Package ‘BinaryEPPM’

October 12, 2022

Type Package

Title Mean and Variance Modeling of Binary Data

Version 2.3

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

Date 2019-07-30

Author David M Smith, Malcolm J Faddy

Maintainer David M. Smith <`smithdm1@us.ibm.com`>

Depends R (>= 3.5.0)

Description Modeling under- and over-dispersed binary data using extended Poisson process models (EPPM) as in the article Faddy and Smith (2012) <doi:10.1002/bimj.201100214> .

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2019-07-31 11:20:06 UTC

R topics documented:

`BinaryEPPM-package` *Mean and Variance Modeling of Binary Data*

Description

Modeling under- and over-dispersed binary data using extended Poisson process models (EPPM) as in the article Faddy and Smith (2012) <doi:10.1002/bimj.201100214> .

Details

The DESCRIPTION file:

Package:	BinaryEPPM
Type:	Package
Title:	Mean and Variance Modeling of Binary Data

Version:	2.3
Imports:	Formula, expm, numDeriv, stats, lmtest, grDevices, graphics
Date:	2019-07-30
Author:	David M Smith, Malcolm J Faddy
Maintainer:	David M. Smith <smithdm1@us.ibm.com>
Depends:	R (>= 3.5.0)
Description:	Modeling under- and over-dispersed binary data using extended Poisson process models (EPPM) as in the artic
License:	GPL-2

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 Variance Modeling of 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 generalized binomial distribution.`
`GasolineYield`	`Data on gasoline yields.`
`Hiroshima.case`	`Individual case data of chromosome aberrations in survivors of Hiroshima.`
`Hiroshima.grouped`	`Data of chromosome aberrations in survivors of Hiroshima grouped into dose ranges and represented as frequency distributions.`
`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.`
`Luningetal.litters`	`Number of trials (implantations) in data of Luning, et al., (1966)`
`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 generalized binomial distributions given p's and b.`
`Model.JMVGB`	`Probabilities for generalized 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.
`Titanic.survivors.case`	Individual case data of Titanic survivors
`Titanic.survivors.grouped`	Titanic survivors data in frequency distribution form.
`Williams.litters`	Number of implantations, data of Williams (1996).
`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
`foodstamp.case`	Participation in the federal food stamp program.
`foodstamp.grouped`	Participation in the federal food stamp program as a list not a data frame.
`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

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 regression models. Smith and Faddy (2019) gives further details about the package as well as examples of its use.

Author(s)

David M Smith, Malcolm J Faddy

Maintainer: David M. Smith <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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 and scale-factor",
       model.name = "generalized 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", "Faddy distribution", "beta binomial", "correlated binomial". If model.type is "p and scale-factor" the model being fitted is either "general" 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

`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 <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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 Goulet, Dutang, Maechler, Firth, Shapira, Stadelmann (2014) which depends on the package Matrix of Bates and Maechler (2012).

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 <smithdm1@us.ibm.com>

References

Bates D, Maechler M (2016). Matrix: Sparse and Dense Matrix Classes and Methods. R package version 1.2-4, `https://CRAN.R-project.org/package=Matrix`.

Goulet V, Dutang C, Maechler M, Firth D, Shapira M, Stadelmann M. (2014). Package 'expm'. R package version 0.99-1.1, `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.

Author(s)

David M. Smith <smithdm1@us.ibm.com>

See Also

`fitted`

`foodstamp.case` *Participation in the federal food stamp program.*

Description

The dependent variable is a binary variable indicating participation in the federal food stamp program. The independent variables are two binary ones i.e., tenancy and supplemental income, and a continuous one of the log(monthly income+1).

Usage

```
data("foodstamp.case")
```

Format

A data frame with 150 observations on the following 6 variables.

`participation` a numeric vector

`tenancy` a numeric vector

`suppl.income` a numeric vector

`income` a numeric vector

`n` a numeric vector

`weights1` a numeric vector

Source

Rousseeuw P, Croux C, Todorov V, Ruckstuhl A, Salibian-Barrera M, Verbeke T, Koller M, Maechler M (2016). ***robustbase**: Basic Robust Statistics*. R package version 0.92-6, `http://robustbase.r-forge.r-project.org/`.

References

Kunsch HR, Stefanski LA, Carroll RJ (1989). Conditionally Unbiased Bounded-Influence Estimation in General Regression Models, with Applications to Generalized Linear Models. *Journal of the American Statistical Association*, **84**(406), 460-466. doi: 10.1080/01621459.1989.10478791.

Examples

```
data(foodstamp.case)
```

`foodstamp.grouped` *Participation in the federal food stamp program as a list not a data frame.*

Description

The dependent variable is a list of frequency distributions of binary variables indicating participation in the federal food stamp program. The independent variables are two binary ones i.e., tenancy and supplemental income, and a continuous one of the log(monthly income+1).

Usage

```
data("foodstamp.grouped")
```

Format

The format is: List of 5 $ l.participation:List of 150 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 .. [list output truncated] $ tenancy : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 2 2 1 2 ... $ suppl.income : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 1 ... $ income : int [1:150] 271 287 714 521 0 518 458 1266 350 168 ... $ l.weights1 :List of 150 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 0.656 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 0.441 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2]

1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 0.127 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 0.647 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 0 1 ..$: num [1:2] 0 0.556 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 ..$: num [1:2] 1 0 .. [list output truncated]

Source

Rousseeuw P, Croux C, Todorov V, Ruckstuhl A, Salibian-Barrera M, Verbeke T, Koller M, Maechler M (2016).***robustbase****: Basic Robust Statistics*. R package version 0.92-6, `http://robustbase.r-forge.r-project.org/`.

References

Kunsch HR, Stefanski LA, Carroll RJ (1989). Conditionally Unbiased Bounded-Influence Estimation in General Regression Models, with Applications to Generalized Linear Models. Journal of the American Statistical Association, **84**(406), 460-466. doi: 10.1080/01621459.1989.10478791.

Examples

```
data(foodstamp.grouped)
```

`GasolineYield` *Data on gasoline yields.*

Description

Data on gasoline yields originally from Prater (1956). Used as an example in Cribari-Neto and Zeileis (2010).

Usage

```
data("GasolineYield")
```

Format

A data frame with 32 observations on the following 8 variables.

`percent` a numeric vector

`vthousand` a numeric vector

`gravity` a numeric vector

`pressure`	a numeric vector
`temp10`	a numeric vector
`temp`	a numeric vector
`batch`	a factor with levels `1` `2` `3` `4` `5` `6` `7` `8` `9` `10`

Source

Prater NH (1956). Estimate Gasoline Yields from crudes. *Petroleum Refiner*, **35**(5), 236-238.

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.

Examples

```
data(GasolineYield)
```

`GBprob` *Calculation of vector of probabilities for the generalized binomial distribution.*

Description

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

Usage

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

Arguments

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

Value

Vector of probabilities

Author(s)

David M. Smith <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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)
```

`Hiroshima.case` *Individual case data of chromosome aberrations in survivors of Hiroshima.*

Description

The data is of chromosome aberrations amongst survivors of the atomic bombs exploded over Japan. This data set is of individual survivors. The data is from Morel and Neechal (2012).

Usage

```
data("Hiroshima.case")
```

Format

A data frame with 648 observations on the following 6 variables.

`m` a numeric vector

`t` a numeric vector

`t65d_gamma` a numeric vector

`t65d_neutron` a numeric vector

`z` a numeric vector

`zz` a numeric vector

Details

The doses vectors z and zz have been standardized to mean of 0, standard deviation of 1 form.

Source

Morel JG, Neechal NK(2012). Overdispersion Models in SAS. SAS Press.

Examples

```
data("Hiroshima.case")
```

`Hiroshima.grouped` *Data of chromosome aberrations in survivors of Hiroshima grouped into dose ranges and represented as frequency distributions.*

Description

The data is of chromosome aberrations amongst survivors of the atomic bombs exploded over Japan. This data set is grouped into four frequency distributions. The data is from Prentice (1986).

Usage

```
data("Hiroshima.grouped")
```

Format

The format is: List of 5 $ fdose : Factor w/ 7 levels "0","1-99","100-199",..: 1 2 3 4 5 6 7 $ vdose : num [1:7] 0 38 144 244 347 ... $ gz : num [1:7] -0.7829 -0.5704 0.0219 0.5823 1.1573 ... $ gzz : num [1:7] 0.61293 0.32532 0.00048 0.3391 1.33927 ... $ number.aberrations:List of 7 ..$: num [1:101] 139 66 35 17 3 2 1 0 0 0 ... ..$: num [1:101] 20 23 6 7 3 2 5 2 0 1 ... ..$: num [1:101] 23 12 20 23 6 12 12 12 5 2 ... ..$: num [1:101] 2 2 5 5 3 14 3 2 4 3 ... ..$: num [1:101] 1 1 1 1 3 3 3 3 3 3 ... ..$: num [1:101] 0 0 0 1 1 3 4 1 0 2 ... ..$: num [1:101] 3 1 2 2 2 0 1 0 1 0 ...

Details

The grouped doses of Prentice (1986) have been standardized to gz and gzz with mean of 0, standard deviation of 1 form. This was done to match the doses used in the related individual survivor data form of the data set Hiroshima.case.Rd.

Source

Prentice RL (1986). "Binary Regression Using an Extended Beta-binomial Distribution, with Discussion of Correlation Induced by Covariate Measurement Errors." Journal of the American Statistical Association, 81, 321-327. doi: 10.1080/01621459.1986.10478275

Examples

```
data("Hiroshima.grouped")
```

`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', 'generalized binomial', 'beta binomial' or 'correlated binomial'. If model.type is 'p and scale-factor' the model being fitted is one of the three 'generalized binomial', 'beta binomial' or 'correlated binomial'. Information about these models is given in Faddy and Smith (2012). The default is 'generalized 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 <smithdm1@us.ibm.com>

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 = "generalized 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', 'generalized binomial', 'beta binomial' or 'correlated binomial'. If model.type is 'p and scale-factor' the model being fitted is one of the three 'generalized binomial', 'beta binomial' or 'correlated binomial'. Information about these models is given in Faddy and Smith (2012). The default is 'generalized 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 <smithdm1@us.ibm.com>

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 = "generalized 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 <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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.

`Luningetal.litters` *Number of trials (implantations) in data of Luning, et al., (1966)*

Description

The data are arranged as a list of frequency distributions of the number of female mice with 0,1,2, etc., dead where the number of implants 5 to 10, the number of implants being the length of the frequency distribution. Dose and number of implants are both included as both variates (vdose, vlitter.size) and as factors (fdose, flitter.size).

Usage

```
data("Luningetal.litters")
```

Format

The format is: List of 3 $ vdose : num [1:3] 0 300 600 $ fdose : Factor w/ 3 levels "0","300","600": 1 2 3 $ number.dead:List of 18 ..$: num [1:6] 30 27 9 5 0 0 ..$: num [1:6] 27 41 32 17 4 0 ..$: num [1:6] 16 32 48 49 15 0 ..$: num [1:7] 86 51 14 4 1 0 0 ..$: num [1:7] 28 47 59 28 6 1 1 ..$: num [1:7] 7 35 45 37 20 9 0 ..$: num [1:8] 111 73 31 8 1 0 0 0 ..$: num [1:8] 31 61 54 20 19 1 0 0 ..$: num [1:8] 5 22 27 36 17 9 3 1 ..$: num [1:9] 79 44 23 3 0 1 0 0 0 ..$: num [1:9] 12 32 24 22 8 1 0 0 0 ..$: num [1:9] 1 4 12 11 8 7 0 2 0 ..$: num [1:10] 32 29 8 1 0 0 0 0 0 0 ..$: num [1:10] 1 6 9 6 1 1 0 0 0 0 ..$: num [1:10] 0 0 2 2 2 0 1 0 0 0 ..$: num [1:11] 5 5 2 0 0 0 0 0 0 0 ... ..$: num [1:11] 1 2 1 0 0 0 0 0 0 0 ... ..$: num [1:11] 0 0 0 0 0 0 0 1 0 0 ...

Source

Luning K, Sheridan W, Ytterborn K, Gullberg U (1966). The relationship between the number of implantations and the rate of intra-uterine death in mice. Mutation Research, 3, 444-451.

Examples

```
data("Luningetal.litters")
```

`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 <smithdm1@us.ibm.com>

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.

Williams D (1996). Overdispersion in logistic linear models. In B Morgan (ed.), Statistics in Toxicology, 75-84. Oxford Science Publications.

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', 'generalized binomial', 'beta binomial' or 'correlated binomial'. If model.type is 'p and scale-factor' the model being fitted is one of the three 'generalized 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

Output which is the output from either Model.BCBinProb, Model.GB, Model.Binary, Model.JMVGB, or Model.ODB.

Author(s)

David M. Smith <smithdm1@us.ibm.com>

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 generalized binomial distributions given p's and b.*

Description

Calculates the probabilities for binomial and generalized 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 'generalized 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 'generalized 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 <smithdm1@us.ibm.com>

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 <- 'generalized 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 generalized 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 'generalized 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 'generalized 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 <smithdm1@us.ibm.com>

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 <- 'generalized 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 <smithdm1@us.ibm.com>

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

Various types of standard diagnostic plots can be produced, involving various types of residuals, influence measures etc.

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`	chaacter. 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 Ferrari and Cribari-Neto (2004) for further details of the displays of **betareg**.

Author(s)

David M. Smith <smithdm1@us.ibm.com>

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(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 <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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.

Author(s)

David M. Smith <smithdm1@us.ibm.com>

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.

Author(s)

David M. Smith <smithdm1@us.ibm.com>

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 geneic function require additional arguments.

Details

Residuals as Cribari-Neto and Zeileis (2010).

Author(s)

David M. Smith <smithdm1@us.ibm.com>

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).

Author(s)

David M. Smith <smithdm1@us.ibm.com>

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`

`Titanic.survivors.case` *Individual case data of Titanic survivors*

Description

These data are from the survival log of the Titanic and consist of the number of survivors out of the number of passengers broken down into age, sex and class categories.

Usage

```
data(Titanic.survivors.case)
```

Format

A data frame with 12 observations on the following 5 variables.

`age`	a factor with levels `child` `adult`
`number.passengers`	a numeric vector
`number.survive`	a numeric vector
`pclass`	a factor with levels `1st class` `2nd class` `3rd class`
`sex`	a factor with levels `female` `male`

Details

Hilbe (2011) first models these data as a logistic model, then finding that they are overdispersed, models them as count data (number of survivors, survive) with offset (log of the number of passengers, cases).

Source

Section 9.5, Example 3, pages 263-268, Hilbe, J. (2011).

References

Hilbe, J. (2011). Negative Binomial Regression. Cambridge University Press, second edition.

Examples

```
data(Titanic.survivors.case)
```

`Titanic.survivors.grouped`

Titanic survivors data in frequency distribution form.

Description

These data are from the survival log of the Titanic and consist of the number of survivors out of the number of passengers broken down into age, sex and class categories. The data are in frequency distribution form i.e., a distribution as a list of numbers surviving for each age, sex and class category.

Usage

```
data("Titanic.survivors.grouped")
```

Format

The format is: List of 4 $ age : Factor w/ 2 levels "child","adult": 1 2 1 2 1 2 1 2 1 2 ... $ sex : Factor w/ 2 levels "female","male": 1 1 2 2 1 1 2 2 1 1 ... $ pclass : Factor w/ 3 levels "1st class","2nd class",..: 1 1 1 1 2 2 2 2 3 3 ... $ number.survive:List of 12 ..$: num [1:2] 0 1 ..$: num [1:145] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:6] 0 0 0 0 0 1 ..$: num [1:176] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:14] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:94] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:12] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:169] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:32] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:166] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:49] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:463] 0 0 0 0 0 0 0 0 0 0 ...

Details

Hilbe (2011) first models these data as a logistic model, then finding that they are overdispersed, models them as count data (number of survivors, survive) with offset (log of the number of passengers, cases).

Source

Section 9.5, Example 3, pages 263-268, Hilbe, J. (2011).

References

Hilbe, J. (2011). Negative Binomial Regression. Cambridge University Press, second edition.

Examples

```
data(Titanic.survivors.grouped)
```

`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")
`...`	

Value

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

Author(s)

David M. Smith <smithdm1@us.ibm.com>

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 <smithdm1@us.ibm.com>

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

```
## Not run:
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(output.fn, output.fn.one, test = c("Chisq", "F"),
  vcov =  vcov)

## End(Not run)
```

`Williams.litters` *Number of implantations, data of Williams (1996).*

Description

The data is arranged as a list of frequency distributions of numbers of surviving foetuses for an given litter size. The litter size is the length of the frequency distribution. Dose is included both as a variate (vdose) and as a factor (fdose).

Usage

```
data("Williams.litters")
```

Format

The format is: List of 3 $ vdose : num [1:34] 0.75 1.5 1.5 0 0.75 1.5 3 0.75 1.5 3 ... $ fdose : Factor w/ 4 levels "0","0.75","1.5",..: 2 3 3 1 2 3 4 2 3 4 ... $ number.surviving:List of 34 ..$: num [1:8] 0 0 0 0 0 0 0 1 ..$: num [1:8] 0 0 0 1 0 0 0 0 ..$: num [1:9] 0 0 0 0 0 0 0 0 1 ..$: num [1:10] 0 0 0 0 0 0 0 1 1 0 ..$: num [1:10] 0 0 0 0 0 0 0 1 0 0 ..$: num [1:10] 0 0 0 0 0 0 0 2 0 0 ..$: num [1:10] 0 0 0 0 1 0 0 2 0 0 ..$: num [1:11] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:11] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:11] 0 0 0 0 1 0 0 0 0 1 ... ..$: num [1:12] 0 0 0 0 0 0 1 0 0 0 ... ..$: num [1:12] 0 0 0 0 0 0 0 0 0 1 ... ..$: num [1:12] 0 0 0 0 0 0 0 0 0 1 ... ..$: num [1:13] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:13] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:13] 0 0 0 0 0 0 0 0 0 1 ... ..$: num [1:13] 0 0 0 0 0 0 0 1 0 0 ... ..$: num [1:14] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:14] 0 0 0 0 0 0 0 0 1 0 ... ..$: num [1:14] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:14] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:15] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:15] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:15] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:15] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:16] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:16] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:16] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:16] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:17] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:17] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:17] 0 0 0 0 0 0 0 1 0 0 ... ..$: num [1:18] 0 0 0 0 0 0 0 0 0 0 ... ..$: num [1:20] 0 0 0 0 0 0 0 0 0 0 ...

Source

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

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

```
data("Williams.litters")
```

`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