theta, no)
```

Arguments

n	The finitization order. It should be an integer > 1.
theta	The parameter of the Logarithmic distribution.
no	The number of random values to be generated.

Value

A vector of integers containing random values generated from the finitized Logarithmic distribution.

Examples

```
library(finitization)
rlog(2, 0.25, 10)
```

rneqbinom	<i>Random values generation for the finitized Negative Binomial distribution.</i>
-----------	---

Description

rneqbinom(n, q, k, no) generates random values according to the finitized Binomial distribution with parameters q, k.

Usage

```
rneqbinom(n, q, k, no)
```

Arguments

n	The finitization order. It should be an integer > 1.
q	The parameter of the finitized Negative Binomial distribution - the success probability for each trial. $q \in [0, 1]$
k	The number of failures until the experiment is stopped, $k > 0$.
no	The number of random values to be generated.

Value

rpois returns a vector of type `integer` containing random values generated according to the finitized Negative Binomial distribution. The number of values is given by the parameter `no`.

Examples

```
library(finitization)
rnegbinom(2, 0.15, 2, 10)
```

rpois	<i>Random values generation for the finitized Poisson distribution.</i>
-------	---

Description

rpois(`n`, `theta`, `no`) generates random values according to the finitized Poisson distribution with parameter `theta`.

Usage

```
rpois(n, theta, no)
```

Arguments

<code>n</code>	The finitization order. It should be an integer > 1.
<code>theta</code>	The parameter of the Poisson distribution.
<code>no</code>	The number of random values to be generated.

Value

rpois returns a vector of type `integer` containing random values generated according to the finitized Poisson distribution. The number of values is given by the parameter `no`.

Examples

```
library(finitization)
rpois(2, 0.5, 10)
```

Index

character, [14](#), [15](#)

dbinom, [2](#)

dlog, [3](#)

dnegbinom, [4](#)

dpois, [5](#)

getBinomialMFPS, [5](#)

getLogarithmicMFPS, [6](#)

getNegativeBinomialMFPS, [7](#)

getPoissonMFPS, [7](#)

integer, [20](#)

pbinom, [8](#)

plog, [9](#)

pnegbinom, [10](#)

ppois, [11](#)

printFinitizedBinomialDensity, [12](#)

printFinitizedLogarithmicDensity, [13](#)

printFinitizedNegativeBinomialDensity,
[13](#)

printFinitizedPoissonDensity, [14](#)

qbinom, [15](#)

qlog, [16](#)

qnegbinom, [16](#)

qpois, [17](#)

rbinom, [18](#)

rlog, [19](#)

rnegbinom, [19](#)

rpois, [20](#)