Package ‘Delaporte’

January 27, 2015

Type Package

Title Statistical Functions for the Delaporte Distribution

Version 2.2-2

Date 2014-12-04

Description Provides probability mass, distribution, quantile, random variate generation, and method of moments parameter estimation functions for the Delaporte distribution.

License GPL (>= 2) | LGPL (>= 3)

URL https://bitbucket.org/aadler/delaporte

BugReports https://bitbucket.org/aadler/delaporte/issues

Imports Rcpp (>= 0.11.0)

LinkingTo Rcpp

Suggests testthat

ByteCompile yes

NeedsCompilation yes

Author Avraham Adler [aut, cph, cre]

Maintainer Avraham Adler <Avraham.Adler@gmail.com>

Repository CRAN

Date/Publication 2014-12-04 15:32:43

R topics documented:

Delaporte-package — *Statistical Functions for the Delaporte Distribution*

Description

Provides probability mass, distribution, quantile, random number generation, and method of moments estimation functions for the Delaporte distribution.

Details

Package:	Delaporte
Type:	Package
Version:	2.2-2
Date:	2014-12-04
License:	GPL (>= 2) | LGPL (>= 3)

Provides probability mass, distribution, quantile, random number generation, and method of moments estimation functions for the Delaporte distribution. It is used similarly to the distributions in `stats`.

Author(s)

Avraham Adler

Maintainer: Avraham Adler <Avraham.Adler@gmail.com>

Delaporte — *The Delaporte Distribution*

Description

Density, distribution, quantile, random variate generation, and method of moments parameter estimation functions for the Delaporte distribution with parameters `alpha`, `beta`, and `lambda`.

Usage

```
ddelap(x, alpha, beta, lambda, log = FALSE)
pdelap(q, alpha, beta, lambda, lower.tail = TRUE, log.p = FALSE)
qdelap(p, alpha, beta, lambda, lower.tail = TRUE, log.p = FALSE, exact = TRUE)
rdelap(n, alpha, beta, lambda, exact = TRUE)

MoMdelap(x)
```

Arguments

`x`	vector of (non-negative integer) quantiles.
`q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`alpha`	vector of alpha parameters of the gamma portion of the Delaporte distribution. Must be strictly positive, but need not be integer.
`beta`	vector of beta parameters of the gamma portion of the Delaporte distribution. Must be strictly positive, but need not be integer.
`lambda`	vector of lambda parameters of the Poisson portion of the Delaporte distribution. Must be strictly positive, but need not be integer.
`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, $P[X > x]$.
`exact`	logical; if TRUE uses double summation to generate quantiles or random variates. Otherwise uses Poisson-negative binomial approximation.

Details

The Delaporte distribution with parameters `alpha`, `beta`, and `lambda` is a discrete probability distribution which can be considered the convolution of a negative binomial distribution with a Poisson distribution. Alternatively, it can be considered a counting distribution with both Poisson and negative binomial components. The Delaporte's probability mass function, called via `ddelap`, is:

$$p(n) = \sum_{i=0}^{n} \frac{\Gamma(\alpha + i)\beta^i \lambda^{n-i} e^{-\lambda}}{\Gamma(\alpha) i! (1 + \beta)^{\alpha+i} (n - i)!}$$

for $n = 0, 1, 2, \ldots; \alpha, \beta, \lambda > 0$.

Its cumulative distribution function, `pdelap`, is calculated through double summation:

$$CDF(n) = \sum_{j=0}^{n} \sum_{i=0}^{j} \frac{\Gamma(\alpha + i)\beta^i \lambda^{j-i} e^{-\lambda}}{\Gamma(\alpha) i! (1 + \beta)^{\alpha+i} (j - i)!}$$

for $n = 0, 1, 2, \ldots; \alpha, \beta, \lambda > 0$. For both the probability mass and distribution calculations, if a non-integer value is passed into the function, it is rounded up to the next integer. If only singleton values for the parameters are passed in, the function uses the shortcut of identifying the largest value passed to it, computes a vector of CDF values for all integers up to and including that value, and having the remaining results read from this vector. This requires only one double summation instead of `length(q)` such summations. If at least one of the parameters is itself a vector of length greater than 1, the function has to build the double summation for each entry in `q`.

The quantile function, `qdelap`, has two versions. When `exact=TRUE`, the function builds a CDF vector and the first value for which the CDF is greater than or equal to `q` is returned as the quantile. While this procedure is accurate, for sufficiently large α, β, or λ it can take a very long time . Therefore, when `exact=FALSE`, the function takes advantage of the Delaporte's definition as a counting distribution with both a Poisson and negative binomial component. It generates up to 10^7 variates from a negative binomial distribution with shape α and scale β (size = α, mean = $\alpha\beta$), and the same number of variates from a Poisson distribution with the mean λ. It then sums the two sets of variates and calls the `quantile` function on the result. The "exact" method is always more accurate and is also significantly faster for reasonable values of the parameters. Ad-hoc testing indicates that the "exact" method should always be used until $\alpha\beta + \lambda \approx 5000$. Both versions return `NaN` for quantiles < 0, 0 for quantiles $= 0$, and `Inf` for quantiles ≥ 1.

The random variate generator, `rdelap`, also has multiple versions. When `exact=TRUE`, it uses inversion by creating a vector of `n` uniformly distributed random variates between 0 and 1. If all the parameters are singletons, a single CDF vector is constructed as per the quantile function, and the entries corresponding to the uniform variates are read off of the constructed vector. If the parameters are themselves vectors, then it passes the entire uniform variate vector to `qdelap`, which is slower. When `exact=FALSE`, regardless of the length of the parameters, the larger of `n` or 10^7 variates from both a Poisson and negative binomial with the appropriate parameters are generated and summed. If `n` $< 10^7$, sampling with replacement is used to generate the `n` samples from the pool of 10^7 pseudo-Delaporte variates.

`MoMdelap` uses the definition of the Delaporte's mean, variance, and skew to calculate the method of moments estimates of α, β, and λ, which it returns as a numeric vector. This estimate is also a reasonable starting point for maximum likelihood estimation using nonlinear optimizers such as `optim` or `nloptr`. If the data is clustered near 0, there are times when method of moments would result in a negative parameter (usually λ). In these cases `MoMdelap` will throw an error.

Value

`ddelap` gives the probability mass function, `pdelap` gives the cumulative distribution function, `qdelap` gives the quantile function, and `rdelap` generates random deviates.

Invalid quantiles passed to `qdelap` will result in return values of `NaN` or `Inf` as appropriate.

The length of the result is determined by `x` for `ddelap`, `q` for `pdelap`, `p` for `qdelap`, and `n` for `rdelap`. The distributional parameters (α, β, λ) are recycled as necessary to the length of the result. When using the `lower.tail = FALSE` or `log / log.p = TRUE` options, some accuracy may be lost at knot points or the tail ends of the distributions due to the limitations of floating point representation.

Author(s)

Avraham Adler <Avraham.Adler@gmail.com>

References

Johnson, Norman Lloyd, Kemp, Adrienne W. and Kotz, Samuel (2005) *Univariate discrete distributions* (Third ed.). John Wiley & Sons. pp. 241–242. ISBN 978-0-471-27246-5.

Vose, David (2008) *Risk analysis: a quantitative guide* (Third, illustrated ed.). John Wiley & Sons. pp. 618–619 ISBN 978-0-470-51284-5

See Also

Distributions for standard distributions, including `dnbinom` for the negative binomial distribution and `dpois` for the Poisson distribution.

Examples

```
## Density and distribution
A <- c(0, seq_len(50))
PMF <- ddelap(A, alpha = 3, beta = 4, lambda = 10)
CDF <- pdelap(A, alpha = 3, beta = 4, lambda = 10)

##Quantile
A <- seq(0,.95, .05)
qdelap(A, alpha = 3, beta = 4, lambda = 10)
A <- c(-1, A, 1, 2)
qdelap(A, alpha = 3, beta = 4, lambda = 10)

## Compare a Poisson, negative binomial, and three Delaporte distributions with the same mean:
P <- rpois(25000, 25)                                  ## Will have the tightest spread
DP1 <- rdelap(10000, alpha = 2, beta = 2, lambda = 21) ## Close to the Poisson
DP2 <- rdelap(10000, alpha = 3, beta = 4, lambda = 13) ## In between
DP3 <- rdelap(10000, alpha = 4, beta = 5, lambda = 5)  ## Close to the Negative Binomial
NB <- rnbinom(10000, size = 5, mu = 25)                ## Will have the widest spread
mean(P);mean(NB);mean(DP1);mean(DP2);mean(DP3)         ## Means should all be near 25
MoMdelap(DP1);MoMdelap(DP2);MoMdelap(DP3)           ## Estimates should be close to originals
```

Index