Package ‘Delaporte’

October 2, 2023

Type Package
Title Statistical Functions for the Delaporte Distribution
Version 8.3.0
Date 2023-10-02
Description Provides probability mass, distribution, quantile, random-variate generation, and method-of-moments parameter-estimation functions for the Delaporte distribution with parameterization based on Vose (2008) <isbn:9780470512845>. The Delaporte 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. It has been studied in actuarial science as a frequency distribution which has more variability than the Poisson, but less than the negative binomial.

License BSD_2_clause + file LICENSE
URL https://github.com/aadler/Delaporte
BugReports https://github.com/aadler/Delaporte/issues
Depends R (>= 3.6.0)
Imports stats, parallel
Suggests covr, tinytest
ByteCompile yes
NeedsCompilation yes
UseLTO yes
SystemRequirements A version of Fortran supporting the LOG_GAMMA Intrinsic
Author Avraham Adler [aut, cph, cre] (<https://orcid.org/0000-0002-3039-0703>)
Maintainer Avraham Adler <Avraham.Adler@gmail.com>
Repository CRAN
Date/Publication 2023-10-02 18:40:07 UTC
R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delaporte-package</td>
<td>2</td>
</tr>
<tr>
<td>Delaporte</td>
<td>3</td>
</tr>
<tr>
<td>Delaporte-defunct</td>
<td>7</td>
</tr>
<tr>
<td>setDelapThreads</td>
<td>7</td>
</tr>
</tbody>
</table>

Index

---

**Description**

Provides probability mass, distribution, quantile, random-variate generation, and method-of-moments parameter-estimation functions for the Delaporte distribution with parameterization based on Vose (2008) <isbn:9780470512845>. The Delaporte 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. It has been studied in actuarial science as a frequency distribution which has more variability than the Poisson, but less than the negative binomial.

**Details**

The DESCRIPTION file:

```r
Package: Delaporte
Type: Package
Title: Statistical Functions for the Delaporte Distribution
Version: 8.3.0
Date: 2023-10-02
Authors@R: c(person(given="Avraham", family="Adler", role=c("aut", "cph", "cre"), email="Avraham.Adler@gmail.com", comment = c(ORCID = "0000-0002-3039-0703"))
Description: Provides probability mass, distribution, quantile, random-variate generation, and method-of-moments parameter-estimation functions for the Delaporte distribution with parameterization based on Vose (2008) <isbn:9780470512845>. The Delaporte 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. It has been studied in actuarial science as a frequency distribution which has more variability than the Poisson, but less than the negative binomial.
License: BSD_2_clause + file LICENSE
URL: https://github.com/aadler/Delaporte
BugReports: https://github.com/aadler/Delaporte/issues
Depends: R (>= 3.6.0)
Imports: stats, parallel
Suggests: covr, tinytest
ByteCompile: yes
NeedsCompilation: yes
UseLTO: yes
SystemRequirements: A version of Fortran supporting the LOG_GAMMA Intrinsic
Author: Avraham Adler [aut, cph, cre] (<https://orcid.org/0000-0002-3039-0703>)
Maintainer: Avraham Adler <Avraham.Adler@gmail.com>
Archs: x64
```
Index of help topics:

- Delaporte: The Delaporte Distribution
- Delaporte-defunct: Defunct functions in package 'Delaporte'
- Delaporte-package: Statistical Functions for the Delaporte Distribution
- setDelapThreads: Set or get the number of OpenMP threads 'Delaporte' should use

Author(s)

Avraham Adler [aut, cph, cre] (<https://orcid.org/0000-0002-3039-0703>)
Maintainer: Avraham Adler <Avraham.Adler@gmail.com>

---

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

```r
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, type = 2L)
```

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, $P [X > x]$. 
Delaporte

exact logical; if TRUE uses double summation to generate quantiles or random variates. Otherwise uses Poisson-negative binomial approximation.

type integer; 1L will return g1, 2L will return G1, and 3L will return b1, as per skewness.

Details

Definition: 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:

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

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

If an element of x is not integer, the result of ddelap is zero with a warning.

The Delaporte’s cumulative distribution function, pdelap, is calculated through double summation:

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

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

If only singleton values for the parameters are passed in, the function uses a shortcut. It identifies the largest value passed to it, computes a vector of CDF values for all integers up to and including that value, and reads the remaining results from this vector. This requires only one double summation instead of $\text{length}(q)$ 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 $q$.

Distributional Functions:

Density and Distribution: ddelap will return 0 for all values $> 2^{31}$ whereas pdelap will not run at all, due to the limitations of integer representation. Also, for values $> 2^{15}$, pdelap will ask for positive input from the user to continue, as otherwise, depending on the parameters, the function can take hours to complete given its double-summation nature.

Quantile: The quantile function, qdelap, is right continuous: qdelap(q, alpha, beta, lambda) is the smallest integer $x$ such that $P(X \leq x) \geq q$. This function 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 $\alpha$, $\beta$, or $\lambda$ it can take a very long time. Therefore, when dealing with singleton parameters, exact = FALSE can be passed to take advantage of the Delaporte’s definition as a counting distribution with both a Poisson and a negative binomial component. Based on Karlis & Xekalaki (2005) it will generate $n$ gamma variates $\Gamma$ with shape $\alpha$ and scale $\beta$ and then $n$ pseudo-Delaporte variates as Poisson random variables with parameter $\lambda + \Gamma$, finally calling the quantile function on the result. The “exact” method is always more accurate and is also significantly faster for reasonable values of the parameters. Also, the “exact” method must be used when passing
parameter vectors, as the pooling would become intractable. Ad-hoc testing indicates that the "exact" method should be used until $\alpha \beta + \lambda \approx 2500$. Both versions return NaN for quantiles $< 0$, 0 for quantiles $= 0$, and Inf for quantiles $\geq 1$.

**Random Variate Generation:** 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, it then passes the entire uniform variate vector to qdelap, which is slower. When exact = FALSE, regardless of the length of the parameters, it generates $n$ gamma variates $\Gamma$ with shape $\alpha$ and scale $\beta$ and then $n$ pseudo-Delaporte variates as Poisson random variables with parameter $\lambda + \Gamma$. As there is no pooling, each individual random variate reflects the parameter triplet which generated it. The non-inversion method is usually faster.

**Method of Moments Fitting:** MoMdelap uses the definition of the Delaporte’s mean, variance, and skew to calculate the method of moments estimates of $\alpha$, $\beta$, and $\lambda$, 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 non-positive parameter. In these cases MoMdelap will throw an error. For the sample skew, the user has the choice to select $g_1$, $G_1$, or $b_1$ as defined in Joanes & Gill (1997) and found in skewness. The selection defaults to option 2, $G_1$, which Joanes & Gill found to have the least mean-square error for non-normal distributions.

**Value**

ddelap gives the probability mass function, pdelap gives the cumulative distribution function, qdelap gives the quantile function, and rdelap generates random deviates. Values close to 0 (less than or equal to machine epsilon) for $\alpha$, $\beta$ or $\lambda$ will return NaN for that particular entry. Proper triplets within a set of vectors should still return valid values. For the approximate versions of qdelap and rdelap, having a value close to 0 will trip an error, sending the user to the exact version which currently properly handles vector-based inputs which contain 0.

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 ($\alpha$, $\beta$, $\lambda$) 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.

MoMdelap returns a triplet comprising a method-of-moments based estimate of $\alpha$, $\beta$, and $\lambda$.

**Author(s)**

Avraham Adler <Avraham.Adler@gmail.com>

**References**


See Also

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

Examples

```r
## 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

## Not run:
plot(density(P), col = "black", lwd = 2, main = "Distribution Comparison",
     xlab = "Value", xlim = c(0, 80))
lines(density(DP1), col = "blue", lwd = 2)
lines(density(DP2), col = "green3", lwd = 2)
lines(density(DP3), col = "orange3", lwd = 2)
lines(density(NB), col = "red", lwd = 2)
legend(x = "topright", legend = c("Poisson (l=25)", "DP (a=2, b=2, l=21)",
                                 "DP (a=3, b=4, l=13)", "DP (a=5, b=5)", "NegBinom (a=5, b=5)"),
            col = c("black", "blue", "green3", "orange3", "red"), lwd = 2)
```

## End(Not run)
Description
These functions are defunct and no longer available.

Details
The following functions are defunct; use the replacement indicated below:

- As of version 5.0.0, the old = TRUE method for the qdelap and rdelap functions is defunct and has been removed.

setDelapThreads

Description
Set or get number OpenMP threads to be used by Delaporte functions which are parallelized. These include ddelap, pdelap, qdelap, and rdelap.

Usage
setDelapThreads(n)
getDelapThreads()

Arguments
n integer: maximum number of threads to be used.

Details
For systems which return valid values for detectCores, the maximum number of threads will be capped at that value.

Value
getDelapThreads returns an integer representing the maximum number of allowed threads.
setDelapThreads invisibly returns the integer passed to it.

Author(s)
Avraham Adler <Avraham.Adler@gmail.com>
See Also

detectCores in parallel package.

Examples

getDelapThreads()
setDelapThreads(2L)
getDelapThreads()
Index

* data
  setDelapThreads, 7
* distribution
  Delaporte, 3
  Delaporte-package, 2
* package
  Delaporte-package, 2

ddelap (Delaporte), 3
Delaporte, 3
delaporte (Delaporte), 3
Delaporte-defunct, 7
Delaporte-package, 2
detectCores, 7, 8
Distributions, 6
dnbinom, 6
dpois, 6
getDelapThreads (setDelapThreads), 7
MoMdelap (Delaporte), 3
nloptr, 5
openMP (setDelapThreads), 7
openmp (setDelapThreads), 7
optim, 5
pdelap (Delaporte), 3
qdelap, 7
qdelap (Delaporte), 3
quantile, 4
rdelap, 7
rdelap (Delaporte), 3
setDelapThreads, 7
skewness, 4–6