

Package ‘DWreg’

May 7, 2026

Title Parametric Regression for Discrete Response

Version 3.0

Description Regression for a discrete response, where the conditional distribution is modelled via a discrete Weibull distribution.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Imports maxLik, DiscreteWeibull, Ecdat,survival

NeedsCompilation no

Author Veronica Vinciotti [aut, cre]

Maintainer Veronica Vinciotti <veronica.vinciotti@unitn.it>

Repository CRAN

Date/Publication 2025-05-19 11:50:02 UTC

Contents

dw	1
dw.meanvar	3
dw.parest	4
dw.reg	5
res.dw	7

Index	9
--------------	----------

dw	<i>Discrete Weibull</i>
----	-------------------------

Description

Density, distribution function, quantile function and random generation for the discrete Weibull distribution with parameters q and β .

Usage

```
ddw(x,q=exp(-1),beta=1)
pdw(x,q=exp(-1),beta=1)
qdw(p,q=exp(-1),beta=1)
rdw(n,q=exp(-1),beta=1)
```

Arguments

x	quantile
p	probability
n	number of observations
q, beta	Parameters of the distribution

Details

The discrete Weibull distribution has density

$$p(x, q, \beta) = q^{x^\beta} - q^{(x+1)^\beta}$$

for $x = 0, 1, 2, \dots$. If q or beta are not specified they assume the default values of exp(-1) and 1, respectively. In this case, DW corresponds to a geometric distribution with $p=1-q$.

Value

ddw gives the density, pdw gives the distribution function, qdw gives the quantile function, and rdw generates random samples from a DW distribution with parameters q and beta.

Author(s)

Veronica Vinciotti

References

Nagakawa T, Osaki S. The discrete Weibull distribution. IEEE transactions on reliability 1975; R-24(5).

Examples

```
x<-rdw(1000,q=0.9,beta=1.5)
hist(x)
plot(x,unlist(lapply(x,ddw,q=0.9,beta=1.5)),ylab="density")
plot(x,unlist(lapply(x,pdw,q=0.9,beta=1.5)),ylab="cdf")
```

dw.meanvar

*Mean and Variance of Discrete Weibull***Description**

Mean and variance of a discrete Weibull distribution with parameters q and beta.

Usage

```
dw.meanvar(q,beta,M)
```

Arguments

q, beta	Parameters of the distribution
M	Maximum value of the summation. Default value is 1000.

Details

The mean and variance are computed using the following approximations:

$$E(X) = \sum_{k=1}^M q^{k^\beta}$$

$$E(X^2) = \sum_{k=1}^M (2k - 1)q^{k^\beta} = 2 \sum_{k=1}^M kq^{k^\beta} - E(X)$$

Value

The function returns the mean and variance of a DW distribution with parameters q and beta.

Author(s)

Veronica Vinciotti

References

Khan M, Khaliq A, Abouammoth A. On estimating parameters in a discrete Weibull distribution. IEEE transactions on Reliability 1989; 38(3):348-350.

Examples

```
dw.meanvar(q=0.9,beta=1.5)
#compare with sample mean/variance from a random sample
x<-rdw(1000,q=0.9,beta=1.5)
mean(x)
var(x)
```

`dw.parest`*Parameter estimation for discrete Weibull*

Description

Estimation of the parameters q and β of a discrete Weibull distribution

Usage

```
dw.parest(data,method,method.opt)
```

Arguments

<code>data</code>	Vector of observations
<code>method</code>	Either "likelihood" or "proportion"
<code>method.opt</code>	Optimization criterion used in maxLik (default is "NR")

Details

If `method="likelihood"`, the parameters q and β are estimated by maximum likelihood.

If `method="proportion"`, the method of Araujo Santos and Fraga Alves (2013) is used, based on count frequencies.

Value

The function returns the parameter estimates of q and β .

Author(s)

Veronica Vinciotti

References

Araujo Santos P, Fraga Alves M. Improved shape parameter estimation in a discrete Weibull model. Recent Developments in Modeling and Applications in Statistics . Studies in Theoretical and Applied Statistics. Springer-Verlag, 2013; 71-80.

Examples

```
x<-rdw(1000,q=0.9,beta=1.5)
dw.parest(x) #maximum likelihood estimates
dw.parest(x,method="proportion") #proportion estimates
```

dw.reg	<i>DW regression</i>
--------	----------------------

Description

Parametric regression for discrete response data. The conditional distribution of the response given the predictors is assumed to be DW with parameters q and β dependent on the predictors.

Usage

```
dw.reg(formula, data, tau=0.5, para.q1=NULL, para.q2=NULL, para.beta=NULL, ...)
```

Arguments

formula	An object of class "formula": a symbolic description of the model to be fitted.
data	An optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>dw.qr</code> is called.
tau	Quantile value (default 0.5). This is used only to extract the conditional quantile from the fitted distribution.
para.q1, para.q2	logical flag. If TRUE, the model includes a dependency of q on the predictors, as explained below.
para.beta	logical flag. If TRUE, the model includes a dependency of β on the predictors, as explained below.
...	Additional arguments to the <code>maxLik</code> function

Details

The conditional distribution of Y (response) given x (predictors) is assumed a $DW(q(x), \beta(x))$.

If `para.q1=TRUE`,

$$\log(q/(1-q)) = \theta_0 + \theta_1 X_1 + \dots + \theta_p X_p.$$

If `para.q2=TRUE`,

$$\log(-\log(q)) = \theta_0 + \theta_1 X_1 + \dots + \theta_p X_p.$$

This is equivalent to a continuous Weibull regression model with interval-censored data.

If `para.q1=NULL` and `para.q2=NULL`, then $q(x)$ is constant.

If `para.beta=TRUE`,

$$\log(\beta) = \gamma_0 + \gamma_1 X_1 + \dots + \gamma_p X_p.$$

Otherwise $\beta(x)$ is constant.

Value

A list of class `dw.reg` containing the following components:

<code>call</code>	the matched call.
<code>data</code>	the input data as a list of response and covariates.
<code>coefficients</code>	the theta and gamma estimated coefficients.
<code>loglik</code>	the log-likelihood of the model.
<code>fitted.values</code>	fitted values (on the response scale) for the specified quantile tau.
<code>fitted.q</code>	fitted q values.
<code>fitted.beta</code>	fitted beta values.
<code>residuals</code>	randomised quantile residuals of the fitted model.
<code>tTable</code>	coefficients, standard errors, etc.
<code>tTable.survreg</code>	Only for the model <code>para.q2=TRUE</code> . Coefficients, standard errors, etc from the <code>survreg</code> parametrization. These estimates are linked to changes of $\log(\text{Median}+1)$.

Author(s)

Veronica Vinciotti, Hadeel Kalktawi, Alina Peluso

References

Kalktawi, Vinciotti and Yu (2016) A simple and adaptive dispersion regression model for count data.

Examples

```
#simulated example (para.q1=TRUE, beta constant)
theta0 <- 2
theta1 <- 0.5
beta<-0.5
n<-500
x <- runif(n=n, min=0, max=1.5)
logq<-theta0 + theta1 * x - log(1+exp(theta0 + theta1 * x))
y<-unlist(lapply(logq,function(x,beta) rdw(1,q=exp(x),beta),beta=beta))
data.sim<-data.frame(x,y) #simulated data
fit<-dw.reg(y~x,data=data.sim,para.q1=TRUE)
fit$tTable
```

```
#simulated example (para.q2=TRUE, beta constant)
theta0 <- -2
theta1 <- -0.5
beta<-0.5
n<-500
x <- runif(n=n, min=0, max=1.5)
logq<--exp(theta0 + theta1 * x)
y<-unlist(lapply(logq,function(x,beta) rdw(1,q=exp(x),beta),beta=beta))
data.sim<-data.frame(x,y) #simulated data
fit<-dw.reg(y~x,data=data.sim,para.q2=TRUE)
```

```
fit$table
fit$survreg

#real example
library(Ecdat)
data(StrikeNb)
fit<-dw.reg(strikes~output,data=StrikeNb,para.q2=TRUE)
fit$table
fit$survreg
```

res.dw

DW regression: Diagnostics

Description

Quantile-Quantile plot of the randomised quantile residuals of a DW regression fitted model with 95% simulated envelope.

Usage

```
res.dw(obj,k)
```

Arguments

obj An object of class "dw.reg": the output of the dw.reg function.
k The number of iterations for the simulated envelope.

Details

Diagnostic check for a DW regression model. The randomised quantile residuals should follow a standard normal distribution.

Value

A q-q plot of the residuals with 95% simulated envelope

Author(s)

Veronica Vinciotti, Hadeel Kalktawi

References

Kalktawi, Vinciotti and Yu (2016) A simple and adaptive dispersion regression model for count data.

Examples

```
#simulated example (para.q2=TRUE, beta constant)
theta0 <- -2
theta1 <- -0.5
beta<-0.5
n<-500
x <- runif(n=n, min=0, max=1.5)
logq<--exp(theta0 + theta1 * x)
y<-unlist(lapply(logq,function(x,beta) rdw(1,q=exp(x),beta),beta=beta))
data.sim<-data.frame(x,y) #simulated data
fit<-dw.reg(y~x,data=data.sim,para.q2=TRUE)
res.dw(fit,k=5)
ks.test(fit$residuals,"pnorm")

#real example
library(Ecdat)
data(StrikeNb)
fit<-dw.reg(strikes~output,data=StrikeNb,para.q2=TRUE)
res.dw(fit,k=5)
ks.test(fit$residuals,"pnorm")
```

Index

- * **dw.meanvar**
 - dw.meanvar, 3
- * **dw.parest**
 - dw.parest, 4
- * **dw.reg**
 - dw.reg, 5
- * **dw**
 - dw, 1
- * **res.dw**
 - res.dw, 7

ddw (dw), 1
dw, 1
dw.meanvar, 3
dw.parest, 4
dw.reg, 5

pdw (dw), 1

qdw (dw), 1

rdw (dw), 1
res.dw, 7