Package: DiscreteWeibull (via r-universe)

October 14, 2024

Type Package

Title Discrete Weibull Distributions (Type 1 and 3)

Version 1.1

Date 2015-10-15

Author Alessandro Barbiero

Maintainer Alessandro Barbiero <alessandro.barbiero@unimi.it>

Description Probability mass function, distribution function, quantile function, random generation and parameter estimation for the type I and III discrete Weibull distributions.

License GPL

LazyLoad yes

Depends Rsolnp, stats

NeedsCompilation no

Repository CRAN

Date/Publication 2015-10-18 01:21:16

Contents

DiscreteWeibull-package
Discrete Weibull Distributions (Type 1 and 3)

Description

Probability mass function, distribution function, quantile function, random generation and parameter estimation for the type I and III discrete Weibull distributions. In the present version, some modifications have been made on the functions in Edweibull for the computation of the moments of the type I discrete Weibull distribution.

Details

Package:	DiscreteWeibull
Type:	Package
Version:	1.1
Date:	2015-10-17
License:	GPL
LazyLoad:	yes
Depends:	Rsolnp

Author(s)

Alessandro Barbiero <alessandro.barbiero@unimi.it>

References

T. Nakagawa and S. Osaki (1975) The discrete Weibull distribution, IEEE Transactions on Reliability, 24(5), pp. 300-301

D.N.P. Murthy, M. Xie, R Jiang (2004) Weibull models, John Wiley & Sons: Hoboken, New Jersey

W.J. Padgett and J.D. Spurrier (1985) Discrete failure models, IEEE Transactions on Reliability, 34(3), pp. 253-256

H. Rinne (2008) The Weibull Distribution: A Handbook. CRC Press: Boca Raton, Florida

M. S. A. Khan, A. Khalique, and A. M. Abouammoh (1989) On estimating parameters in a discrete Weibull distribution, IEEE Transactions on Reliability, 38(3), pp. 348-350

See Also

ddweibull, estdweibull, Edweibull, ddweibull3, estdweibull3, Edweibull3

`Discrete Weibull (Type 1)` *The type 1 discrete Weibull distribution*

Description

Probability mass function, distribution function, quantile function and random generation for the discrete Weibull distribution with parameters q and β

Usage

```
ddweibull(x, q, beta, zero = FALSE)
pdweibull(x, q, beta, zero = FALSE)
qdweibull(p, q, beta, zero = FALSE)
rdweibull(n, q, beta, zero = FALSE)
```

Arguments

`x`	vector of quantiles
`p`	vector of probabilities
`q`	first parameter
`beta`	second parameter
`zero`	`TRUE`, if the support contains 0; `FALSE` otherwise
`n`	sample size

Details

The discrete Weibull distribution has probability mass function given by $P(X = x; q, \beta) = q^{(x-1)^\beta} - q^{x^\beta}$, $x = 1, 2, 3, \ldots$, if `zero=FALSE`; or $P(X = x; q, \beta) = q^{x^\beta} - q^{(x+1)^\beta}$, $x = 0, 1, 2, \ldots$, if `zero=TRUE`. The cumulative distribution function is $F(x; q, \beta) = 1 - q^{x^\beta}$ if `zero=FALSE`; $F(x; q, \beta) = 1 - q^{(x+1)^\beta}$ otherwise

Value

`ddweibull` gives the probability function, `pdweibull` gives the distribution function, `qdweibull` gives the quantile function, and `rdweibull` generates random values.

Author(s)

Alessandro Barbiero

Examples

```
# Ex.1
x <- 1:10
q <- 0.6
beta <- 0.8
ddweibull(x, q, beta)
t <- qdweibull(0.99, q, beta)
t
pdweibull(t, q, beta)
#
x <- 0:10
ddweibull(x, q, beta, zero=TRUE)
t <- qdweibull(0.99, q, beta, zero=TRUE)
t
pdweibull(t, q, beta, zero=TRUE)

# Ex.2
q <- 0.4
beta <- 0.7
n <- 100
x <- rdweibull(n, q, beta)
tabulate(x)/sum(tabulate(x))
y <- 1:round(max(x))
# compare with
ddweibull(y, q, beta)
```

`Discrete Weibull (Type 3)` *The type 3 discrete Weibull distribution*

Description

Probability mass function, distribution function, quantile function, random generation, and hazard function for the type 3 discrete Weibull distribution with parameters c and β

Usage

```
ddweibull3(x, c, beta)
pdweibull3(x, c, beta)
qdweibull3(p, c, beta)
rdweibull3(n, c, beta)
hdweibull3(x, c, beta)
```

Arguments

x	vector of values/quantiles
p	vector of probabilities
c	first parameter

`beta`	second parameter
`n`	sample size

Details

The type 3 discrete Weibull distribution is characterized by the following cumulative distribution function: $F(x; c, \beta) = 1 - \exp(-c \sum_{j=0}^{x+1} j^{\beta})$, for $x = 0, 1, 2, \ldots$, with $c > 0$ and $\beta >= -1$.

Value

`ddweibull3` gives the probability function, `pdweibull3` gives the distribution function, `qdweibull3` gives the quantile function, `hdweibull3` gives the hazard function, and `rdweibull` generates random values.

Author(s)

Alessandro Barbiero

Examples

```
# ddweibull3
x <- 0:10
c <- 0.3
beta <- 0.75
p <- ddweibull3(x, c, beta)
p
plot(x, p, type="b", ylab=expression(P(X)==x))
# pdweibull3
x <- 0:10
c <- 0.5
beta <- 0.5
p <- pdweibull3(x, c, beta)
p
cumsum(ddweibull3(x, c, beta))
plot(x, p, type="s", ylab=expression(P(X<=x)))
# qdweibull3
p <- c(1:9)/10
p
c <- 0.1
beta <- 0.5
qdweibull3(p, c, beta)
pdweibull3(10, c, beta)
pdweibull3(9, c, beta)
# rdweibull3
n <- 20
c <- 0.25
beta <- -0.25
x <- rdweibull3(n, c, beta)
x
beta <- 0
x <- rdweibull3(n, c, beta)
x
```

```
beta <- 0.25
x <- rdweibull3(n, c, beta)
x
n <- 1000
x <- rdweibull3(n, c, beta)
obs <- c(sum(x==0), tabulate(x))
obs <- obs/sum(obs)
theo <- ddweibull3(min(x):max(x), c, beta)
barplot(rbind(obs, theo), beside=TRUE, names.arg=min(x):max(x),
ylab="relative frequency/probability", col=1:2)
legend(24, 0.1, c("observed", "theoretical"), pch=15, col=1:2)
#hdweibull3
x <- 0:15
c <- 0.5
hn<-hdweibull3(x, c, beta = -0.5)
h0<-hdweibull3(x, c, beta = 0)
hp<-hdweibull3(x, c, beta = 0.5)
plot(x, hn, type="b", ylim = c(0, 1), ylab="hazard rate")
points(x, h0, type = "b", col=2)
points(x, hp, type = "b", col=3)
legend(11, 0.5, c("beta<0", "beta=0", "beta>0"), col=1:3, pch=21)
```

`Edweibull` *Expected values*

Description

First and second order moments, variance and expected value of the reciprocal for the type 1 discrete Weibull distribution

Usage

```
Edweibull(q, beta, eps = 1e-04, nmax = 1000, zero = FALSE)
E2dweibull(q, beta, eps = 1e-04, nmax = 1000, zero = FALSE)
Vdweibull(q, beta, eps = 1e-04, nmax = 1000, zero = FALSE)
ERdweibull(q, beta, eps = 1e-04, nmax = 1000)
```

Arguments

`q`	first parameter
`beta`	second parameter
`eps`	error threshold for the numerical computation of the expected value
`nmax`	maximum value considered for the numerical approximate computation of the expected value;
`zero`	`TRUE`, if the support contains 0; `FALSE` otherwise

Details

The expected value is numerically computed considering a truncated support: integer values smaller than or equal to $2F^{-1}(1 - eps; q, \beta)$ are considered, where F^{-1} is the inverse of the cumulative distribution function (implemented by the function `qdweibull`). However, if such value is greater than `nmax`, the expected value is computed recalling the formula of the expected value of the corresponding continuous Weibull distribution (see the reference), adding 0.5. Similar arguments apply to the other moments.

Value

the (approximate) expected values of the discrete Weibull distribution: `Edweibull` gives the first order moment, `E2dweibull` the second order moment, `Vdweibull` the variance, `ERdweibull` the expected value of the reciprocal (only if `zero` is `FALSE`)

Author(s)

Alessandro Barbiero

References

M. S. A. Khan, A. Khalique, and A. M. Abouammoh (1989) On estimating parameters in a discrete Weibull distribution, IEEE Transactions on Reliability, 38(3), pp. 348-350

Examples

```
q <- 0.75
beta <- 1.25
Edweibull(q, beta)
E2dweibull(q, beta)
Vdweibull(q, beta)
ERdweibull(q, beta)
# if beta=0.75...
beta <- 0.75
Edweibull(q, beta)
Edweibull(q, beta, nmax=100)
# here above, the approximation through the continuous model intervenes
# if beta=1...
beta <- 1
Edweibull(q, beta)
# which equals...
1/(1-q)
```

`Edweibull3` *Expected values*

Description

First and second order moments for the type 3 discrete Weibull distribution

Usage

```
Edweibull3(c, beta, eps = 1e-04)
E2dweibull3(c, beta, eps = 1e-04)
```

Arguments

c	first parameter
beta	second parameter
eps	error threshold for the numerical computation of the expected value

Details

The expected values are numerically computed considering a truncated support: integer values smaller than or equal to $2F^{-1}(1 - eps; c, \beta))$, where F^{-1} is the inverse of the cumulative distribution function (implemented by the function `qdweibull3`)

Value

the (approximate) expected values of the discrete Weibull distribution: `Edweibull3` gives the first order moment, `E2dweibull3` the second order moment

Author(s)

Alessandro Barbiero

Examples

```
c <- 0.4
beta <- 0.25
Edweibull3(c,beta)
c <- 0.4
beta <- -0.75
Edweibull3(c, beta) # may require too much time
Edweibull3(c, beta, eps=0.001) # try with a smaller eps->worse approximation
c <- rep(0.1, 11)
beta <- (0:10)/10
Edweibull3(c, beta)
c <- rep(0.5, 11)
beta <- (-5:5)/10
Edweibull3(c,beta)
# E2dweibull3
c <- 0.4
beta <- 0.25
E2dweibull3(c, beta)
c <- rep(0.1, 11)
beta <- (0:10)/10
Edweibull3(c, beta)
c <- rep(0.8, 11)
beta <- (-5:5)/11
E2dweibull3(c, beta)
```

estdweibull *Estimation of parameters*

Description

Estimation of the parameters of the type 1 discrete Weibull distribution

Usage

```
estdweibull(x, method = "ML", zero = FALSE, eps = 1e-04, nmax=1000)
```

Arguments

`x`	the vector of sample values
`method`	`"ML"` for the maximum likelihood method; `"M"` for the method of moments; `"P"` for the method of proportions
`zero`	`TRUE`, if the support contains 0; `FALSE` otherwise
`eps`	error threshold for the computation of the moments of the distribution
`nmax`	maximum value considered for the numerical computation of the expected value

Value

the vector of the estimates of q and β

Author(s)

Alessandro Barbiero

See Also

`ddweibull`

Examples

```
# Ex1
n <- 10
q <- 0.5
beta <- 0.8
x <- rdweibull(n, q, beta)
estdweibull(x, "ML") # maximum likelihood method
# it may return some harmless warnings
# that depend on the optimization function used in the maximization routine
estdweibull(x, "M") # method of moments
estdweibull(x, "P") # method of proportion
# the estimates provided by the three methods may be quite different
# from the true values... and to each other
# change the sample size
n <- 50
```

```
q <- 0.5
beta <- 0.8
x <- rdweibull(n, q, beta)
estdweibull(x, "ML") # maximum likelihood method
estdweibull(x, "M") # method of moments
estdweibull(x, "P") # method of proportion
# the estimates should be (on average) closer to the true values
# ...and to each other

# When the estimation methods fail...
# Ex2
# only 1s and 2s
x <- c(1,1,1,1,1,1,2,2,2,2)
estdweibull(x, "ML") # fails!
estdweibull(x, "M") # fails!
estdweibull(x, "P") # fails!

# Ex3
# no 1s
x <- c(2,2,3,4,5,5,5,6,6,8,10)
estdweibull(x, "ML") # works
estdweibull(x, "M") # works
estdweibull(x, "P") # fails!
```

`estdweibull3` *Estimation of parameters*

Description

Estimation of the parameters of the type 3 discrete Weibull distribution

Usage

```
estdweibull3(x, method = "P", eps = 1e-04)
```

Arguments

`x`	the vector of sample values
`method`	`"ML"` for the maximum likelihood method; `"M"` for the method of moments; `"P"` for the method of proportions
`eps`	error threshold for the computation of the moments of the distribution

Value

the vector of the estimates of c and β

Author(s)

Alessandro Barbiero

See Also

ddweibull3

Examples

```
# Ex1
x <- c(0,0,0,0,0,1,1,1,1,1,1,1,2,2,2,2,3,3,4,6)
estdweibull3(x, "P")
estdweibull3(x, "ML")
estdweibull3(x, "M")
# Ex 2
n <- 20
c <- 1/3
beta <- 2/3
x <- rdweibull3(n, c, beta)
estdweibull3(x, "P")
par <- estdweibull3(x, "ML")
par
-loglikedw3(par, x)
par <- estdweibull3(x, "M")
par
lossdw3(par, x)
n <- 50
x <- rdweibull3(n, c, beta)
estdweibull3(x, "P")
estdweibull3(x, "ML")
estdweibull3(x, "M")
n <- 100
x <- rdweibull3(n, c, beta)
estdweibull3(x, "P")
estdweibull3(x, "ML")
estdweibull3(x, "M")
# Ex 3: a piece of simulation study
nSim <- 50
n <- 50
c <- 0.2
beta <- 0.7
par <- matrix(0, nSim, 2)
for(i in 1:nSim)
{
x <- rdweibull3(n, c, beta)
par[i,] <- estdweibull3(x, "ML")
}
op <- par(mfrow = c(1,2))
boxplot(par[,1], xlab=expression(hat(c)[ML]))
abline(h = c)
boxplot(par[,2], xlab=expression(hat(beta)[ML]))
abline(h = beta)
op <- par()
```

loglikedw *Loglikelihood function*

Description

Loglikelihood function (changed in sign) for the type 1 discrete Weibull distribution

Usage

```
loglikedw(par, x, zero = FALSE)
```

Arguments

par	the vector of parameters, q and β
x	the vector of sample values
zero	TRUE, if the support contains 0; FALSE otherwise

Value

the value of the loglikelihood function (changed in sign) for the observed sample x under the parameters par

Author(s)

Alessandro Barbiero

See Also

estdweibull

Examples

```
x <- c(1,1,1,2,2,2,2,2,2,3,4,4,5,6,8)
-loglikedw(c(0.8, 1), x) # loglikelihood function for q=0.8 and beta=1
-loglikedw(c(0.4, 2), x) # loglikelihood function for q=0.4 and beta=2
par <- estdweibull(x, "ML")# parameter estimates derived by the ML method
par
-loglikedw(par, x) # the maximum value of the loglikelihood function
```

`loglikedw3` *Loglikelihood function*

Description

Loglikelihood function (changed in sign) for the type 3 discrete Weibull distribution

Usage

```
loglikedw3(par, x)
```

Arguments

`par`	the vector of parameters, c and β
`x`	the vector of sample values

Value

the value of the loglikelihood function (changed in sign) for the observed sample `x` under the parameters `par`

Author(s)

Alessandro Barbiero

See Also

`estdweibull3`

Examples

```
n <- 20
c <- 1/3
beta <- 2/3
x <- rdweibull3(n, c, beta)
par <- estdweibull3(x, "ML")
par
-loglikedw3(par, x)
```

lossdw *Loss function*

Description

Loss function for the method of moments (type 1 discrete Weibull)

Usage

```
lossdw(par, x, zero = FALSE, eps = 1e-04, nmax=1000)
```

Arguments

`par`	vector of parameters q and β
`x`	the vector of sample values
`zero`	`TRUE`, if the support contains 0; `FALSE` otherwise
`eps`	error threshold for the numerical computation of the expected value
`nmax`	maximum value considered for the numerical computation of the expected value

Details

The loss function is given by $L(x; q, \beta) = [m_1 - \mathrm{E}(X; q, \beta)]^2 + [m_2 - \mathrm{E}(X^2; q, \beta)]^2$, where $\mathrm{E}(\cdot)$ denotes the expected value, m_1 and m_2 are the first and second order sample moments respectively.

Value

the value of the quadratic loss function

Author(s)

Alessandro Barbiero

See Also

`Edweibull`

Examples

```
x <- c(1,1,1,1,1,2,2,2,3,4)
lossdw(c(0.5, 1), x)
par <- estdweibull(x, "M") # parameter estimates derived by the method of moments
par
lossdw(par, x) # the loss is zero using these estimates
```

lossdw3 *Loss function*

Description

Loss function for the method of moments (type 3 discrete Weibull)

Usage

```
lossdw3(par, x, eps = 1e-04)
```

Arguments

par	vector of parameters q and β
x	the vector of sample values
eps	error threshold for the numerical computation of the expected value

Details

The loss function is given by $L(x; c, \beta) = [m_1 - \mathrm{E}(X; c, \beta)]^2 + [m_2 - \mathrm{E}(X^2; c, \beta)]^2$, where $\mathrm{E}(\cdot)$ denotes the expected value, m_1 and m_2 are the first and second order sample moments respectively.

Value

the value of the quadratic loss function

Author(s)

Alessandro Barbiero

See Also

Edweibull3

Examples

```
n <- 25
c <- 1/3
beta <- 2/3
x <- rdweibull3(n, c, beta)
par <- estdweibull3(x, "M")
par
lossdw3(par, x)
```

varFisher — *Observed Fisher information matrix*

Description

Observed Fisher information matrix on a sample from the type 1 discrete Weibull distribution

Usage

```
varFisher(x, zero = FALSE)
```

Arguments

`x`	a vector of sample values
`zero`	`TRUE`, if the support contains 0; `FALSE` otherwise

Value

a list of two matrices: the observed Fisher information matrix, and its inverse, which contains asymptotic variances and covariances of the maximum likelihood estimators of q and β

Author(s)

Alessandro Barbiero

See Also

`estdweibull`

Examples

```
x <- rdweibull(100, 2/3, 3/2)
estdweibull(x, "ML")
IF <- varFisher(x)[[2]]
diag(IF)
diag(IF)/length(x)
# asymptotic variances of the ML estimators directly estimated from the sample
```

Index