Package ‘ewp’

April 22, 2025

Type Package

Title An Empirical Model for Underdispersed Count Data

Version 0.1.2

Description Count regression models for underdispersed small counts (lambda < 20) based on the three-parameter exponentially weighted Poisson distribution of Ridout & Besbeas (2004) <DOI:10.1191/1471082X04st064oa>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 2.10)

LinkingTo BH, Rcpp

Imports Rcpp, mvtnorm

Suggests covr, DHARMa, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation yes

Author Philipp Boersch-Supan [aut, cre] (<`https://orcid.org/0000-0001-6723-6833`>),
James Clarke [aut] (<`https://orcid.org/0000-0003-1826-2060`>)

Maintainer Philipp Boersch-Supan <`pboesu@gmail.com`>

Repository CRAN

Date/Publication 2025-04-22 11:20:02 UTC

Contents

`coef.ewp` *Extract coefficients*

Description

Extract coefficients

Usage

```
## S3 method for class 'ewp'
coef(object, ...)
```

Arguments

`object`	an object of class ewp
`...`	ignored

Value

a vector of coefficient values. Beware that the lambda parameters are on the log-link scale, whereas the betas are estimated using an identity link.

`dewp3` *Probability mass function of the three-parameter EWP*

Description

Probability mass function of the three-parameter EWP

Usage

```
dewp3(x, lambda, beta1, beta2, sum_limit = max(x) * 3)
```

Arguments

`x`	vector of (positive integer) quantiles.
`lambda`	centrality parameter
`beta1`	lower-tail dispersion parameter
`beta2`	upper tail dispersion parameter
`sum_limit`	summation limit for the normalizing factor

Value

a vector of probabilities

`dewp3_cpp` *Probability mass function of the three-parameter EWP*

Description

Probability mass function of the three-parameter EWP

Usage

```
dewp3_cpp(x, lambda, beta1, beta2, sum_limit)
```

Arguments

`x`	vector of (positive integer) quantiles.
`lambda`	centrality parameter
`beta1`	lower-tail dispersion parameter
`beta2`	upper tail dispersion parameter
`sum_limit`	summation limit for the normalizing factor

Value

a probability mass

`ewp_reg` *Exponentially weighted Poisson regression model*

Description

Exponentially weighted Poisson regression model

Usage

```
ewp_reg(
  formula,
  family = "ewp3",
  data,
  verbose = TRUE,
  method = "Nelder-Mead",
  hessian = TRUE,
  autoscale = TRUE,
  maxiter = 500,
  sum_limit = round(max(Y) * 3),
  start_val = NULL
)
```

Arguments

`formula` an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.

`family` choice of "ewp2" or "ewp3"

`data` a data frame containing the variables in the model.

`verbose` logical, defaults to TRUE; print model fitting progress

`method` string, passed to optim, defaults to ’BFGS’

`hessian` logical, defaults to TRUE; calculate Hessian?

`autoscale` logical, defaults to TRUE; automatically scale model parameters inside the optimisation routine based on initial estimates from a Poisson regression.

`maxiter` numeric, maximum number of iterations for optim

`sum_limit` numeric, defaults to 3*maximum count; upper limit for the sum used for the normalizing factor.

`start_val` list, defaults to fitting a Poisson regression; specify starting values

Value

an ewp model

`fitted.ewp` *Extract fitted values*

Description

Extract fitted values

Usage

```
## S3 method for class 'ewp'
fitted(object, ...)
```

Arguments

`object`	an object of class ewp
`...`	ignored

Value

a vector of fitted values on the response scale

`linnet` *Linnet clutch sizes*

Description

A dataset containing the clutch sizes for linnet, recreated from Ridout & Besbeas 2004

Usage

```
linnet
```

Format

A data frame with 5414 rows and 3 variables:

eggs clutch size

cov1 a synthetic random noise covariate

cov2 a synthetic covariate that is positively correlated with the outcome

Source

Ridout & Besbeas 2004, P. Boersch-Supan

`logLik.ewp` *Extract log likelihood*

Description

Extract log likelihood

Usage

```
## S3 method for class 'ewp'
logLik(object, ...)
```

Arguments

`object`	an object of class ewp
`...`	ignored

Value

a numeric

`mmean` *Estimate marginal means*

Description

Estimate marginal means

Usage

```
mmean(object, cov, ci = TRUE, nsamples = 250, ...)
```

Arguments

`object`	ewp model object
`cov`	character, covariate to find marginal mean for
`ci`	logical, defaults to TRUE, whether or not to include confidence intervals
`nsamples`	numeric, defaults to 250, number of samples for use in obtaining the confidence intervals
`...`	ignored

Value

printout of the marginal means

`predict.ewp` *Predict from fitted model*

Description

Predict from fitted model

Usage

```
## S3 method for class 'ewp'
predict(object, newdata, type = c("response"), na.action = na.pass, ...)
```

Arguments

`object`	ewp model object
`newdata`	optional data.frame
`type`	character; default="response", no other type implemented
`na.action`	defaults to na.pass()
`...`	ignored

Value

a vector of predictions

`print.ewp` *Print ewp model object*

Description

Print ewp model object

Usage

```
## S3 method for class 'ewp'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	ewp model object
`digits`	digits to print
`...`	ignored

Value

a summary printout of the ewp model call and fitted coefficients.

`print.summary.ewp` *Print ewp model summary*

Description

Print ewp model summary

Usage

```
## S3 method for class 'summary.ewp'
print(x, digits = max(3, getOption(”digits”) - 3), ...)
```

Arguments

`x`	ewp model summary
`digits`	number of digits to print
`...`	additional arguments to printCoefmat()

Value

printout of the summary object

`rewp3` *Random samples from the three-parameter EWP*

Description

Random samples from the three-parameter EWP

Usage

```
rewp3(n, lambda, beta1, beta2, sum_limit = 30)
```

Arguments

`n`	number of observations
`lambda`	centrality parameter
`beta1`	lower-tail dispersion parameter
`beta2`	upper tail dispersion parameter
`sum_limit`	summation limit for the normalizing factor

Value

random deviates from the EWP_3 distribution

simulate.ewp *simulate from fitted model*

Description

simulate from fitted model

Usage

```
## S3 method for class 'ewp'
simulate(object, nsim = 1, ...)
```

Arguments

`object`	ewp model object
`nsim`	number of response vectors to simulate. Defaults to 1.
`...`	ignored

Value

a data frame with ‘nsim‘ columns.

summary.ewp *Model summary*

Description

Model summary

Usage

```
## S3 method for class 'ewp'
summary(object, ...)
```

Arguments

`object`	ewp model fit
`...`	ignored

Value

The function ‘summary.ewp‘ computes and returns a list of summary statistics of the fitted ewp model.

vcov.ewp *Extract estimated variance-covariance matrix*

Description

Extract estimated variance-covariance matrix

Usage

```
## S3 method for class 'ewp'
vcov(object, ...)
```

Arguments

object	an object of class ewp
...	ignored

Value

a matrix

Index