Package 'finitization'

April 28, 2025

Title Finitization Methods for Discrete Probability 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

NeedsCompilation yes

SystemRequirements CLN (>= 1.3.6, https://www.ginac.de/CLN/),
GiNaC (>= 1.8.3, https://www.ginac.de/Download.html),
GMP (>= 6.2.0, https://gmplib.org/download/gmp/),
GNU Make,
On Windows: precompiled static libraries and headers are provided under 'inst/libs/x64/' for 64-bit builds (see README for details).

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

VignetteBuilder knitr

LinkingTo Rcpp

Config/testthat/edition 3

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

R topics documented:

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

Index **19**

`dbinom` *The density for the finitized Binomial distribution.*

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 <= p <= 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 ... 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 ... 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 *The density for the finitized Poisson distribution.*

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 *Maximum feasible parameter space for the finitized Binomial distribution.*

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(4)
```

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` *The cumulative distribution function (CDF) for the finitized Binomial distribution.*

Description

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

Usage

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

Arguments

n	The finitization order. An integer > 0.
p	The success probability for each trial (0 <= p <= 1).
N	The number of trials.
val	A vector of values at which the CDF is computed. If NULL, a data frame containing all possible values (from 0 to n) and the corresponding cumulative probabilities is returned.
lower.tail	Logical; if TRUE (default) probabilities are computed as $P(X \leq x)$; if FALSE, as $P(X > x)$.
log.p	Logical; if TRUE, the cumulative probabilities are returned on the logarithmic scale.

Value

If val is provided, a numeric vector of cumulative probabilities (or their logarithms if log.p = TRUE) is returned. Otherwise, a data.frame with columns val and cdf is returned.

Examples

```
library(finitization)
pbinom(4, 0.5, 4, val = c(0, 2, 4))
pbinom(4, 0.5, 4, lower.tail = FALSE)
pbinom(4, 0.5, 4, log.p = TRUE)
```

---

plog *The cumulative distribution function (CDF) for the finitized Logarithmic distribution.*

Description

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

Usage

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

Arguments

n	The finitization order. An integer > 0.
theta	The parameter of the finitized Logarithmic distribution.
val	A vector with the values at which the CDF is computed. If NULL, a data frame containing all possible values (0, 1, ..., n) and their cumulative probabilities is returned.
log.p	Logical; if TRUE, the cumulative probabilities are returned on the logarithmic scale.
lower.tail	Logical; if TRUE (default) probabilities are computed as $P(X \leq x)$; if FALSE, as $P(X > x)$.

Value

If val is provided, a numeric vector of cumulative probabilities (or their logarithms if log.p = TRUE) is returned. Otherwise, a data.frame with columns val and cdf is returned.

Examples

```
library(finitization)
plog(4, 0.1, val = c(0, 2, 4))
plog(4, 0.1, log.p = TRUE)
plog(4, 0.1, val = c(0, 2, 4), lower.tail = FALSE)
```

---

pnegbinom *The cumulative distribution function (CDF) for the finitized Negative Binomial distribution.*

Description

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

Usage

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

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.
val	A vector with the values at which the CDF is computed. If NULL, a data frame containing all possible values (0, 1, ..., n) and their cumulative probabilities is returned.
log.p	Logical; if TRUE, the cumulative probabilities are returned on the logarithmic scale.
lower.tail	Logical; if TRUE (default) probabilities are calculated as $P(X \leq x)$; if FALSE, as $P(X > x)$.

Value

If val is provided, a numeric vector of cumulative probabilities (or their logarithms if log.p = TRUE) is returned. Otherwise, a data.frame with columns val and cdf is returned.

Examples

```
library(finitization)
pnegbinom(4, 0.12, 4, val = c(0, 2, 4))
pnegbinom(4, 0.12, 4, log.p = TRUE)
pnegbinom(4, 0.12, 4, val = c(0, 2, 4), lower.tail = FALSE)
```

ppois *The cumulative distribution function (CDF) for the finitized Poisson distribution.*

Description

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

Usage

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

Arguments

n	The finitization order. An integer > 0.
theta	The parameter of the finitized Poisson distribution.
val	A vector of values at which the CDF is computed. If NULL, a data frame containing all possible values (0, 1, ..., n) and their cumulative probabilities is returned.
log.p	Logical; if TRUE, the cumulative probabilities are returned on the logarithmic scale.
lower.tail	Logical; if TRUE (default) probabilities are calculated as $P(X \leq x)$; if FALSE, as $P(X > x)$.

Value

If val is provided, a numeric vector of cumulative probabilities (or their logarithms if log.p = TRUE) is returned. Otherwise, a data.frame with columns val and cdf is returned.

Examples

```
# For a finitized Poisson distribution with n = 4 and theta = 0.5:
ppois(n = 4, theta = 0.5, val = c(0, 2, 4))
ppois(n = 4, theta = 0.5, log.p = TRUE)
ppois(n = 4, theta = 0.5, val = c(0, 2, 4), lower.tail = FALSE)
```

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

n	The finitization order. It should be an integer > 0.
val	The value for which the pdf is printed. If NULL, the pdf for all values (0 ... n) is printed.
latex	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 .. 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 .. 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

n	The finitization order. An integer > 0.
p	The success probability for each trial (0 <= p <= 1).
N	The number of trials.
prob	A vector of probabilities (or log-probabilities if log.p = TRUE) for which the quantiles are computed. Each probability must be between 0 and 1.
lower.tail	Logical; if TRUE (default) the input probabilities are interpreted as $P(X \leq x)$; if FALSE, they are interpreted as $P(X > x)$.
log.p	Logical; if TRUE, the probabilities in prob 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 \(x\) (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 lower.tail = TRUE (default), these are interpreted as $P(X \leq x)$; if lower.tail = FALSE, they represent $P(X > x)$.
lower.tail	Logical; if TRUE (default) the input probabilities are lower-tail probabilities, otherwise they are upper-tail probabilities.
log.p	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 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, they are upper-tail.
log.p	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 <= p <= 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 — *Random values generation for the finitized Logarithmic distribution.*

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)
```

rnegbinom — *Random values generation for the finitized Negative Binomial distribution.*

Description

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

Usage

```
rnegbinom(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, 0.5, 2, 10)
```

---

rpois *Random values generation for the finitized Poisson distribution.*

Description

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

Usage

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

Arguments

n	The finitization order. It should be an integer > 1.
theta	The parameter of the Poisson distribution.
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 Poisson distribution. The number of values is given by the parameter no.

Examples

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

Index

character, *12, 13*

dbinom, 2
dlog, 3
dnegbinom, 4
dpois, 4

getBinomialMFPS, 5
getLogarithmicMFPS, 6
getNegativeBinomialMFPS, 6
getPoissonMFPS, 7

integer, *18*

pbinom, 7
plog, 8
pnegbinom, 9
ppois, 10
printFinitizedBinomialDensity, 10
printFinitizedLogarithmicDensity, 11
printFinitizedNegativeBinomialDensity, 12
printFinitizedPoissonDensity, 12

qbinom, 13
qlog, 14
qnegbinom, 15
qpois, 15

rbinom, 16
rlog, 17
rnegbinom, 17
rpois, 18