Package 'BinaryEPPM'

May 6, 2026

Type Package

Title Mean and Scale-Factor Modeling of Under- And Over-Dispersed Binary Data

Version 3.0

Imports Formula, expm, numDeriv, stats, lmtest, grDevices, graphics

Date 2024-06-03

Depends R (>= 3.5.0)

Description Under- and over-dispersed binary data are modeled using an extended Poisson process model (EPPM) appropriate for binary data. A feature of the model is that the under-dispersion relative to the binomial distribution only needs to be greater than zero, but the over-dispersion is restricted compared to other distributional models such as the beta and correlated binomials. Because of this, the examples focus on under-dispersed data and how, in combination with the beta or correlated distributions, flexible models can be fitted to data displaying both under- and over-dispersion. Using Generalized Linear Model (GLM) terminology, the functions utilize linear predictors for the probability of success and scale-factor with various link functions for p, and log link for scale-factor, to fit a variety of models relevant to areas such as bioassay. Details of the EPPM are in Faddy and Smith (2012) <doi:10.1002/bimj.201100214> and Smith and Faddy (2019) <doi:10.18637/jss.v090.i08>.

License GPL-2

Suggests R.rsp

VignetteBuilder R.rsp

NeedsCompilation no

Author David M. Smith [aut, cre], Malcolm J. Faddy [aut]

Maintainer David M. Smith <dmccsmith@verizon.net>

Repository CRAN

Date/Publication 2024-06-04 10:32:25 UTC

Contents

BinaryEPPM-package *Mean and Scale-Factor Modeling of Under- And Over-Dispersed Binary Data*

Description

Under- and over-dispersed binary data are modeled using an extended Poisson process model (EPPM) appropriate for binary data. A feature of the model is that the under-dispersion relative to the binomial distribution only needs to be greater than zero, but the over-dispersion is restricted compared to other distributional models such as the beta and correlated binomials. Because of this, the examples focus on under-dispersed data and how, in combination with the beta or correlated distributions, flexible models can be fitted to data displaying both under- and over-dispersion. Using Generalized Linear Model (GLM) terminology, the functions utilize linear predictors for the probability of success and scale-factor with various link functions for p, and log link for scale-factor, to fit a variety of models relevant to areas such as bioassay. Details of the EPPM are in Faddy and Smith (2012) and Smith and Faddy (2019). Two important changes from version 2.3 are the change to scale-factor rather than variance modeling, and the inclusion of a vignette.

Details

Index of help topics:

BBprob	Calculation of vector of probabilities for the beta binomial distribution.
Berkshires.litters	The data are of the number of male piglets born in litters of varying sizes for the Berkshire breed of pigs.
BinaryEPPM	Fitting of EPPM models to binary data.
BinaryEPPM-package	Mean and Scale-Factor Modeling of Under- And Over-Dispersed Binary Data
CBprob	Calculation of vector of probabilities for the correlated binomial distribution.
EPPMprob	Calculation of vector of probabilities for a extended Poisson process model (EPPM).
GBprob	Calculation of vector of probabilities for the EPPM binomial distribution.
KupperHaseman.case	Kupper and Haseman example data
LL.Regression.Binary	Function called by optim to calculate the log likelihood from the probabilities and hence perform the fitting of regression models to the binary data.
LL.gradient	Function used to calculate the first derivatives of the log likelihood with respect to the model parameters.
Model.BCBinProb	Probabilities for beta and correlated binomial distributions given p's and scale-factors.
Model.Binary	Function for obtaining output from

	distributional models.
Model.GB	Probabilities for binomial and EPPM extended binomial distributions given p's and b.
Model.JMVGB	Probabilities for EPPM extended binomial distributions given p's and scale-factors.
Parkes.litters	The data are of the number of male piglets born in litters of varying sizes for the Parkes breed of pigs.
Yorkshires.litters	The data are of the number of male piglets born in litters of varying sizes for the Yorkshire breed of pigs.
coef.BinaryEPPM	Extraction of model coefficients for BinaryEPPM Objects
cooks.distance.BinaryEPPM	Cook's distance for BinaryEPPM Objects
doubexp	Double exponential Link Function
doubrecip	Double reciprocal Link Function
fitted.BinaryEPPM	Extraction of fitted values from BinaryEPPM Objects
hatvalues.BinaryEPPM	Extraction of hat matrix values from BinaryEPPM Objects
logLik.BinaryEPPM	Extract Log-Likelihood
loglog	Log-log Link Function
negcomplog	Negative complementary log-log Link Function
plot.BinaryEPPM	Diagnostic Plots for BinaryEPPM Objects
powerlogit	Power Logit Link Function
predict.BinaryEPPM	Prediction Method for BinaryEPPM Objects
print.BinaryEPPM	Printing of BinaryEPPM Objects
print.summaryBinaryEPPM	Printing of summaryBinaryEPPM Objects
residuals.BinaryEPPM	Residuals for BinaryEPPM Objects
ropespores.case	Dilution series for the presence of rope spores.
ropespores.grouped	Dilution series for the presence of rope spores.
summary.BinaryEPPM	Summary of BinaryEPPM Objects
vcov.BinaryEPPM	Variance/Covariance Matrix for Coefficients
waldtest.BinaryEPPM	Wald Test of Nested Models for BinaryEPPM Objects
wordcount.case	Number of occurences of an article in five-word and ten-word samples from two authors.
wordcount.grouped	Number of occurences of an article in five-word and ten-word samples from two authors.

Author(s)

David M. Smith [aut, cre], Malcolm J. Faddy [aut]

Maintainer: David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

Faddy M, Smith D. (2012). Extended Poisson Process Modeling and Analysis of Grouped Binary Data. *Biometrical Journal*, **54**, 426-435. doi:10.1002/bimj.201100214.

Grun B, Kosmidis I, Zeileis A. (2012). Extended Beta Regression in R: Shaken, Stirred, Mixed, and Partitioned. *Journal of Statistical Software*, **48**(11), 1-25. doi:10.18637/jss.v048.i11.

Smith D, Faddy M. (2019). Mean and Variance Modeling of Under-Dispersed and Over-Dispersed Grouped Binary Data. *Journal of Statistical Software*, **90**(8), 1-20. doi:10.18637/jss.v090.i08.

Zeileis A, Croissant Y. (2010). Extended Model Formulas in R: Multiple Parts and Multiple Responses. *Journal of Statistical Software*, **34**(1), 1-13. doi:10.18637/jss.v034.i01.

See Also

CountsEPPM betareg

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution),
                  model.type = 'p only', model.name = 'binomial')
summary(output.fn)
```

BBprob *Calculation of vector of probabilities for the beta binomial distribution.*

Description

Given a vector of parameters and a scalar of the number of trials the function returns a vector of probabilities.

Usage

```
BBprob(twoparameter, nt)
```

Arguments

twoparameter	A vector of the parameters of the beta binomial distribution.
nt	The number of trials.

Value

Vector of probabilities

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Smith D (1982). Algorithm AS189. Maximum Likelihood Estimation of the Parameters of the Beta Binomial Distribution. Applied Statistics, 32, 196-204.

Williams D (1996). "Overdispersion in Logistic Linear Models." In B Mrgan (ed.), Statistics in Toxicology, pp75-84. Oxford Science Publications.

Examples

```
twoparameter <- c(0.96477815,0.7561417)
names(twoparameter) <- c('p','theta')
nt <- 37
BBprob(twoparameter,nt)
```

Berkshires.litters *The data are of the number of male piglets born in litters of varying sizes for the Berkshire breed of pigs.*

Description

The data are arranged as a list of binomial frequency distributions where the listing is by litter size which is included both as a variate (vsize) and as a factor (fsize)

Usage

```
data("Berkshires.litters")
```

Format

The format is: List of 3 $ fsize : Factor w/ 7 levels " size 5"," size 6",..: 1 2 3 4 5 6 7 $ vsize : int [1:7] 5 6 7 8 9 10 11 $ number.success:List of 7 ..$: num [1:6] 8 29 72 65 40 3 ..$: num [1:7] 5 22 89 129 74 35 4 ..$: num [1:8] 1 25 62 131 136 89 26 5 ..$: num [1:9] 1 15 79 179 219 149 71 33 4 ..$: num [1:10] 2 6 47 117 172 181 117 40 9 2 ..$: num [1:11] 2 1 23 65 131 145 120 61 20 3 ... ..$: num [1:12] 0 3 9 22 53 94 72 54 20 4 ...

Source

Brooks, R.J., James, W.H., Gray, E. (1993). Modelling Sub-Binomial Variation in the Frequency of Sex Combinations in Litters of Pigs. Biometrics 47, 403-417.

Examples

```
data("Berkshires.litters")
```

BinaryEPPM *Fitting of EPPM models to binary data.*

Description

Fits regression models to under- and over-dispersed binary data using extended Poisson process models.

Usage

```
BinaryEPPM(formula, data, subset = NULL, na.action = NULL,
       weights = NULL, model.type = "p only",
       model.name = "EPPM extended binomial", link = "cloglog",
       initial = NULL, method = "Nelder-Mead",
       pseudo.r.squared.type = "square of correlation", control = NULL)
```

Arguments

formula	Formulae for the probability of a success p and scale-factor. The object used is from the package Formula of Zeileis and Croissant (2010) which allows multiple parts and multiple responses. "formula" should consist of a left hand side (lhs) of single response variable and a right hand side (rhs) of one or two sets of variables for the linear predictors for the mean and (if two sets) the variance. This is as used for the R function "glm" and also, for example, as for the package "betareg" (Cribari-Neto and Zeileis, 2010). The function identifies from the argument data whether a data frame (as for use of "glm") or a list has been input. The list should be exactly the same as for a data frame except that the response variable is a list of vectors of frequency distributions rather than two vectors of paired counts of number responding (r) out of number tested as for the data frame. The subordinate functions fit models where the response variables are "p.obs", or "scalef.obs" according to the model type being fitted. The values for these response variables are not input as part of "data", they are calculated within the function from a list of grouped binary data input. If the "model.type" is "p only", "formula" consists of a lhs of the response variable and a rhs of the terms of the linear predictor for the mean model. If the "model.type" is "p and scale-factor" there are two sets of terms in the rhs of "formula" i.e., "p.obs" and "scalef.obs" together with the two sets of terms for the linear predictors of p and scale-factor.
data	"data" should be either a data frame (as for use of "glm") or a list. The list should be exactly the same as for a data frame except that the response variable is a list of vectors of frequency distributions rather than a vector of single counts as for the data frame. Only one list is allowed within "data" as it is identified as the dependent variable. If other lists are in "data", for example for use as weights, they should be removed from "data" prior to calling this function. The extracted list can be called using the "weights" argument to this function. Within the function a working list "listcounts" and data frames with components such as "p.obs", "scalef.obs", "covariates", "offset.mean", "offset.variance" are set up .

The component "covariates" is a data frame of vectors of covariates in the model. The component "listcounts" is a list of vectors of frequency distributions, or the single pairs of r/n in grouped form if "data" is a data frame.

`subset` Subsetting commands.

`na.action` Action taken for NAs in data.

`weights` Vector of list of lists of weights.

`model.type` Takes one of two values i.e. "p only" or "p and scale-factor". The "p only" value fits a linear predictor function to the parameter *a* in equation (3) of Faddy and Smith (2012). If the model type being fitted is binomial, modeling *a* is the same as modeling the mean. For the negative binomial the mean is *b exp(a)-1)*, *b* also being as in equation (3) of Faddy and Smith (2012). The "p and scale-factor" value fits linear predictor functions to both the probability of a success p and the scale-factor.

`model.name` If model.type is "p only" the model being fitted is one of the four "binomial", "EPPM extended binomial", "beta binomial", "correlated binomial". If model.type is "p and scale-factor" the model being fitted is either "EPPM extended binomial" i.e. as equations (4) and (6) of Faddy and Smith (2012) or one of the two "beta binomial", "correlated binomial".

`link` Takes one of nine values i.e., 'logit', 'probit', 'cloglog', 'cauchit', 'log', 'loglog', 'double exponential', 'double reciprocal', 'power logit'. The default is 'cloglog'. The 'power logit' has an attribute of 'power' for which the default is 1 i.e., a logit link.

`initial` This is a vector of initial values for the parameters. If this vector is NULL then initial values based on a fitting binomial models using "glm" are calculated within the function.

`method` Takes one of the two values "Nelder-Mead" or "BFGS" these being arguments of `optim`.

`pseudo.r.squared.type`
Takes one of the three values "square of correlation", "R square" or "max-rescaled R square". The "default" is as used in Cribari-Neto and Zeileis (2010) and is the square of the correlation between the observed and predicted values on the GLM linear predictor scale. The other two are as described in Cox and Snell (1989), and Nagelkerke (1991) and apply to logistic regression.

`control` "control" is a list of control parameters as used in "optim". If this list is NULL the defaults for "optim" are set as "control <- list(fnscale=-1, trace=0, maxit=1000)". The control parameters that can be changed by inputting a variable length list are "fnscale, trace, maxit, abstol, reltol, alpha, beta, gamma". Details of "optim" and its control parameters are available in the online R help manuals.

Value

An object of class "BinaryEPMM" is returned. A list of object items follows.

`data.type` The type of the data i.e., data frame or list

`list.data` Data as a list of lists of frequency distributions

call	The call of the function
formula	The formula argument
model.type	The type of model being fitted
model.name	The model being fitted
link	The link function
covariates.matrix.p	The design matrix for the probability of a success
covariates.matrix.scalef	The design matrix for the scalefactor
offset.p	The offset vector for the probability of a success
offset.scalef	The offset vector for the scalefactor
coefficients	Estimates of model parameters
loglikelihood	Loglikelihood
vcov	The variance/covariance matrix
n	The number of observations
nobs	The number of observations
df.null	The degrees of freedom of the null model
df.residual	The degrees of freedom of the residual
vnmax	Vector of maximums of grouped count data vectors in list.counts
weights	Vector or list of weights
converged	Whether the iterative process converged, TRUE or FALSE
iterations	Number of iterations taken
method	Method for optim either Nelder-Mead or BFGS
pseudo.r.squared	Pseudo R**2 value
start	Starting values for iterative process
optim	Estimates of model parameters
control	Control parameters for optim
fitted.values	Fitted values for probability of success
y	Dependent variable
terms	Terms in model fitted

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cox DR, Snell EJ. (1989). *Analysis of Binary Data*. Second Edition. Chapman & Hall.

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

Grun B, Kosmidis I, Zeileis A. (2012). Extended Beta Regression in R: Shaken, Stirred, Mixed, and Partitioned. *Journal of Statistical Software*, **48**(11), 1-25. doi:10.18637/jss.v048.i11.

Faddy M, Smith D. (2012). Extended Poisson Process Modeling and Analysis of Grouped Binary Data. *Biometrical Journal*, **54**, 426-435. doi:10.1002/bimj.201100214.

Nagelkerke NJD. (1991). A Note on a General Definition of the Coefficient of Determination. *Biometrika*, **78**, 691-692.

Smith D, Faddy M. (2019). Mean and Variance Modeling of Under-Dispersed and Over-Dispersed Grouped Binary Data. *Journal of Statistical Software*, **90**(8), 1-20. doi:10.18637/jss.v090.i08.

Zeileis A, Croissant Y. (2010). Extended Model Formulas in R: Multiple Parts and Multiple Responses. *Journal of Statistical Software*, **34**(1), 1-13. doi:10.18637/jss.v034.i01.

See Also

`CountsEPPM` `betareg`

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution),
                  model.type = "p only", model.name = "binomial")
summary(output.fn)
```

`CBprob` *Calculation of vector of probabilities for the correlated binomial distribution.*

Description

Given a vector of parameters and a scalar of the number of trials the function returns a vector of probabilities.

Usage

```
CBprob(twoparameter, nt)
```

Arguments

`twoparameter`	A vector of the parameters of the correlated binomial distribution.
`nt`	The number of trials.

Value

Vector of probabilities

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Kupper L, Haseman J (1978). The Use of a Correlated Binomial Model for the Analysis of Toxicological Experiments. Biometrics, 34(1), 69-76.

Examples

```
twoparameter <- c(0.971242852,0.001465007)
names(twoparameter) <- c('p','rho')
nt <- 37
CBprob(twoparameter,nt)
```

`coef.BinaryEPPM` *Extraction of model coefficients for BinaryEPPM Objects*

Description

Extract the regression model coefficients from models of class "BinaryEPMM".

Usage

```
## S3 method for class 'BinaryEPPM'
coef(object, prtpar = c("full", "p", "scale.factor"), ...)
```

Arguments

`object`	fitted model object of class "BinaryEPPM".
`prtpar`	character indicating coefficients of the fitted model to be output: all coefficients ("full"), coefficients of the model for probability of success ("p"), coefficients of the model for scale-factor ("scale.factor")
`...`	some methods for this generic function require additional arguments.

Details

One of a set of standard extractor functions for fitted model objects of class "BinaryEPPM.

Value

Vector of coefficients of fitted regression model.

Author(s)

David M. Smith <dmccsmith@verizon.net>

See Also

betareg

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution))
coef(output.fn, prtpar = "full")
coef(output.fn, prtpar = "p")
coef(output.fn, prtpar = "scale.factor")
```

---

cooks.distance.BinaryEPPM

Cook's distance for BinaryEPPM Objects

Description

Calculates Cook's distances for BinaryEPPM objects.

Usage

```
## S3 method for class 'BinaryEPPM'
cooks.distance(model, ...)
```

Arguments

model	fitted model object of class "BinaryEPPM".
...	some methods for this generic function require additional arguments.

Details

Cook's distances as in GLMs.

Value

A vector of Cook's distances.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

betareg

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution),
                  model.type = 'p only', model.name = 'binomial')
cooks.distance(output.fn)
```

---

doubexp *Double exponential Link Function*

Description

Computes the double exponential link function, including its inverse.

Usage

```
doubexp()
```

Value

The double exponential transformation of theta.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Ford I, Torsney B, Wu C (1992). "The Use of a Canonical Form in the Construction of Locally Optimal Designs for Non-linear Problems." Journal of the Royal Statistical Society B, 54, 569-583. doi:10.1111/j.25176161.1992.tb01897.x

doubrecip *Double reciprocal Link Function*

Description

Computes the double reciprocal link function, including its inverse.

Usage

```
doubrecip()
```

Value

The double reciprocal transformation of theta.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Ford I, Torsney B, Wu C (1992). "The Use of a Canonical Form in the Construction of Locally Optimal Designs for Non-linear Problems." Journal of the Royal Statistical Society B, 54, 569-583. doi:10.1111/j.25176161.1992.tb01897.x

EPPMprob *Calculation of vector of probabilities for a extended Poisson process model (EPPM).*

Description

Calculates a vector of probabilities given a vector of rates using the matrix exponential function from Maechler, Dutang, Goulet, Bates, Firth (2023).

Usage

```
EPPMprob(vlambda)
```

Arguments

vlambda	a vector of rates of an extended Poisson process.

Details

This is a similar function to that in Smith and Faddy (2014).

Value

The value returned is a vector of probabilities.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Maechler M, Dutang C, Goulet V, Bates D, Firth D. (2023). expm: Matrix Exponential. R package version 0.999-8, https://CRAN.R-project.org/package=expm.

Smith D, Faddy M (2014). CountsEPPM: Mean and Variance Modeling of Count Data. R package version 2.0, https://CRAN.R-project.org/package=CountsEPPM.

`fitted.BinaryEPPM` *Extraction of fitted values from BinaryEPPM Objects*

Description

This function is generic. Extract the fitted values from models of class "BinaryEPMM".

Usage

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

Arguments

`object`	fitted model object of class "BinaryEPPM".
`...`	currently not used.

Details

This function is included so that function lrtest from package lmtest can be used.

Value

An vector of class "numeric" of the fitted values from the object of class "BinaryEPMM".

Author(s)

David M. Smith <dmccsmith@verizon.net>

See Also

`fitted`

GBprob *Calculation of vector of probabilities for the EPPM binomial distribution.*

Description

Given a vector of parameters and a scalar of the number of trials the function returns a vector of probabilities. The name GBprob is used to avoid confusion with EPPMprob which is the function calculating the probabilties given the constructed vector vector of lambdas.

Usage

```
GBprob(twoparameter, nt)
```

Arguments

twoparameter	A vector of the parameters of the EPPM binomial distribution.
nt	The number of trials.

Value

Vector of probabilities

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Faddy M, Smith D. (2012). Extended Poisson Process Modeling and Analysis of Grouped Binary Data. *Biometrical Journal*, **54**, 426-435. doi:10.1002/bimj.201100214.

Examples

```
twoparameter <- c(0.971242852,0.001465007)
names(twoparameter) <- c('p','b')
nt <- 37
GBprob(twoparameter,nt)
```

`hatvalues.BinaryEPPM` *Extraction of hat matrix values from BinaryEPPM Objects*

Description

Extract the values of the hat matrix from models of class "BinaryEPMM".

Usage

```
## S3 method for class 'BinaryEPPM'
hatvalues(model, ...)
```

Arguments

`model`	fitted model object of class "BinaryEPPM".
`...`	some methods for this generic function require additional arguments.

Value

The calculated hat values for the fitted model. These are used to calculate Cook's distances.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

`betareg`

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution),
                  model.type = 'p only', model.name = 'binomial')
hatvalues(output.fn)
```

KupperHaseman.case *Kupper and Haseman example data*

Description

Data of the number of deaths out of number of implants for pregnant female mice for two groups each of size 10.

Usage

```
data("KupperHaseman.case")
```

Format

A data frame with 20 observations on the following 3 variables.

`Group` a factor with levels `Control` `Treated`

`Number.Deaths` a numeric vector

`Number.Implants` a numeric vector

Source

Kupper L, Haseman J. (1978). The Use of a Correlated Binomial Model for the Analysis of Toxicological Experiments. Biometrics, 34(1), 69-76.

Examples

```
data("KupperHaseman.case")
```

LL.gradient *Function used to calculate the first derivatives of the log likelihood with respect to the model parameters.*

Description

Function used to calculate the first derivatives of the log likelihood with respect to the model parameters. These are numerical derivatives calculated using the numerical derivative functions of Gilbert and Varadhan (2015).

Usage

```
LL.gradient(parameter, model.type, model.name, link, ntrials, nsuccess,
            covariates.matrix.p, covariates.matrix.scalef,
            offset.p, offset.scalef, weights, grad.method)
```

Arguments

parameter	A vector of the parameters of the model which is set to initial estimates on function call.
model.type	Takes one of two values i.e. 'p only' or 'p and scale-factor'. The 'p only' value fits linear predictor functions to the probability of a success 'p' as in Faddy and Smith (2012). The 'p and scale-factor' value fits linear predictor functions to both the 'p' and the scale-factor. The default is 'p and scale-factor'.
model.name	If model.type is 'p only' the model being fitted is one of the four 'binomial', 'EPPM extended binomial', 'beta binomial' or 'correlated binomial'. If model.type is 'p and scale-factor' the model being fitted is one of the three 'EPPM extended binomial', 'beta binomial' or 'correlated binomial'. Information about these models is given in Faddy and Smith (2012). The default is 'EPPM extended binomial'.
link	Takes one of nine values i.e., 'logit', 'probit', 'cloglog', 'cauchit', 'log', 'loglog', 'double exponential', 'double reciprocal', 'power logit'. The default is 'cloglog'. The 'power logit' has an attribute of 'power' for which the default is 1 i.e., a logit link.
ntrials	A vector length 'n+1' representing the number of trials 'n' i.e., a vector with all elements equal to 'n'.
nsuccess	A vector representing the frequency distribution of the binomial distribution for fixed number of trials 'n'.
covariates.matrix.p	A matrix of covariates for the mean where rows are the number of values in list.binary and columns the covariates. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.
covariates.matrix.scalef	A matrix of covariates for the variance where rows are the number of values in list.binary and columns the covariates. The default is a vector of ones. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.
offset.p	An offset vector for the probability of success p. The default is a vector of ones.
offset.scalef	An offset vector for the scale-factor. The default is a vector of ones.
weights	A vector or list of weights for the modeling of probability of success. The default is a vector of ones.
grad.method	Numerical method used to calculate gradients when the optimization method for optim is BFGS either simple or Richardson. This is the grad.method attribute of argument method of BinaryEPPM. The default is simple.

Value

A vector of numerical first derivatives.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Gilbert P, Varadhan R. (2015). numDeriv: Accurate Numerical Derivatives. R Package version 2014.2-1, https://CRAN.R-project.org/package=numDeriv.

Examples

```
link <- 'cloglog'
attr(link, which="p") <- make.link(link)
nsuccess <- list(c(rep(0,5),352,479,530,291,101,17))
ntrials  <- list(c(rep(10,11)))
parameter <- c(0.06363398,-0.47085362)
LL.gradient(parameter, model.type = "p and scale-factor",
   model.name = "EPPM extended binomial", link = link, ntrials = ntrials, nsuccess = nsuccess,
     covariates.matrix.p = matrix(c(1), nrow=1),
     covariates.matrix.scalef = matrix(c(1), nrow=1),
     offset.p = c(0), offset.scalef = c(0), weights = list(c(rep(1,11))),
     grad.method = "Richardson")
```

`LL.Regression.Binary` *Function called by optim to calculate the log likelihood from the probabilities and hence perform the fitting of regression models to the binary data.*

Description

Fits specified regression models to the data.

Usage

```
LL.Regression.Binary(parameter,model.type,model.name,link,ntrials,nsuccess,
                     covariates.matrix.p,covariates.matrix.scalef,
                     offset.p,offset.scalef,weights,grad.method)
```

Arguments

`parameter`	A vector of the parameters of the model which is set to initial estimates on function call.
`model.type`	Takes one of two values i.e. 'p only' or 'p and scale-factor'. The 'p only' value fits linear predictor functions to the probability of a success 'p' as in Faddy and Smith (2012). The 'p and scale-factor' value fits linear predictor functions to both the 'p' and the scale-factor. The default is 'p and scale-factor'.
`model.name`	If model.type is 'p only' the model being fitted is one of the four 'binomial', 'EPPM extended binomial', 'beta binomial' or 'correlated binomial'. If model.type is 'p and scale-factor' the model being fitted is one of the three 'EPPM extended binomial', 'beta binomial' or 'correlated binomial'. Information about these models is given in Faddy and Smith (2012). The default is 'EPPM extended binomial'.

link	Takes one of nine values i.e., 'logit', 'probit', 'cloglog', 'cauchit', 'log', 'loglog', 'double exponential', 'double reciprocal', 'power logit'. The default is 'cloglog'. The 'power logit' has an attribute of 'power' for which the default is 1 i.e., a logit link.
ntrials	A vector length 'n+1' representing the number of trials 'n' i.e., a vector with all elements equal to 'n'.
nsuccess	A vector representing the frequency distribution of the binomial distribution for fixed number of trials 'n'.
covariates.matrix.p	A matrix of covariates for the mean where rows are the number of values in list.binary and columns the covariates. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.
covariates.matrix.scalef	A matrix of covariates for the variance where rows are the number of values in list.binary and columns the covariates. The default is a vector of ones. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.
offset.p	An offset vector for the probability of success p. The default is a vector of ones.
offset.scalef	An offset vector for the scale-factor. The default is a vector of ones.
weights	A vector or list of weights for the modeling of probability of success. The default is a vector of ones.
grad.method	Numerical method used to calculate gradients either simple or Richardson. The default is simple.

Value

The log likelihood is returned.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Faddy M, Smith D. (2012). Extended Poisson Process Modeling and Analysis of Grouped Binary Data. *Biometrical Journal*, **54**, 426-435. doi:10.1002/bimj.201100214.

Examples

```
link <- 'cloglog'
attr(link, which="p") <- make.link(link)
nsuccess <- list(c(rep(0,5),352,479,530,291,101,17))
ntrials  <- list(c(rep(10,11)))
parameter <- c(0.06363398,-0.47085362)
LL.Regression.Binary(parameter, model.type = "p and scale-factor",
            model.name = "EPPM extended binomial", link, ntrials, nsuccess,
```

```
covariates.matrix.p = matrix(c(1), nrow=1),
covariates.matrix.scalef = matrix(c(1), nrow=1),
offset.p = c(0), offset.scalef = c(0),
weights = list(c(rep(1,11))))
```

`logLik.BinaryEPPM` *Extract Log-Likelihood*

Description

This function is generic. It is a method for extracting the log-likelihood for objects of class "BinaryEPPM".

Usage

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

Arguments

`object`	fitted model object of class "BinaryEPPM".
`...`	some methods for this generic function require additional arguments

Details

logLik is most commonly used for a model fitted by maximum likelihood as is done here.

Value

The log likelihood value for the fitted model object.

Author(s)

David M. Smith <dmccsmith@verizon.net>

See Also

`betareg`

`loglog` *Log-log Link Function*

Description

Computes the loglog link function, including its inverse.

Usage

```
loglog()
```

Details

Same link function as in Cribari-Neto and Zeileis (2010).

Value

The loglog of theta where the logarithms are to base e.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

`Model.BCBinProb` *Probabilities for beta and correlated binomial distributions given p's and scale-factors.*

Description

Calculates the probabilities for beta and correlated binomials given values for p's and scale-factors.

Usage

```
Model.BCBinProb(parameter, model.type, model.name, link, ntrials, covariates.matrix.p,
covariates.matrix.scalef = matrix(c(rep(1, nrow(covariates.matrix.p))), ncol = 1),
offset.p = c(rep(0, length(ntrials))), offset.scalef = c(rep(0, length(ntrials))))
```

Arguments

parameter	A vector of the parameters of the model which is set to initial estimates on function call.
model.type	Takes one of two values i.e. 'p only' or 'p and scale-factor'. The 'p only' value fits a linear predictor function to the parameter p which is the 'm(1)' in equation (6) of Faddy and Smith (2012) divided by 'N'. The 'p and scale-factor' value fits linear predictor functions to both p and the scale-factor.
model.name	The model being fitted is one of the two 'beta binomial' or 'correlated binomial'.
link	Takes one of nine values i.e., 'logit', 'probit', 'cloglog', 'cauchit', 'log', 'loglog', 'double exponential', 'double reciprocal', 'power logit'. The default is 'cloglog'. The 'power logit' has an attribute of 'power' for which the default is 1 i.e., a logit link.
ntrials	This is a scalar representing the denominator i.e., the length of the probability mass function returned is this scalar + 1.
covariates.matrix.p	A matrix of covariates for p where rows are the number of values in listbinary and columns the covariates. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.
covariates.matrix.scalef	A matrix of covariates for the scale-factor where rows are the number of values in listbinary and columns the covariates. The default is a vector of ones. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.
offset.p	An offset vector for p. The default is a vector of ones.
offset.scalef	An offset vector for the scale-factor. The default is a vector of ones.

Value

List of arguments input together with a list of probabilities vectors and a data frame of values of p, theta (beta binomial) or rho (correlated binomial) and the limits for theta or rho.

model	The model is either 'beta binomial' or 'correlated binomial'.
link	The link is either 'logit' or 'cloglog'.
parameter	A vector of the parameters of the model which is set to initial estimates on function call.
probabilities	A list of the vectors of probabilities of the model.
probabilities	A data frame of values of p, theta (beta binomial) or rho (correlated binomial) and the limits for theta or rho.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Hughes G, Madden L (1995). Some methods allowing for aggregated patterns of disease incidence in the analysis of data from designed experiments. Plant Pathology, 44, 927-943.

Kupper L, Haseman J (1978). The use of a correlated binomial model for the analysis of toxicological epxeriments. Biometrics, 34(1), 69-76.

Examples

```
link <- 'cloglog'
attr(link, which="p") <- make.link(link)
parameter <- c(-0.68294630,0.03451481)
names(parameter) <- c('p','rho')
model.type <- 'p and scale-factor'
model.name <- 'correlated binomial'
ntrials    <- list(c(rep(10,11)))
Model.BCBinProb(parameter, model.type, model.name, link, ntrials,
                covariates.matrix.p = matrix(c(1),nrow=1),
                covariates.matrix.scalef = matrix(c(1),nrow=1),
                offset.p = c(0), offset.scalef = c(0))
```

`Model.Binary` *Function for obtaining output from distributional models.*

Description

Produces output of model, parameters and probabilities from the various models.

Usage

```
Model.Binary(parameter, model.type, model.name, link, ntrials, covariates.matrix.p,
covariates.matrix.scalef, offset.p, offset.scalef)
```

Arguments

`parameter`	A vector of the parameters of the model which is set to initial estimates on function call.
`model.type`	Takes one of two values i.e. 'p only' or 'p and scale-factor'. The 'p only' value fits a linear predictor function to the parameter p which is the 'm(1)' in equation (6) of Faddy and Smith (2012) divided by 'N'. The 'p and scale-factor' value fits linear predictor functions to both p and the scale-factor.
`model.name`	If model.type is 'p only' the model being fitted is one of the six 'binomial', 'over-dispersed-one', 'over-dispersed-two', 'EPPM binomial', 'beta binomial' or 'correlated binomial'. If model.type is 'p and scale-factor' the model being fitted is one of the three 'EPPM binomial', 'beta binomial' or 'correlated binomial'.

`link` Takes one of nine values i.e., 'logit', 'probit', 'cloglog', 'cauchit', 'log', 'loglog', 'double exponential', 'double reciprocal', 'power logit'. The default is 'cloglog'. The 'power logit' has an attribute of 'power' for which the default is 1 i.e., a logit link.

`ntrials` This is a scalar representing the denominator i.e., the length of the probability mass function returned is this scalar + 1.

`covariates.matrix.p`
A matrix of covariates for p where rows are the number of values in listbinary and columns the covariates. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.

`covariates.matrix.scalef`
A matrix of covariates for the scale-factor where rows are the number of values in listbinary and columns the covariates. The default is a vector of ones. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.

`offset.p` An offset vector for p. The default is a vector of ones.

`offset.scalef` An offset vector for the scale-factor. The default is a vector of ones.

Value

The output from either Model.BCBinProb, Model.GB, Model.Binary, Model.JMVGB, or Model.ODB.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Faddy M, Smith D. (2012). Extended Poisson Process Modeling and Analysis of Grouped Binary Data. *Biometrical Journal*, **54**, 426-435. doi:10.1002/bimj.201100214.

Examples

```
link <- 'cloglog'
attr(link, which="p") <- make.link(link)
parameter <- c(-0.68294630,0.03451481)
names(parameter) <- c('p','rho')
model.type <- 'p and scale-factor'
model.name <- 'correlated binomial'
ntrials    <- list(c(rep(10,11)))
Model.Binary(parameter, model.type, model.name, link, ntrials,
             covariates.matrix.p = matrix(c(1),nrow=1),
             covariates.matrix.scalef = matrix(c(1),nrow=1),
             offset.p = c(0), offset.scalef = c(0))
```

`Model.GB` *Probabilities for binomial and EPPM extended binomial distributions given p's and b.*

Description

Calculates the probabilities for binomial and EPPM extended binomial given values for p's and b.

Usage

```
Model.GB(parameter, model.name, link, ntrials, covariates.matrix.p,
         offset.p = c(rep(0, length(ntrials))))
```

Arguments

`parameter`	A vector of the parameters of the model which is set to initial estimates on function call.
`model.name`	The model being fitted is one of the two 'binomial' or 'EPPM extended binomial'.
`link`	Takes one of nine values i.e., 'logit', 'probit', 'cloglog', 'cauchit', 'log', 'loglog', 'double exponential', 'double reciprocal', 'power logit'. The default is 'cloglog'. The 'power logit' has an attribute of 'power' for which the default is 1 i.e., a logit link.
`ntrials`	This is a scalar representing the denominator i.e., the length of the probability mass function returned is this scalar + 1.
`covariates.matrix.p`	A matrix of covariates for p where rows are the number of values in listbinary and columns the covariates. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.
`offset.p`	An offset vector for p. The default is a vector of ones.

Value

List of arguments input together with a list of probabilities vectors and a data frame of values of a and b of Equation (5) of Faddy and Smith (2012).

`model`	The model is either 'binomial' or 'EPPM extended binomial'.
`link`	The link is either 'logit' or 'cloglog'.
`parameter`	A vector of the parameters of the model which is set to initial estimates on function call.
`probabilities`	A list of the vectors of probabilities of the model.
`Dparameters`	A data frame of values of a and b of Equation (5) of Faddy and Smith (2012).

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Faddy M, Smith D. (2012). Extended Poisson Process Modeling and Analysis of Grouped Binary Data. *Biometrical Journal*, **54**, 426-435. doi:10.1002/bimj.201100214.

Examples

```
link <- 'cloglog'
attr(link, which="p") <- make.link(link)
parameter <- c(0.9423342,0.5846321)
names(parameter) <- c('p','b')
model.name <- 'EPPM extended binomial'
ntrials <- list(c(rep(10,11)))
Model.GB(parameter, model.name, link, ntrials,
         covariates.matrix.p = matrix(c(1),ncol=1),
         offset.p = c(0))
```

`Model.JMVGB` *Probabilities for EPPM extended binomial distributions given p's and scale-factors.*

Description

Calculates the probabilities for binomial and generalized binomial given values for p's and scale-factors.

Usage

```
Model.JMVGB(parameter, model.name, link, ntrials,
            covariates.matrix.p, covariates.matrix.scalef,
            offset.p = c(rep(0, length(ntrials))),
            offset.scalef = c(rep(0, length(ntrials))))
```

Arguments

`parameter`	A vector of the parameters of the model which is set to initial estimates on function call.
`model.name`	The model being fitted is one of the two 'binomial' or 'EPPM extended binomial'.
`link`	Takes one of nine values i.e., 'logit', 'probit', 'cloglog', 'cauchit', 'log', 'loglog', 'double exponential', 'double reciprocal', 'power logit'. The default is 'cloglog'. The 'power logit' has an attribute of 'power' for which the default is 1 i.e., a logit link.

`ntrials` This is a scalar representing the denominator i.e., the length of the probability mass function returned is this scalar + 1.

`covariates.matrix.p`
A matrix of covariates for p where rows are the number of values in listbinary and columns the covariates. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.

`covariates.matrix.scalef`
A matrix of covariates for the scale-factor where rows are the number of values in listbinary and columns the covariates. The default is a vector of ones. This matrix is extracted from the formulae in function BinaryEPPM. However, in the accompanying example it is shown how it can be constructed independently of function BinaryEPPM.

`offset.p` An offset vector for p. The default is a vector of ones.

`offset.scalef` An offset vector for the scale-factor. The default is a vector of ones.

Value

List of arguments input together with a list of probabilities vectors and a data frame of values of a and b of Equation (5) of Faddy and Smith (2012).

`model` The model is either 'binomial' or 'EPPM extended binomial'.

`link` The link is either 'logit' or 'cloglog'.

`parameter` A vector of the parameters of the model which is set to initial estimates on function call.

`probabilities` A list of the vectors of probabilities of the model.

`Dparameters` A data frame of values of a and b of Equation (5) of Faddy and Smith (2012).

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Faddy M, Smith D. (2012). Extended Poisson Process Modeling and Analysis of Grouped Binary Data. *Biometrical Journal*, **54**, 426-435. doi:10.1002/bimj.201100214.

Examples

```
link <- 'cloglog'
attr(link, which="p") <- make.link(link)
parameter <- c(-0.68294630,0.03451481)
names(parameter) <- c('p','scale-factor')
model.name <- 'EPPM extended binomial'
ntrials <- list(c(rep(10,11)))
Model.JMVGB(parameter, model.name, link, ntrials,
            covariates.matrix.p = matrix(c(1),nrow=1),
            covariates.matrix.scalef = matrix(c(1),nrow=1),
            offset.p = c(0), offset.scalef = c(0))
```

negcomplog *Negative complementary log-log Link Function*

Description

Computes the negative complementary log-log link function, including its inverse.

Usage

```
negcomplog()
```

Value

The negative complementary log-log of theta.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Tibshirani RJ, Ciampi A (1983). "A Family of Proportional- and Additive-Hazards Models for Survival Data". Biometrics 39(1), 141-147.

Parkes.litters *The data are of the number of male piglets born in litters of varying sizes for the Parkes breed of pigs.*

Description

The data are arranged as a list of binomial frequency distributions where the listing is by litter size which is included both as a variate (vsize) and as a factor (fsize)

Usage

```
data("Parkes.litters")
```

Format

The format is: List of 3 $ fsize : Factor w/ 7 levels " size 5"," size 6",..: 1 2 3 4 5 6 7 $ vsize : int [1:7] 5 6 7 8 9 10 11 $ number.success:List of 7 ..$: num [1:6] 2 20 41 35 14 4 ..$: num [1:7] 3 16 53 78 53 18 0 ..$: num [1:8] 0 21 63 117 104 46 21 2 ..$: num [1:9] 1 8 37 81 162 77 30 5 1 ..$: num [1:10] 0 2 23 72 101 83 46 12 7 0 ..$: num [1:11] 0 7 8 19 79 82 48 24 10 0 ... ..$: num [1:12] 0 1 3 15 15 33 13 12 8 1 ...

Source

Brooks, R.J., James, W.H., Gray, E. (1993). Modelling Sub-Binomial Variation in the Frequency of Sex Combinations in Litters of Pigs. Biometrics 47, 403-417.

Examples

```
data("Parkes.litters")
```

`plot.BinaryEPPM` *Diagnostic Plots for BinaryEPPM Objects*

Description

This function is generic. Various types of standard diagnostic plots can be produced, involving various types of residuals, influence measures etc. It is a minorly modified version of the generic plot function of **betareg** with details of the displays given in Cribari-Neto and Zeileis (2010). The same six displays and arguments list as in Cribari-Neto and Zeileis (2010) are used. The six displays are "Residuals vs indices of obs", "Cook's distance plot", "Leverage vs predicted values", "Residuals vs linear predictor", "Normal Q-Q plot of residuals", "Predicted vs observed values".

Usage

```
## S3 method for class 'BinaryEPPM'
plot(x, which = 1:4,
    caption = c("Residuals vs indices of obs.", "Cook's distance plot",
    "Leverage vs predicted values", "Residuals vs linear predictor",
    "Normal Q-Q plot of residuals", "Predicted vs observed values"),
    sub.caption = " ", main = "",
    ask = prod(par("mfcol"), 1) < length(which) && dev.interactive(),
    ..., type = "spearson")
```

Arguments

`x`	fitted model object of class "BinaryEPPM".
`which`	numeric. If a subset of plots is required, specify a subset of the numbers 1:6.
`caption`	character. Captions to appear above the plots.
`sub.caption`	character. Common title-above figures if there are multiple.
`main`	character. Title to each plot in addition to the above caption.
`ask`	logical. If true, the user is asked before each plot.
`...`	other parameters to be passed through to plotting functions.
`type`	character indicating type of residual to be used, see residuals.BinaryEPPM.

Details

The plot method for BinaryEPPM objects produces various plots of diagnostic plots similar to those produced by **betareg**. See Cribari-Neto and Zeileis (2010) for further details of the displays of **betareg**.

Value

No return value.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

plot.betareg

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution),
                  model.type = 'p only', model.name = 'binomial')
plot.BinaryEPPM(output.fn, which = 1, type= "sdeviance")
```

---

powerlogit *Power Logit Link Function*

Description

Computes the power logit link function, including its inverse.

Usage

```
powerlogit(power = 1)
```

Arguments

power	power value for the power logit link function.

Value

The power logit transformation of theta. All logarithms are natural ones, i.e., to base e.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Gaudard MA, Karson MJ, Linder E, Tse Sk (1993). Efficient Designs for Estimation in the Power Logistic Quantal Response Model." Statistica Sinica, 3, 233-243.

---

predict.BinaryEPPM *Prediction Method for BinaryEPPM Objects*

Description

Extract various types of predictions from BinaryEPPM regression models.

Usage

```
## S3 method for class 'BinaryEPPM'
predict(object, newdata = NULL, type = c("response",
     "linear.predictor.p", "linear.predictor.scale.factor",
 "p", "scale.factor", "scale.factor.limits", "mean",
 "variance",  "distribution", "distribution.parameters"), na.action = na.pass, ...)
```

Arguments

object	fitted model object of class "BinaryEPPM".
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the original observations are used.
type	character indicating type of predictions: fitted means of responses ("response"), linear predictors ("linear.predictor.p", "linear.predictor.scale.factor"), fitted value of probability of success ("p"), fitted value of scale-factor ("scale.factor"), fitted value of mean ("mean"), scale factor limits ("scale.factor.limits"), fitted value of variance ("variance"), fitted probability distribution ("distribution"), parameters of fitted distributions ("distribution.parameters")
na.action	function determining what should be done with missing values in *newdata*. The default is to predict NA.
...	some methods for this generic function require additional arguments.

Value

A vector or list of the predicted values from the fitted model object.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

predict.betareg

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution),
                  model.type = 'p only', model.name = 'binomial')
predict(output.fn, type = "response")
predict(output.fn, type = "linear.predictor.p")
```

---

print.BinaryEPPM *Printing of BinaryEPPM Objects*

Description

Prints objects of class "BinaryEPPM".

Usage

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

Arguments

x	fitted model object of class "BinaryEPPM".
digits	digits of printed output.
...	not currently used.

Value

An object of class "BinaryEPPM" is constructed. This object has the following attributes.

data.type	Indicator of the type of data either 0 "data.frame" or 1 "list".
list.data	Regardless of the "data.type", the data in list form.
call	The "call" to the function "BinaryEPPM".
formula	The model formula in "call".
model.type	The model type in "call".
model.name	The model name in "call".
link	The link function in "call".

`covariates.matrix.p`
: The matrix of covariates for the model for p.

`covariates.matrix.scalef`
: The matrix of covariates for the model for scale-factor.

`offset.p`
: The vector of offsets for the model for p.

`offset.scalef`
: The vector of offsets for the model for scale-factor.

`coefficients`
: The coefficients of the fitted model.

`loglik`
: The log-likelihood of the fitted model.

`vcov`
: The variance-covariance matrix of the fitted model.

`n`
: The number of observations. Relabelled duplication of "nobs" needed when calling function "lrtest".

`nobs`
: The number of observations.

`df.null`
: The degrees of freedom of the null model.

`df.residual`
: The degrees of freedom of the residual model.

`vnmax`
: Vector of number of "trials" in each observation.

`weights`
: Vector of weights for observation.

`converged`
: Indicator of convergence.

`method`
: Method used to calculate pseudo.r.squared.

`pseudo.r.squared`
: The value of the coefficient of determination r squared.

`start`
: Initial estimates.

`optim`
: Final model fit.

`control`
: Control parameters for optimization function "optim".

`fitted.values`
: The fitted values.

`y`
: The dependent variable in the model.

`terms`
: The terms in the model.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

`betareg`

Examples

```
data("ropespores.case")
BinaryEPPM(data = ropespores.case,
           number.spores / number.tested ~ 1 + offset(logdilution),
           model.type = 'p only', model.name = 'binomial')
```

print.summaryBinaryEPPM *Printing of summaryBinaryEPPM Objects*

Description

Prints the objects of class "summaryBinaryEPPM".

Usage

```
## S3 method for class 'summaryBinaryEPPM'
print(x, ...)
```

Arguments

x	object output by summary.BinaryEPPM.
...	not currently used.

Value

No return value.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

betareg

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution),
                  model.type = 'p only', model.name = 'binomial')
print(summary(output.fn))
```

`residuals.BinaryEPPM` *Residuals for BinaryEPPM Objects*

Description

This function is generic. Extract various types of residuals from objects of class "BinaryEPPM".

Usage

```
## S3 method for class 'BinaryEPPM'
residuals(object, type = c("spearson", "deviance", "pearson",
        "response", "likelihood", "sdeviance"), ...)
```

Arguments

`object`	Fitted model object of class "BinaryEPPM".
`type`	Type of residuals wanted i.e., standardized Pearson "spearson", deviance "deviance", Pearson "pearson",response "response", likelihood "likelihood", standardized deviance "sdeviance".
`...`	Some methods for this generic function require additional arguments.

Details

Residuals as Cribari-Neto and Zeileis (2010).

Value

An vector of class "numeric" of residuals of a specified type from the object of class "BinaryEPMM".

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

`residuals.betareg`

ropespores.case *Dilution series for the presence of rope spores.*

Description

Dilution series where at each dilution of a suspension of potato flour a number of samples were examined for the presence of rope spores. These data are in data frame form.

Usage

```
data("ropespores.case")
```

Format

A data frame with 10 observations on the following 5 variables.

`vdilution` a numeric vector

`fdilution` a factor with levels `0.25` `0.5` `1` `2` `4` `8` `16` `32` `64` `128`

`logdilution` a numeric vector

`number.spores` a numeric vector

`number.tested` a numeric vector

Source

Finney, D.J. (1971). Statistical Methods in Biological Assay. Griffin, London, 2nd edition.

Examples

```
data("ropespores.case")
```

ropespores.grouped *Dilution series for the presence of rope spores.*

Description

Dilution series where at each dilution of a suspension of potato flour a number of samples were examined for the presence of rope spores. These data are in list form.

Usage

```
data("ropespores.grouped")
```

Format

The format is: List of 4 $ vdilution : num [1:10] 0.25 0.5 1 2 4 8 16 32 64 128 $ fdilution : Factor w/ 10 levels "0.25","0.5","1",..: 1 2 3 4 5 6 7 8 9 10 $ offset.p : num [1:10] 1.386 0.693 0 -0.693 -1.386 ... $ number.spores:List of 10 ..$: num [1:6] 0 0 0 0 0 1 ..$: num [1:6] 0 0 0 0 0 1 ..$: num [1:6] 0 0 0 0 0 1 ..$: num [1:6] 0 0 0 0 0 1 ..$: num [1:6] 0 0 0 0 1 0 ..$: num [1:6] 0 0 0 1 0 0 ..$: num [1:6] 0 0 1 0 0 0 ..$: num [1:6] 0 0 1 0 0 0 ..$: num [1:6] 1 0 0 0 0 0 ..$: num [1:6] 1 0 0 0 0 0

Source

Finney, D.J. (1971). Statistical Methods in Biological Assay. Griffin, London, 2nd edition.

Examples

```
data("ropespores.grouped")
```

---

summary.BinaryEPPM *Summary of BinaryEPPM Objects*

Description

This function is generic. Summary of objects of class "BinaryEPPM".

Usage

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

Arguments

object	Fitted model object of class "BinaryEPPM".
...	some methods for this generic function require additional arguments.

Details

Similar output to that of summary.glm "summary.glm" and summary.betareg Cribari-Neto and Zeileis (2010).

Value

An object of class "summaryBinaryEPPM" is constructed. This object has the following attributes.

data.type	Indicator of the type of data either 0 "data.frame" or 1 "list".
call	The "call" to the function "BinaryEPPM".
formula	The model formula in "call".
model.type	The model type in "call".
model.name	The model name in "call".

`link`	The link function in "call".
`offset.p`	The vector of offsets for the model for p.
`offset.scalef`	The vector of offsets for the model for scale-factor.
`coeff.table.p`	The coefficients of the fitted model for p.
`coeff.table.scalef`	The coefficients of the fitted model for scale-factor.
`loglik`	The log-likelihood of the fitted model.
`n`	The number of observations. Relabelled duplication of "nobs" needed when calling function "lrtest".
`nobs`	The number of observations.
`df.null`	The degrees of freedom of the null model.
`df.residual`	The degrees of freedom of the residual model.
`vnmax`	Vector of number of "trials" in each observation.
`weights`	Vector of weights for observation.
`converged`	Indicator of convergence.
`method`	Method used to calculate pseudo.r.squared.
`pseudo.r.squared`	The value of the coefficient of determination r squared.
`optim`	Final model fit.
`control`	Control parameters for optimization function "optim".
`fitted.values`	The fitted values.
`y`	The dependent variable in the model.
`terms`	The terms in the model.
`npar`	The number of parameters in the model.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

`summary.betareg` `print.summaryBinaryEPPM`

vcov.BinaryEPPM *Variance/Covariance Matrix for Coefficients*

Description

Variance/covariance matrix for coefficients of fitted model.

Usage

```
## S3 method for class 'BinaryEPPM'
vcov(object, model = c("full", "p", "scale.factor"), ...)
```

Arguments

object	fitted model object of class "BinaryEPPM".
model	character indicating variance/covariance matrix for all coefficients to be output: all coefficients ("full"), variance/covariance matrix for coefficients of probability of success ("p"), variance/covariance matrix for coefficients of scale-factor ("scale.factor")
...	other parameters to be passed through to plotting functions.

Value

The variance/covariance matrix of the parameters of the fitted model object.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

See Also

betareg

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
                  number.spores / number.tested ~ 1 + offset(logdilution),
                  model.type = 'p only', model.name = 'binomial')
vcov(output.fn)
```

`waldtest.BinaryEPPM` *Wald Test of Nested Models for BinaryEPPM Objects*

Description

`waldtest` is a generic function for comparisons of nested (generalized) linear models via Wald tests.

Usage

```
## S3 method for class 'BinaryEPPM'
waldtest(object, ..., vcov = NULL,
   test = c("Chisq", "F"))
```

Arguments

`object`	an object of class "BinaryEPPM".
`...`	further object specifications passed to methods. See below for details.
`vcov`	a function for estimating the covariance matrix of the regression coefficients. If only two models are compared it can also be the covariance matrix of the more general model.
`test`	character specifying whether to compute the large sample Chi-squared statistic (with asymptotic Chi-squared distribution) or the finite sample F statistic (with approximate F distribution).

Details

`waldtest` is a generic function for comparisons of nested (generalized)linear models via Wald tests. It does not have the same functionality as the versions of **betareg** and **lmtest** with a reduced list of arguments. With these caveats, more details can be obtained from the **Details** pages of those packages.

Value

An object of class `"anova"` which contains the residual degrees of freedom, the difference in degrees of freedom, Wald statistic (either `"Chisq"` or `"F"`) and corresponding p value.

Author(s)

David M. Smith <dmccsmith@verizon.net>

References

Cribari-Neto F, Zeileis A. (2010). Beta Regression in R. *Journal of Statistical Software*, **34**(2), 1-24. doi:10.18637/jss.v034.i02.

Zeileis A, Hothorn T. (2002). Diagnostic Checking in Regression Relationships. *R News*, **2**(3), 7-10. https://CRAN.R-project.org/doc/Rnews/.

See Also

waldtest betareg

Examples

```
data("ropespores.case")
output.fn <- BinaryEPPM(data = ropespores.case,
  number.spores / number.tested ~ 1 + offset(logdilution),
  model.type = 'p only', model.name = 'binomial')
output.fn.one <- BinaryEPPM(data = ropespores.case,
  number.spores / number.tested ~ 1 + offset(logdilution),
  model.type = 'p only', model.name = 'beta binomial')
waldtest.BinaryEPPM(output.fn, output.fn.one, test = c("Chisq", "F"),
  vcov =  vcov)
```

---

wordcount.case *Number of occurences of an article in five-word and ten-word samples from two authors.*

Description

The data are the number of occurences of an article in five-word and ten-word samples from Macaulay's 'Essay on Milton' and G.K. Chesterton's essay 'About the workers'.

Usage

```
data("wordcount.case")
```

Format

A data frame with 340 observations on the following 5 variables.

`author` a factor with levels `Macaulay` `Chesterton`

`fsize` a factor with levels `5` `10`

`vsize` a numeric vector

`number.words` a numeric vector

`number.tested` a numeric vector

Source

Bailey, B.J.R. (1990). A model for Function Word Counts. Appl. Statist. 39(1), 107-114.

References

Sellers, K.F., Swift, A.W., Weems, K.S. (2017). A flexible distribution class for count data. Journal of Statistical Distributions and Applications 41(12), 2616-2626.

Examples

```
data(wordcount.case)
```

`wordcount.grouped` *Number of occurences of an article in five-word and ten-word samples from two authors.*

Description

The data are the number of occurences of an article in five-word and ten-word samples from Macaulay's 'Essay on Milton' and G.K. Chesterton's essay 'About the workers'.

Usage

```
data("wordcount.grouped")
```

Format

The format is: List of 4 $ author : Factor w/ 2 levels " Macaulay"," Chesterton": 1 1 2 2 $ fsize : Factor w/ 2 levels "5","10": 1 2 1 2 $ vsize : num [1:4] 5 10 5 10 $ number.words:List of 4 ..$: num [1:6] 45 49 6 0 0 0 ..$: num [1:11] 27 44 26 3 0 0 0 0 0 0 0 ... ..$: num [1:6] 32 35 3 0 0 0 ..$: num [1:11] 14 38 16 2 0 0 0 0 0 0 0 ...

Source

Bailey, B.J.R. (1990). A model for Function Word Counts. Appl. Statist. 39(1), 107-114.

References

Sellers, K.F., Swift, A.W., Weems, K.S. (2017). A flexible distribution class for count data. Journal of Statistical Distributions and Applications 41(12), 2616-2626.

Examples

```
data(wordcount.grouped)
```

`Yorkshires.litters` *The data are of the number of male piglets born in litters of varying sizes for the Yorkshire breed of pigs.*

Description

The data are arranged as a list of binomial frequency distributions where the listing is by litter size which is included both as a variate (vsize) and as a factor (fsize)

Usage

```
data("Yorkshires.litters")
```

Format

The format is: List of 3 $ fsize : Factor w/ 9 levels " size 5"," size 6",..: 1 2 3 4 5 6 7 8 9 $ vsize : int [1:9] 5 6 7 8 9 10 11 12 13 $ number.success:List of 9 ..$: num [1:6] 3 22 30 37 13 5 ..$: num [1:7] 7 18 44 62 27 17 4 ..$: num [1:8] 2 14 25 63 69 41 12 5 ..$: num [1:9] 2 15 32 70 127 90 45 18 1 ..$: num [1:10] 0 8 33 63 106 115 62 30 11 1 ..$: num [1:11] 0 3 20 49 79 119 91 59 23 4 ... ..$: num [1:12] 0 0 7 20 60 94 100 47 31 9 ... ..$: num [1:13] 0 1 6 16 29 52 66 43 34 22 ... ..$: num [1:14] 0 2 2 2 14 19 44 45 22 13 ...

Source

Brooks, R.J., James, W.H., Gray, E. (1993). Modelling Sub-Binomial Variation in the Frequency of Sex Combinations in Litters of Pigs. Biometrics 47, 403-417.

Examples

```
data("Yorkshires.litters")
```

Index