Package 'PROreg'

June 13, 2020

Type Package

Title Patient Reported Outcomes Regression Analysis

Version 1.1

Author Josu Najera-Zuloaga <jnajera@deusto.es>, Dae-
Jin Lee <dlee@bcamath.org>, Inmaculada Arostegui <inmaculada.arostegui@ehu.eus>

Maintainer Josu Najera-Zuloaga <jnajera@deusto.es>

Description
Offers a variety of tools, such as specific plots and regression model approaches, for analyzing different patient reported questionnaires. Specially, mixed-effects models based on the beta-binomial distribution are implemented to deal with binomial data with over-dispersion (see Najera-Zuloaga J., Lee D.-J. and Arostegui I. (2017) <doi:10.1177/0962280217690413>).

License GPL

Imports fmsb, car, RColorBrewer, matrixcalc, rootSolve, numDeriv,
Matrix

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-13 15:00:06 UTC

R topics documented:

`BB` *The Beta-Binomial Distribution*

Description

Density and random generation for the beta-binomial distribution.

Usage

```
dBB(m,p,phi)
rBB(k,m,p,phi)
```

Arguments

`k`	number of simulations.
`m`	maximum socre number in each beta-binomial observation..
`p`	probability parameter of the beta-binomial distribution.
`phi`	dispersion parameter of the beta-binomial distribution.

Details

The beta-binomial distribution consists of a finite sum of Bernoulli dependent variables whose probability parameter is random and follows a beta distribution. Assume that we have y_j a set of variables, $j = 1, ..., m$, with m integer, that conditioned on a random variable u, are independent and follow a Bernoulli distribution with probability parameter u. On the other hand, the random variable u follows a beta distribution with parameter p/phi and $(1-p)/phi$. Namely,

$$y_j \sim Ber(u), u \sim Beta(p/phi, (1-p)/phi),$$

where $0 < p < 1$ and $phi > 0$. The first and second order marginal moments of this distribution are defined as

$$E[y_j] = p, Var[y_j] = p(1-p),$$

and correlation between observations is defined as

$$Corr[y_j, y_k] = phi/(1+phi),$$

where $j, k = 1, ..., m$ are different. Consequently, phi can be considered as a dispersion parameter.

If we sum up all the variables we will define a new variable which follows a new distribution that is called beta-binomial distribution, and it is defined as follows. The variable y follows a beta-binomial distribution with parameters m, p and phi if

$$y|u \sim Bin(m, u), u \sim Beta(p/phi, (1-p)/phi).$$

Value

`dBB` gives the density of a beta-binomial distribution with the defined `m`, `p` and `phi` parameters.

`rBB` generates `k` random observations based on a beta-binomial distribution with the defined `m`, `p` and `phi` parameters.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Arostegui I., Nunez-Anton V. & Quintana J. M. (2006): Analysis of short-form-36 (SF-36): The beta-binomial distribution approach, *Statistics in Medicine*, **26**, 1318-1342

See Also

The `rbeta` and `rbinom` functions of package `stats`.

Examples

```
set.seed(12)
# We define
m <- 10
p <- 0.4
phi <- 1.8

# We perform k beta-binomial simulations for those parameters.
k <- 100
bb <- rBB(k,m,p,phi)
bb
dd <- dBB(m,p,phi)

# We are going to plot the histogram of the created variable,
```

```
# and using dBB() function we are going to fit the distribution:
hist(bb,col="grey",breaks=seq(-0.5,m+0.5,1),probability=TRUE,
  main="Histogram",xlab="Beta-binomial random variable")
lines(seq(0,m),dd,col="red",lwd=4)
```

BBest *Estimation of the parameters of a beta-binomial distribution*

Description

This function performs the estimation of the parameters of a beta-binomial distribution for the given data and maximum score number in each observation.

There are two different approaches available for performing the estimation of the parameters: (i) Method of moments, and, (ii) maximum likelihood approach.

Usage

```
BBest(y,m,method="MM")
```

Arguments

y	response variable which folloes a beta-binomial distribution.
m	maximum score number in each beta-binomial observation.
method	the method used for performing the estimation of the probability and dispersion parameters of a beta-binomial distribution, "MM" for the method of moments and "MLE" for maximum likelihood estimation. Default "MM".

Details

BBest function performs estimations in the parameters of a beta-binomial distribution for the given data. The estimations can be performed using two different approaches, the methods of moments and the maximum likelihood estimation approach.

The density function of a given observation y that follows a beta-binomial distribution with paramters m, p and phi is defined as

$$f(y) = [\Gamma(1/phi)*\Gamma(p/phi+m)*\Gamma((1-p)/phi+m)]/[\Gamma(1/phi+m)*\Gamma(p/phi)*\Gamma((1-p)/phi)].$$

The first and second order moments are defined as

$$E[y] = mp$$

$$Var[y] = mp(1-p)[1+(n-1)phi/(1+phi)].$$

Hence, if $y = (y_1, ..., y_n)$ is the given data, we can conclude the method of moments from the previous as

$$p = E/m,$$

$$phi = [V - mp(1-p)]/[mp(1-p)m - V],$$

where E is the sample mean and V is the sample variance.

On the other hand, the maximum likelihood estimation of both parameters consits of solving the derivative of the log-likelihood defined by the density function with respect to each parameter and equaling them to zero. An iterative algorithm is needed for both parameter estimation as the score equations the parameters depend each other.

The variance of the estimation of the probability parameter of the beta-binomial distribution for the given data set is computed by the inverse of the Fisher information, i.e., the inverse of the negative second derivate of the log-likelihodd remplacing p by its estimation.

Value

`BBest` returns an object of class `"BBest"`.

The function `summary` (i.e., `summary.BBest`) can be used to obtain or print a summary of the results.

`p`	estimated probability parameter of the beta-binomial distribution.
`phi`	estimated dispersion parameter of the beta-binomial distribution.
`pVar`	variance of the estimation of the probability parameter `p`.
`psi`	estimation of the logarithm of the dispersion parameter `phi`.
`psiVar`	variance of the estimation of the logarithm of the dispersion parameter `psi`.
`m`	maximum score number in each beta-binomial observation.
`balanced`	if the response variable is balanced it returns "yes", otherwise "no".
`method`	the used approach for performing the estimations.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Arostegui I., Nunez-Anton V. & Quintana J. M. (2006): Analysis of short-form-36 (SF-36): The beta-binomial distribution approach, *Statistics in Medicine*, **26**, 1318-1342

Examples

```
# We simulate 1000 observations of a beta-binomial distribution
# for the fixed paramters.
m <- 10
k <- 1000
p <- 0.7
phi <- 1.6

set.seed(5)
y <- rBB(k,m,p,phi)
```

```
# Performing the estimation of the parameters

# Method of moments:
MM <- BBest(y,m)
MM

# Maximum likelihood approach
MLE <- BBest(y,m,method="MLE")
MLE
```

BBmm *Beta-binomial mixed-effects model*

Description

BBmm function performs beta-binomial mixed-effects models, i.e., it allows the inclusion of gaussian random effects in the linear predictor of a logitic beta-binomial regression model.

The structure of the random part of the model can be expecified by two different ways: (i) determining the random.formula argument, or (ii) especifying the model matrix of the random effects, Z, and determining the number of random effects in each random component, nRandComp.

The estimation of the parameters can also be done by means of two approaches: (i) BB-NR, special Newton-Raphson algorithm developed for beta-binomial mixed-effect models, and (ii) using the rootSolve R-package.

Usage

```
BBmm(fixed.formula,X,y,random.formula,Z=NULL,nRandComp=NULL,m,data,
      method="NR",maxiter=50,show=FALSE,nDim=1)
```

Arguments

fixed.formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the fixed part of the model to be fitted. It must be specified in cases where the model matrix of the fixed effects X and the outcome variable y are not specified.
X	a matrix class object containing the covariate structure of the fixed part of the model to be fitted. If the fixed.formula argument is specified this argument should not be defined, as we will be specifying twice the fixed structure of the model.
y	a vector containing the outcome variable(s). If joint analysis is expected, the outcome variables must be listed one after another in a vector.
random.formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the random part of the model to be fitted. It must be specified in cases where the model matrix of the random effects Z is not determined.

`Z`	the design matrix of the random effects. If the `random.formula` argument is specified this argument should not be specified, as we will be specifying twice the random structure of the model.
`nRandComp`	the number of random effects in each random component of the model. It must be especified as a vector, where the 'i'th value must correspond with the number of random effects of the 'i'th random component. It must be only determined when we specify the random structure of the model by the model matrix of the random effects, `Z`.
`m`	maximum score number in each beta-binomial observation.
`data`	an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula).
`method`	the methodology for performing the estimation of the parameters. Options "NR" or "Delta". Default "NR".
`maxiter`	the maximum number of iterations in the parameters estimation algorithm. Default 50.
`show`	logical parameter. If `TRUE`, the step by step optimization process will be shown in the screen. `FALSE` by default.
`nDim`	number of dimensions/dependent outcome variables involved in the multidimensional analysis. `nDim=1` by default.

Details

`BBmm` function performs beta-binomial mixed effects models. It extends the beta-binomial logistic regression to the inclusion of random effects in the linear predictor of the model. The model is defined as, conditional on some gaussian random effects u the response variable y follows a beta-binomial distribution of parameters m, p and phi,

$$y|u \sim BB(m, p, phi), u \sim N(0, D)$$

where

$$log(p/(1 - p)) = X * beta + Z * u$$

and D is determined by some dispersion parameters icluded in the parameter vector $theta$.

The estimation of the regression paramters $beta$ and the prediction of the random effects u is done via the extended likelihood, where the marginal likelihood is approximated to the h-likelihood by a first order Laplace approximation,

$$h = f(y|beta, u, theta) + f(u|theta)$$

The previous formula do not have a closed form and numerical methods are needed for the estimation precedure. Two approches are available in the function in order to perform the fixed and random effects estimation: (i) A special case of a Newton-Raphson algorithm developed for beta-binomial mixed-effects model estimations, and (ii) the general Newton-Raphson algorithm available in R-package `rootSolve`.

On the other hand, the estimation of dispersion parameters by the h-likelihood can be bias due to the estimation of the regression parameters. Consequenlty, a penalization of the h-likelihood must be performed to get an unbiased profile h-likelihood of the dispersion parameters. Lee and Nelder (1996) proposed the adjusted profile h-likelihood, where the following penalization is applied,

$$h(theta) = h + (1/2) * log[det(2\pi H^{-1})]$$

where H is the Hessian matrix of the model, and the terms $beta$ and u involved in the previous formula are replaced by their estimates.

The method iterates between both estimation processes until convergence is reached.

Value

`BBmm` returns an object of class `"BBmm"`.

The function `summary` (i.e., `summary.BBmm`) can be used to obtain or print a summary of the results..

`fixed.coef`	estimated value of the fixed coefficients of the regression.
`fixed.vcov`	variance and covariance matrix of the estimated fixed coefficients of the regression.
`random.coef`	predicted random effects of the regression.
`sigma.coef`	estimated value of the random effects variance parameters.
`sigma.var`	variance of the estimated value of the random effects variance parameters.
`phi.coef`	estimated value of the dispersion parameter of the conditional beta-binomial distribution.
`psi.coef`	estimated value of the logrithm of the dispersion parameter of the conditional beta-binomial distribution.
`psi.var`	variance of the estimation of the logarithm of the conditional beta-binomial distribution.
`fitted.values`	the fitted mean values of the probability parameter of the conditional beta-binomial distribution.
`conv`	convergence of the methodology. If the method has converged it returns "yes", otherwise "no".
`deviance`	deviance of the model.
`df`	degrees of freedom of the model.
`null.deviance`	null-deviance, deviance of the null model. The null model will only include an intercept as the estimation of the probability parameter.
`null.df`	degrees of freedom of the null model.
`nRand`	number of random effects.
`nComp`	number of random components.
`nRandComp`	number of random effects in each random component of the model.
`namesRand`	names of the random components.
`iter`	number of iterations in the estimation method.
`nObs`	number of observations in the data.
`y`	dependent response variable in the model.

`X`	model matrix of the fixed effects.
`Z`	model matrix of the random effects.
`D`	variance and covariance matrix of the random effects.
`balanced`	if the response dependent variable is balanced it returns "yes", otherwise "no".
`m`	maximum score number in each beta-binomial observation.
`call`	the matched call.
`formula`	the formula supplied.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Breslow N. E. & Calyton D. G. (1993): Approximate Inference in Generalized Linear Mixed Models, *Journal of the American Statistical Association*, **88**, 9-25

Lee Y. & Nelder J. A. (1996): Hierarchical generalized linear models, *Journal of the Royal Statistical Society. Series B*, **58**, 619-678

Najera-Zuloaga J., Lee D.-J. & Arostegui I. (2017): Comparison of beta-binomial regression model approaches to analyze health related quality of life data, *Statistical Methods in Medical Research*. DOI: 10.1177/0962280217690413

See Also

The `multiroot` and `uniroot` functions of the R-package `rootSolve` for the general Newton-Raphson algorithm.

Examples

```
set.seed(14)

# Defining the parameters
k <- 100
m <- 10
phi <- 0.5
beta <- c(1.5,-1.1)
sigma <- 0.5

# Simulating the covariate and random effects
x <- runif(k,0,10)
X <- model.matrix(~x)
z <- as.factor(rBI(k,4,0.5,2))
Z <- model.matrix(~z-1)
u <- rnorm(5,0,sigma)
```

```
# The linear predictor and simulated response variable
eta <- beta[1]+beta[2]*x+crossprod(t(Z),u)
p <- 1/(1+exp(-eta))
y <- rBB(k,m,p,phi)
dat <- data.frame(cbind(y,x,z))
dat$z <- as.factor(dat$z)

# Apply the model
model <- BBmm(fixed.formula = y~x,random.formula = ~z,m=m,data=dat)
model
```

BBreg *Fit a beta-binomial logistic regression model*

Description

BBreg function fits a beta-binomial logistic regression model, i.e., it links the probability parameter of a beta-binomial distribution with the given covariates by means of a logistic link function. The estimation of the parameters in the model is done via maximum likelihood estimation.

Usage

```
BBreg(formula,m,data,maxiter=100)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
m	maximum score number in each beta-binomial observation.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula).
maxiter	the maximum number of iterations in the estimation process. Default 100.

Details

There are two different ways of defining a regression model based on the beta-binomial distribution: (i) the marginal regression approach, (ii) hierarchical generalized linear model approach. *Najera-Zuloaga et al. (2017)* proved that the first approach is more adequate when the interest consists of the interpretation of the regression coefficients. Consequenlty, this function is based on the first approach, i.e., the marginal regression approach.

Once the marginal density function of the beta-binomial distribution is explicity calculated, we connect the probability parameter with the given covariates by means of a logistic link function:

$$logit(p) = log(p/(1-p)) = X * beta$$

where X a model matrix composed by the given covariates and $beta$ are the regression coefficients of the model.

Remplacing the previous linear predictor in the marginal density function of the beta-binomial distribution, we can derive maximum likelihood estimations of both regression and dispersion parameters. Forcina and Franconi (1988) presented an estimation algorithm based on the Newton-Raphson approach. This function performs the estimation of the parameters following the presented methodology.

Value

`BBreg` returns an object of class `"BBreg"`.

The function `summary` (i.e., `summary.BBreg`) can be used to obtain or print a summary of the results.

- `coefficients` the estimated value of the regression coefficients.
- `vcov` the variance-covariance matrix of the estimated coefficients of the regression.
- `phi` the estimation of the dispersion parameter of the beta-binomial distribution.
- `psi` the estimation of the logarithm of the dispersion parameter of the beta-binomial distribution.
- `psi.var` the variance of the estimated logarithm of the dispersion parameter of the beta-binomial distribution.
- `conv` convergence of the methodology. If the method has converged it returns "yes", otherwise "no".
- `fitted.values` the fitted mean values of the model.
- `deviance` the deviance of the model.
- `df` degrees of freedom of the model.
- `null.deviance` null-deviance, the deviance for the null model. The null model will only include an intercept as the estimation of the probability parameter.
- `null.df` the degrees of freedom for the null model.
- `iter` number of iterations in the estimation process.
- `X` the model matrix.
- `y` the dependent response variable in the model.
- `m` maximum score number in each beta-binomial observation.
- `balanced` if the response beta-binomial variable is balanced it returns "yes", otherwise "no".
- `nObs` number of observations.
- `call` the matched call.
- `formula` the formula supplied.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Forcina A. & Franconi L. (1988): Regression analysis with Beta-Binomial distribution, *Revista di Statistica Applicata*, **21**, 7-12

Najera-Zuloaga J., Lee D.-J. & Arostegui I. (2017): Comparison of beta-binomial regression model approaches to analyze health related quality of life data, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280217690413

Examples

```
# We simulate a covariate, fix the paramters of the beta-binomial
# distribution and simulate a response variable.

# Then we apply the model, and try to get the same values.
set.seed(18)
k <- 1000
m <- 10
x <- rnorm(k,5,3)

beta <- c(-10,2)
p <- 1/(1+exp(-(beta[1]+beta[2]*x)))
phi <- 1.2

y <- rBB(k,m,p,phi)

model <- BBreg(y~x,m)
model
```

BI — *The Binomial distribution with optional Dispersion Parameter*

Description

Density and random generation for the binomial distribution with optional dispersion parameter.

Usage

```
dBI(m,p,phi)
rBI(k,m,p,phi)
```

Arguments

k	number of simulations.
m	number of trials in each binomial observation.
p	probability parameter of the binomial distribution.
phi	dispersion parameter of the binomial distribution. If phi=1, the binomial distribution without dispersion parameter will be considered. Default 1.

Details

The binomial distribution belongs to the exponential family of distributions. Consequenlty, although the usual binomial distribution only consists of two paramters, an additional dispersion parameter can be included. The inclusion of a dispersion parameter softens the relationship between the expectation and variance that the binomial distribution keeps, i.e. the model allows overdispersion to be included,

$$E[y] = mp, Var[y] = phi * mp(1 - p).$$

The density function of the binomial distribution with dispersion parameter is based on the exponential family approach and it is defined as

$$f(y) = exp\{[y * log(p/(1 - p)) + m * log(1 - p)]/phi + c(y, phi)\},$$

where $c()$ is a function that it is approximated with the deviance of the model by quadratic approximations of the log-likelihood function.

Value

`dBI` gives the density of the binomial distribution for those `m`, `p` and `phi` parameters.

`rBI` generates `k` random observations based on a binomial distribution for those `m`, `p` and `phi` parameters.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Pawitan Y. (2001): *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford University Press

See Also

The `rbinom` functions of package `stats`. This function performs simulations based on a binomial distribution without dispersion parameter.

Examples

```
k <- 1000
m <- 10
p <- 0.765
phi <- 4.35

#simulating
y <- rBI(k,m,p,phi)
y

#density function
```

```
d <- dBI(m,p,phi)
d

#plot the simulated variable and fit the density
hist(y,col="lightgrey")
lines(seq(0,m),k*d,col="blue",lwd=2)
```

BIest *Estimation of the parameters of a binomial distribution with optional dispersion parameter.*

Description

BIest function estimates the probability parameter of a binomial distribution for the given data and number of trials. It is possible to include a dispersion parameter in the binomial distribution and get the estimation by the method of moments or maximum quasi-likelihood approach. This function also returns the standard deviation of the estimated probability parameter and the upper and lower bounds of the 95% confidence interval.

Usage

```
BIest(y,m,disp=FALSE,method="MM")
```

Arguments

y	response variable wich follows a binomial distribution.
m	number of trials in each binomial observation.
disp	dispersion parameter of the binomial distribution. If phi=FALSE, then the binomial distribution without dispersion parameter will be considered for estimation. Default FALSE.
method	the method used for estimating the parameters, "MM" for the method of moments and "MLE" for maximum quasi-likelihood. Default "MM".

Details

This function performs the estimation of the parameters involved in a binomial distribution for a given data.

The estimation of the probability parameter is done by either maximum likelihood approach or method of moments due to the fact that both approaches give the same estimation,

$$p = sum(y)/(m * n),$$

where m is the number of trials and n is the number of observations.

If the dispersion parameter is included in the model, BIest function performs its estimation by the method of moments or maximum quasi-likelihood methodology. The method of moments is based on the variance equation of a binomial distribution with dispersion parameter

$$Var[y] = phi * mp(1 - p).$$

The maximum quasi-likelihood approach is based on the quadratic approximation of the log-likelihood function of a binomial distribution with dispersion parameter, where the unknown term involving phi is estimated with the deviance of the model.

The standard deviation of the estimated probability parameter is calculated by the Fisher information, i.e., the negative of the second derivative of the log-likelihood (log-quasi-likelihood) function.

Value

`BIest` returns an object of class "`BIest`".

`p`	estimation of the probability parameter. Both estimating approaches, the method of moments and the maximum likelihood estimation, perform the same estimation.
`pVar`	the variance of the estimated probability parameter.
`pIC.low`	the lower bound of the 95% confidence interval of the estimated probability parameter.
`pIC.up`	the upper bound of the 95% confidence interval of the estimated probability parameter.
`phi`	if the `disp` option is TRUE, it returns the estimated value of the dispersion parameter. Default FALSE.
`m`	number of trials in each binomial observation.
`balanced`	if the data is balanced it returns "yes", otherwise "no".
`method`	the used methodology for performing the estimation of the parameters.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Pawitan Y. (2001): *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, Oxford University Press

See Also

The `rBI` and `dBI` functions of package `HRQoL`. The functions perform simulations and estimate the density of a binomial distribution with optional dispersion parameter for a given set of parameters.

Examples

```
set.seed(9)
# We simulate the binomial data with some fixed parameters and
# then try to reach the same estimations.
m <- 10
k <- 100
```

```
p <- 0.654
y <- rBI(k,m,p) #Simulations

# without dispersion parameter
BIest(y,m)

# with dispersion parameter
# estimation by method of moments.
BIest(y,m,disp=TRUE,method="MM")
# estimation by maximum quasi-likelihood.
BIest(y,m,disp=TRUE,method="MLE")
```

`BIiwls` *Iterative Weighted Least Squares (IWLS) method for binomial logistic regression*

Description

`BIiwls` performs estimation of the coefficients of binomial logistic regressions by iterative wieghted least squares method.

Usage

```
BIiwls(y,X,m,maxiter)
```

Arguments

`y`	response dependent variable which follows a binomial distribution.
`X`	model matrix.
`m`	number of trials in each binomial observation.
`maxiter`	maximum number of iterations in the method.

Details

The iterative weighted least squares (IWLS) is a general algorithm to find the maximum likelihood estimations (mle) and standard deviations in generalized linear models. There are several ways to derive it, but the one that has been developed in this function is via the Newton-Raphson method. It consists of making a Taylor expansion in the score function, the first derivate of the log-likelihood, around the mle. This especific IWLS, `BIiwls`, has been developed to find out the mle and the standard errors in logistic regression by the introduction of a dependent variable, a matrix model of the regression covariates and the number of trials of the binomial dependent variable.

Value

`beta`	maximum likelihood estimations of the logistic regression coefficients.
`vcov`	variance-covariance matrix of the estimated regression coefficients.
`iter`	number of iterations of the algorithm.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

Examples

```
# We create a variable and a matrix model that consists of
# two covariates.
k=1000
m=10
maxiter=100

y <- rbinom(k,m,0.87)
x1 <- rnorm(k,1,50)
x2 <- rnorm(k,30,9)
X <- cbind(1,x1,x2)

BIiwls(y,X,m,maxiter)
```

BImm *Binomial Logistic Mixed-Effects Model Regression.*

Description

`BImm` function performs binomial logistic mixed-effects models, i.e., it allows the inclusion of gaussian random effects in the linear predictor of a logistic binomial regression model.

The structure of the random part of the model can be expecified by two different ways: (i) determining the `random.formula` argument, or (ii) especifying the model matrix of the random effects, `Z`, and determining the number of random effects in each random component, `nRandComp`.

Usage

```
BImm(fixed.formula,random.formula,Z=NULL,nRandComp=NULL,m,data,maxiter=100)
```

Arguments

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

`random.formula` an object of class `"formula"` (or one that can be coerced to that class): a symbolic description of the random part of the model to be fitted. It must be specified in cases where the model matrix of the random effects `Z` is not determined.

Z
: the design matrix of the random effects. If the random.formula argument is specified this argument should not be specified, as we will be specifying twice the random structure of the model.

nRandComp
: the number of random components/levels in each random effect of the model. It must be especified as a vector, where the 'i'th value must correspond with the number of random components of the 'i'th random effect. It must be only determined when we specify the random structure of the model by the model matrix of the random effects, Z.

m
: number of trials in each binomial observation.

data
: an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula).

maxiter
: the maximum number of iterations in the parameters estimation algorithm. Default 100.

Details

The model that is performed by this function is a especial case of generalized linear mixed models (GLMMs), in which conditioned on some random components the response variable has a binomial distribution. As in the binomial (logistic) regression a logit link function is applied to the probability parameter of the conditioned binomial distribution, allowing the inclusion of random effects in the linear predictor,

$$logit(p) = X * beta + Z * u,$$

where p is the probability parameter, X a full rank matrix composed by the covariables, $beta$ the fixed effects, Z the design matrix for the random effects ang u are the random effects. These random effects are independent and have a normal distribution with the same variance and mean 0.

The model estimates the fixed effects, predicts the random effects, and gets the estimation of the random effects variance parameters.

The estimation procedure is done by likelihood approximation, via iteartive weighted least squares method. The process is performed in two steps: (i) fixed and random parameters are estimated for some given values of the random effects variance parameters, and (ii) random effects variance parameters are estimated for some given regression and random coefficients using a penalized profile likelihood. The estimation approach iterates between (i) and (ii) until convergence is obtained.

Value

BImm returns an object of class "BImm".

The function summary (i.e., summary.BImm) can be used to obtain or print a summary of the results.

fixed.coef
: estimated value of the fixed coefficients in the regression.

fixed.vcov
: variance and covariance matrix of the estimated fixed coefficients in the regression.

random.coef
: predicted random effects of the regression.

sigma.coef
: estimated value of the random effects variance parameters.

`sigma.var`	variance of the estimated value of the random effects variance parameters.
`fitted.values`	the fitted mean values of the probability parameter of the conditional beta-binomial distribution.
`conv`	convergence of the methodology. If the method has converged it returns "yes", otherwise "no".
`deviance`	deviance of the model.
`df`	degrees of freedom of the model.
`nRand`	number of random effects.
`nComp`	number of random components.
`nRandComp`	number of random effects in each random component of the model.
`namesRand`	names of the random components.
`iter`	number of iterations in the estimation method.
`nObs`	number of observations in the data.
`y`	dependent response variable in the model.
`X`	model matrix of the fixed effects.
`Z`	model matrix of the random effects.
`balanced`	if the conditional beta-binomial response variable is balanced it returns "yes", otherwise "no".
`m`	maximum score number in each binomial observation.
`call`	the matched call.
`formula`	the formula supplied.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Breslow N. E. & Calyton D. G. (1993): Approximate Inference in Generalized Linear Mixed Models, *Journal of the American Statistical Association*, **88**, 9-25

McCulloch C. E. & Searle S. R. (2001): Generalized, Linear, and Mixed Models, *Jhon Wiley & Sons*

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

Examples

```
set.seed(5)
# Fixing parameters for the simulation:
nObs <- 1000
m <- 10
beta <- c(1.5,-1.1)
sigma <- 0.8

# Simulating the covariate:
x <- runif(nObs,-5,5)

# Simulating the random effects:
z <- as.factor(rBI(nObs,5,0.5,2))
u <- rnorm(6,0,sigma)

# Getting the linear predictor and probability parameter.
X <- model.matrix(~x)
Z <- model.matrix(~z-1)
eta <- beta[1]+beta[2]*x+crossprod(t(Z),u)
p <- 1/(1+exp(-eta))

# Simulating the response variable
y <- rBI(nObs,m,p)

# Apply the model
model <- BImm(fixed.formula = y~x,random.formula = ~z,m=m)
model
```

BIreg — *Fit a binomial logistic regression model*

Description

`BIreg` function fits a binomial logistic regression model, i.e., it links the probability parameter of a binomial distribution with the given covariates by means of a logistic link function. There is the option to include a dispersion parameter in the binomial distribution, which will be estimated by the bias corrected method of moments.

Usage

```
BIreg(formula,m,data,disp=FALSE,maxiter=20)
```

Arguments

`formula`	an object of class `"formula"` (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`m`	number of trials in each binomial observation.

`data` an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula).

`disp` if TRUE a dispersion parameter will be estimated. Default FALSE.

`maxiter` the maximum number of iterations in IWLS method. Default 20.

Details

`BIreg` function performs a regression model linking by a logistic function the probability paramater of a binomial distribution with a linear predictor that consists of the given covariates. Following the exponential family theory, the binomial distribution with dispersion parameter has the following log-likelihood function:

$$l = [y * log(p/(1-p)) + m * log(1-p)]/phi + c(y, phi)$$

where $c()$ is a known function. If we any dispersion parameter is not considered the usual density function of the binomial distribution will be used,

$$l = y * log(p) + (m - y) * log((1 - p)).$$

As explained before we link the probablity parameter with the given covariates by

$$logit(p) = log(p/(1-p)) = x_i' * beta$$

where $beta$ are the regression coefficients and x_i is the ith row of a full rank design matrix X composed by the given covariables.

The estimation of the regression parameters $beta$ is done via maximum likelihood approach, where the iterative weighted least square (IWLS) method is applied.

If `disp` is TRUE, a dispersion parameter will be added in the binomial distribution and, consequently, the method will deal with the general definition of the log-likelihood formula, otherwise the usual and simpler one will be used. In case the dispersion parameter is included, the estimation will be done with a bias-corrected method of moments:

$$phi = Var[y]/[(m - q) * p * (1 - p)]$$

where q is the number of estimated regression paramters, and p is the estimated probability parameter.

The deviance of the model is defined by the ratio between the log-likelihood of the estimated model and saturated or null model. If the dispersion paramter is included the scaled deviance is obtained dividing the deviance by the dispersion parameter.

Value

`BIreg` returns an object of class "`BIreg`".

The function `summary` (i.e., `summary.BIreg`) can be used to obtain or print a summary of the results.

`coefficients` the estimated value of the regression coefficients.

`vcov` the variance and covariance matrix of the estimated regression coefficients.

`phi`	if `disp` TRUE, it returns the estimated value of the dispersion parameter. If `disp` FALSE, then the estimated value is 1. Default FALSE.
`fitted.values`	the fitted mean values of the model.
`residuals`	working residuals, i.e. the residuals in the final iteration of the IWLS method.
`deviance`	deviance of the model.
`df`	degrees of freedom of the model.
`null.deviance`	null-deviance, deviance for the null model. The null model will include only an intercept.
`df.null`	degrees of freedom for the null model.
`iter`	number of iterations in the IWLS method.
`conv`	if the algorithm has converged it returns "yes", otherwise "no".
`X`	model matrix.
`y`	dependent variable in the model.
`balanced`	if the response binomial variable is balanced it returns "yes", otherwise "no".
`m`	number of trials in each binomial observation.
`nObs`	number of observations.
`call`	the matched call.
`formula`	the formula supplied.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

Williams D. A. (1982): Extra-Binomial Variation in Logistic Linear Regression, *Journal of the Royal Statistical Society. Series C*, **31**, 144-148

See Also

Iterative weighted least squares method function `BIiwls` in R-package `HRQoL`.

Examples

```
set.seed(1234)
# We simulate a covariable and construct the outcome variable applying
# an inverse logit link function on it.

m <- 10
k <-100
```

```
covariate <- rnorm(k,2,0.5)

beta <- c(-6,4)
p <- 1/(1+exp(-(beta[1]+beta[2]*covariate)))

# without dispersion parameter
outcome <- rBI(k,m,p)
model <- BIreg(outcome~covariate,m,disp=FALSE)
model

# with dispersion parameter
phi <- 2
outcome.disp <- rBI(k,m,p,phi)
model.disp <- BIreg(outcome.disp~covariate,m,disp=TRUE)
model.disp
```

HRQoLplot *Spider plot of the dimensions of the Short Form-36 Health Survey*

Description

This function creates a spider plot with the 8 health realted quality of life dimensions provided by the Short Form-36 Health Survey.

Usage

```
HRQoLplot(data,legend=FALSE,title="Short Form-36 Health Survey",
  dimlabel.cex=NULL,legend.cex=1,linewidth=3,title.cex=1,lty=1)
```

Arguments

`data` a data frame with each column relative to the observations of each SF-36 dimension. The columns of the data frame must be introduced in the following order:

1. column -> Physical Functioning
2. column -> Role Physical
3. column -> Body Pain
4. column -> General Health
5. column -> Vitality
6. column -> Social Functioning
7. column -> Role Emotional
8. column -> Mental Health

`legend` logical parameter, if TRUE the legend with the name of the rows of the data will appear. Default FALSE.

`title` the title of the plot. Default "Short Form-36 Health Survey".

`dimlabel.cex`	font size magnification for the labels of the dimension in the plot. If NULL, the font size is fixed at text()'s default. Default NULL.
`legend.cex`	font size of legend text(). Default 1.
`linewidth`	the width of the lines of the plot. Default 3.
`title.cex`	the font size of the title. Default 1.
`lty`	the line type of the plot and the legend. Default 1.

Details

The Short Form-36 Health Survey is a commonly used technique to measure the Health Related Quality of Life (HRQoL) in chronich diseases. It was developed within the Medical Outcomes Study *(Ware et al. (1993))*. It measures generic HRQoL concepts and provides an objective way to measure HRQoL from the patients point of view by scoring standardized responses to standardized questions. The validity and reability of this instrument has been broadly tested *(Stansfeld et al. (1997))*. The SF-36 has 36 items, with different answer options. It was constructed to respresent eight health dimensions, which are *physical functioning* (PF), *role physical* (RP), *body pain* (BP), *general health* (GH), *vitality* (VT), *social functioning* (SF), *role emotional* (RE) and *mental health* (MH). Each item is assigned to a unique helath dimension. Each of the multi-item dimensions contains two to ten items. The first four dimensions are mainly physical, whereas the last four measure mental aspects of HRQoL. The resulting raw scores are tipically transformed to standardized scale scores from 0 to 100, where a higher score indicates a better health status.

Arostegui et al. (2013) proposed a recoding methodology for the Short Form-36 Health Survey (SF-36) dimensions in order to apply a beta-binomial distribution. The `HRQoLplot` function plots the SF-36 dimensions scores in a spider plot. Each axis of the plot refers to an expecific SF-36 dimension. Hence, the order of the dimensions in the data frame object of the function has been stablished as it has been explained in Arguments section. Each observation of the data frame, the value of each observation in all the dimensions, is drawn with a line of a different color in the plot. The plot shows the name of each dimension and the maximum number of scores each dimension can obtain in each axis of the plot.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

This function depends on the function `radarchart` of the package `fmsb` created by Minato Nakazawa.

References

Arostegui I., Nunez-Anton V. & Quintana J. M. (2013): On the recoding of continuous and bounded indexes to a binomial form: an application to quality-of-life scores, *Journal of Applied Statistics*, **40**, 563-583

See Also

As it is said in the author section, the function depends on the function `radarchart` of the package `fmsb`

Examples

```
set.seed(5)
# We insert the columns in the order that has been determined:
n <- c(20,4,9,20,20,8,3,13)
k=3
p=runif(8,0,1)
phi <- runif(8,1,3)
dat <- data.frame(
  PF=rBB(k,n[1],p[1],phi[1]),
  RP=rBB(k,n[2],p[2],phi[2]),
  BP=rBB(k,n[3],p[3],phi[3]),
  GH=rBB(k,n[4],p[4],phi[4]),
  VT=rBB(k,n[5],p[5],phi[5]),
  SF=rBB(k,n[6],p[6],phi[6]),
  RE=rBB(k,n[7],p[7],phi[7]),
  MH=rBB(k,n[8],p[8],phi[8]))

rownames(dat) <- c("ID1", "ID2", "ID3")
HRQoLplot(dat,TRUE)
```

---

print.BBest *Print a BBest class model.*

Description

`print.BBest` is the BBest specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

`x`	a BBest class model.
`...`	for extra arguments.

Value

Prints a BBest object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Arostegui I., Nunez-Anton V. & Quintana J. M. (2006): Analysis of short-form-36 (SF-36): The beta-binomial distribution approach, *Statistics in Medicine*, **26**, 1318-1342

See Also

BBest

Examples

```
set.seed(9)
# Simulate a binomial distribution
y <- rBB(100,10,0.5,2)

# Apply the model
model <- BBest(y,10)
print(model)   # or just model
```

---

print.BBmm — *Print a BBmm class model.*

Description

print.BBmm is the BBmm specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

x	a BBmm class model.
...	for extra arguments.

Value

Prints a BBmm object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Breslow N. E. & Calyton D. G. (1993): Approximate Inference in Generalized Linear Mixed Models, *Journal of the American Statistical Association*, **88**, 9-25

Lee Y. & Nelder J. A. (1996): Hierarchical generalized linear models, *Journal of the Royal Statistical Society. Series B*, **58**, 619-678

Najera-Zuloaga J., Lee D.-J. & Arostegui I. (2017): Comparison of beta-binomial regression model approaches to analyze health related quality of life data, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280217690413

See Also

BBmm

Examples

```
set.seed(14)

# Defining the parameters
k <- 100
m <- 10
phi <- 0.5
beta <- c(1.5,-1.1)
sigma <- 0.5

# Simulating the covariate and random effects
x <- runif(k,0,10)
X <- model.matrix(~x)
z <- as.factor(rBI(k,4,0.5,2))
Z <- model.matrix(~z-1)
u <- rnorm(5,0,sigma)


# The linear predictor and simulated response variable
eta <- beta[1]+beta[2]*x+crossprod(t(Z),u)
p <- 1/(1+exp(-eta))
y <- rBB(k,m,p,phi)
dat <- data.frame(cbind(y,x,z))
dat$z <- as.factor(dat$z)

# Apply the model
model <- BBmm(fixed.formula = y~x,random.formula = ~z,m=m,data=dat)
print(model) # or just model
```

`print.BBreg` *Print a BBreg class model.*

Description

`print.BBreg` is the BBreg specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

`x`	a BBreg class model.
`...`	for extra arguments.

Value

Prints a BBreg object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Forcina A. & Franconi L. (1988): Regression analysis with Beta-Binomial distribution, *Revista di Statistica Applicata*, **21**, 7-12

Najera-Zuloaga J., Lee D.-J. & Arostegui I. (2017): Comparison of beta-binomial regression model approaches to analyze health related quality of life data, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280217690413

See Also

BBreg

Examples

```
# We simulate a covariate, fix the paramters of the beta-binomial
# distribution and simulate a response variable.

# Then we apply the model, and try to get the same values.
set.seed(18)
k <- 1000
m <- 10
x <- rnorm(k,5,3)

beta <- c(-10,2)
p <- 1/(1+exp(-(beta[1]+beta[2]*x)))
```

```
phi <- 1.2

y <- rBB(k,m,p,phi)

model <- BBreg(y~x,m)
print(model) # or just model
```

---

print.BIest *Print a BIest class model.*

Description

`print.BIest` is the BIest specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

`x`	a BIest class model.
`...`	for extra arguments.

Value

Prints a BIest object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Pawitan Y. (2001): *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, Oxford University Press

See Also

`BIest`

Examples

```
set.seed(9)
# Simulate a binomial distribution
y <- rBI(100,10,0.5)

# Apply the model
model <- BIest(y,10)
print(model)   # or just model
```

---

print.BImm — *Print a BImm class model.*

Description

`print.BImm` is the BImm specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

`x`	a BImm class model.
`...`	for extra arguments.

Value

Prints a BImm object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Breslow N. E. & Calyton D. G. (1993): Approximate Inference in Generalized Linear Mixed Models, *Journal of the American Statistical Association*, **88**, 9-25

McCulloch C. E. & Searle S. R. (2001): Generalized, Linear, and Mixed Models, *Jhon Wiley & Sons*

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

See Also

BImm

Examples

```
set.seed(5)
# Fixing parameters for the simulation:
nObs <- 1000
m <- 10
beta <- c(1.5,-1.1)
sigma <- 0.8

# Simulating the covariate:
x <- runif(nObs,-5,5)

# Simulating the random effects:
z <- as.factor(rBI(nObs,5,0.5,2))
u <- rnorm(6,0,sigma)

# Getting the linear predictor and probability parameter.
X <- model.matrix(~x)
Z <- model.matrix(~z-1)
eta <- beta[1]+beta[2]*x+crossprod(t(Z),u)
p <- 1/(1+exp(-eta))

# Simulating the response variable
y <- rBI(nObs,m,p)

# Apply the model
model <- BImm(fixed.formula = y~x,random.formula = ~z,m=m)
print(model) # or just model
```

---

print.BIreg *Print a BIreg class model.*

Description

print.BIreg is the BIreg specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

x	a BIreg class model.
...	for extra arguments.

Value

Prints a BIreg object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

Williams D. A. (1982): Extra-Binomial Variation in Logistic Linear Regression, *Journal of the Royal Statistical Society. Series C*, **31**, 144-148

See Also

BIreg

Examples

```
set.seed(1234)
# We simulate a covariable and construct the outcome variable applying
# an inverse logit link function on it.

m <- 10
k <-100
covariate <- rnorm(k,2,0.5)

beta <- c(-6,4)
p <- 1/(1+exp(-(beta[1]+beta[2]*covariate)))

# without dispersion parameter
outcome <- rBI(k,m,p)
model <- BIreg(outcome~covariate,m,disp=FALSE)
print(model) # or just model
```

`print.summary.BBest` *Print a summary.BBest class model.*

Description

`print.summary.BBest` is the summary.BBest specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

`x`	a summary.BBest class model.
`...`	for extra arguments.

Value

Prints a summary.BBest object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Arostegui I., Nunez-Anton V. & Quintana J. M. (2006): Analysis of short-form-36 (SF-36): The beta-binomial distribution approach, *Statistics in Medicine*, **26**, 1318-1342

See Also

`BBest`,`summary.BBest`

Examples

```
set.seed(9)
# Simulate a binomial distribution
y <- rBB(100,10,0.5,2)

# Apply the model
model <- BBest(y,10)
sum.model <- summary(model)
print(sum.model) # or just sum.model
```

`print.summary.BBmm` *Print a summary.BBmm class model.*

Description

`print.summary.BBmm` is the summary.BBmm specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

`x`	a summary.BBmm class model.
`...`	for extra arguments.

Value

Prints a summary.BBmm object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Breslow N. E. & Calyton D. G. (1993): Approximate Inference in Generalized Linear Mixed Models, *Journal of the American Statistical Association*, **88**, 9-25

Lee Y. & Nelder J. A. (1996): Hierarchical generalized linear models, *Journal of the Royal Statistical Society. Series B*, **58**, 619-678

Najera-Zuloaga J., Lee D.-J. & Arostegui I. (2017): Comparison of beta-binomial regression model approaches to analyze health related quality of life data, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280217690413

See Also

`BBmm`, `summary.BBmm`

Examples

```
set.seed(14)

# Defining the parameters
k <- 100
m <- 10
phi <- 0.5
beta <- c(1.5,-1.1)
sigma <- 0.5

# Simulating the covariate and random effects
x <- runif(k,0,10)
X <- model.matrix(~x)
z <- as.factor(rBI(k,4,0.5,2))
Z <- model.matrix(~z-1)
u <- rnorm(5,0,sigma)


# The linear predictor and simulated response variable
eta <- beta[1]+beta[2]*x+crossprod(t(Z),u)
p <- 1/(1+exp(-eta))
y <- rBB(k,m,p,phi)
dat <- data.frame(cbind(y,x,z))
dat$z <- as.factor(dat$z)

# Apply the model
model <- BBmm(fixed.formula = y~x,random.formula = ~z,m=m,data=dat)
sum.model <- summary(model)
print(sum.model) # or just sum.model
```

---

print.summary.BBreg *Print a summary.BBreg class model.*

Description

print.summary.BBreg is the summary.BBreg specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

x	a summary.BBreg class model.
...	for extra arguments.

Value

Prints a summary.BBreg object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Forcina A. & Franconi L. (1988): Regression analysis with Beta-Binomial distribution, *Revista di Statistica Applicata*, **21**, 7-12

Najera-Zuloaga J., Lee D.-J. & Arostegui I. (2017): Comparison of beta-binomial regression model approaches to analyze health related quality of life data, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280217690413

See Also

BBreg,summary.BBreg

Examples

```
# We simulate a covariate, fix the paramters of the beta-binomial
# distribution and simulate a response variable.

# Then we apply the model, and try to get the same values.
set.seed(18)
k <- 1000
m <- 10
x <- rnorm(k,5,3)

beta <- c(-10,2)
p <- 1/(1+exp(-(beta[1]+beta[2]*x)))
phi <- 1.2

y <- rBB(k,m,p,phi)

model <- BBreg(y~x,m)
sum.model <- summary(model)
print(sum.model) # or just sum.model
```

`print.summary.BImm` *Print a summary.BImm class model.*

Description

`print.summary.BImm` is the summary.BImm specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

`x`	a summary.BImm class model.
`...`	for extra arguments.

Value

Prints a summary.BImm object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Breslow N. E. & Calyton D. G. (1993): Approximate Inference in Generalized Linear Mixed Models, *Journal of the American Statistical Association*, **88**, 9-25

McCulloch C. E. & Searle S. R. (2001): Generalized, Linear, and Mixed Models, *Jhon Wiley & Sons*

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

See Also

`BImm`, `summary.BImm`

Examples

```
set.seed(5)
# Fixing parameters for the simulation:
nObs <- 1000
m <- 10
beta <- c(1.5,-1.1)
sigma <- 0.8

# Simulating the covariate:
x <- runif(nObs,-5,5)

# Simulating the random effects:
z <- as.factor(rBI(nObs,5,0.5,2))
u <- rnorm(6,0,sigma)

# Getting the linear predictor and probability parameter.
X <- model.matrix(~x)
Z <- model.matrix(~z-1)
eta <- beta[1]+beta[2]*x+crossprod(t(Z),u)
p <- 1/(1+exp(-eta))

# Simulating the response variable
y <- rBI(nObs,m,p)

# Apply the model
model <- BImm(fixed.formula = y~x,random.formula = ~z,m=m)
sum.model <- summary(model)
print(sum.model) # or just sum.model
```

`print.summary.BIreg` *Print a summary.BIreg class model.*

Description

`print.summary.BIreg` is the summary.BIreg specific method fot the generic function print which prints objects returned by modelling functions.

Usage

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

Arguments

`x`	a summary.BIreg class model.
`...`	for extra arguments.

Value

Prints a summary.BIreg object.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

Williams D. A. (1982): Extra-Binomial Variation in Logistic Linear Regression, *Journal of the Royal Statistical Society. Series C*, **31**, 144-148

See Also

BIreg, summary.BIreg

Examples

```
set.seed(1234)
# We simulate a covariable and construct the outcome variable applying
# an inverse logit link function on it.

m <- 10
k <-100
covariate <- rnorm(k,2,0.5)

beta <- c(-6,4)
p <- 1/(1+exp(-(beta[1]+beta[2]*covariate)))
outcome <- rBI(k,m,p)

model <- BIreg(outcome~covariate,m,disp=FALSE)
sum.model <- summary(model)
print(sum.model) # or just sum.model
```

SF36rec *Short Form-36 Health Survey recode*

Description

The SF36rec function recodes to a binomial form the 0-100 original standardized scores of the dimensions provided by the Short Form-36 Health Survey (SF-36) based on *Arostegui et al. (2013).*

Usage

```
SF36rec(x,k)
```

Arguments

`x` the 0-100 scale standardized dimension of the SF-36. It must be numeric and bounded between 0 and 100.

`k` an integer from 1 to 8 that defines which SF-36 dimension belongs `x`. These are the dimensions depending on the `k` value:

`k=1` -> Physical functioning

`k=2` -> Role physical

`k=3` -> Body pain

`k=4` -> General health

`k=5` -> Vitality

`k=6` -> Social functioning

`k=7` -> Role emotional

`k=8` -> Mental health

Details

The Short Form-36 Health Survey is a commonly used technique to measure the Health Related Quality of Life (HRQoL) in chronich diseases. It was developed within the Medical Outcomes Study *(Ware et al. (1993))*. It measures generic HRQoL concepts and provides an objective way to measure HRQoL from the patients point of view by scoring standardized responses to standardized questions. The validity and reability of this instrument has been broadly tested *(Stansfeld et al. (1997))*. The SF-36 has 36 items, with different answer options. It was constructed to respresent eight health dimensions, which are *physical functioning* (PF), *role physical* (RP), *body pain* (BP), *general health* (GH), *vitality* (VT), *social functioning* (SF), *role emotional* (RE) and *mental health* (MH). Each item is assigned to a unique helath dimension. Each of the multi-item dimensions contains two to ten items. The first four dimensions are mainly physical, whereas the last four measure mental aspects of HRQoL. The resulting raw scores are tipically transformed to standardized scale scores from 0 to 100, where a higher score indicates a better health status.

Arostegui et al. (2013) proposed a recoding methodology of the SF-36 standardized scores to a binomial form in order to apply the beta-binomial distribution. The method was mainly based on the possible number of values each dimension can obtain, which comes from the number of items related with the construction of each dimension.

The `SF36rec` function performes the cited recoding methodology to the specified SF-36 dimension. It has two inputs. The first one is the dimension that will be recoded, and the second one identifies which SF-36 dimension is.

Value

The score values of the recoded dimension.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Arostegui I., Nunez-Anton V. & Quintana J. M. (2013): On the recoding of continuous and bounded indexes to a binomial form: an application to quality-of-life scores, *Journal of Applied Statistics*, **40**, 563-583

Ware J. E., Snow K. K., Kosinski M. A. & Gandek B. (1993): *SF-36 Health Survey, Manual and Interpretation Guides*. The Health Institute, New England Medical Center.

Stansfeld S. A., Roberts R. & Foot S. P. (1997): Assessing the validity of the SF-36 general health survey. Quality of Life Research, **6**, 217-224.

Examples

```
set.seed(2)
# We simulate a variable bounded between 0 and 100.
BodyPain <- runif(1000,0,100)

# We specify that the simulated dimension corresponds
# with body pain dimension.
k <- 3

# We perform the recoding.
BodyPain.rec <- SF36rec(BodyPain,k)
```

---

summary.BBest — *Summarizes a BBest class model.*

Description

`summary.BBest` si the BBest specific method for the generic function `summary` which summarizes objects returned by modelling functions.

Usage

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

Arguments

`object`	a BBest class model.
`...`	for extra arguments.

Details

`summary.BBest` summarizes all the relevant information about the estimation of the parameters in a BBest class model.

Value

summary.BBest returns an object of class "summary.BBest".

coefficients — a table with the estimated parameters is in the BBest class model.

p.coefficients — a summarized table of the estimation of the probability parameter of the beta-binomial distribution. The table contents the estimation of the probability parameter and the standard errors of the estimations.

psi.coefficients — a summarized table of the estimation of the logarithm of the dispersion parameter of the beta-binomial distribution. The table contents the estimation of the logarithm of the dispersion parameter and the standard errors of the estimations.

m — the maximum score number in each beta-binomial observation.

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Arostegui I., Nunez-Anton V. & Quintana J. M. (2006): Analysis of short-form-36 (SF-36): The beta-binomial distribution approach, *Statistics in Medicine*, **26**, 1318-1342

See Also

BBest

Examples

```
set.seed(9)
# Simulate a binomial distribution
y <- rBB(100,10,0.5,2)

# Apply the model
model <- BBest(y,10)
sum.model <- summary(model)
```

---

summary.BBmm — *Summarizes a BBmm class model.*

Description

summary.BBmm si the BBmm specific method for the generic function summary which summarizes objects returned by modelling functions.

Usage

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

Arguments

`object`	a BBmm class model.
`...`	for extra arguments.

Details

`summary.BBmm` summarizes all the relevant information about the estimation of the parameters in a BBmm class model.

The function performs statistical significance hypothesis about the estimated fixed parameters based on the normal distribution of the estimates. It also performs a goodness of fit test based on the difference between the calculated deviance of the model and the null deviance or deviance of the null model, which it is suppose to follow a Chi-square distribution with degrees of freedom equal to the difference in degrees of freedom of the models.

Value

`summary.BBmm` returns an object of class "`summary.BBmm`".

`fixed.coefficients`
: a table with all the relevant information about the significance of the fixed effects estimates in the model. It includes the estimates, the standard errors of the estimates, the test-statistics and the p-values.

`sigma.table`
: a table which inlcudes the estimates and the standard errors of the estimates of the random effects variance parameters.

`psi.table`
: a table which includes the estimate and the standard errors of the estimate of the logarithm of the dispersion parameter of the conditional beta-binomial distribution.

`random.coef`
: predicted random effects of the regression.

`iter`
: number of iterations in the estimation method.

`nObs`
: number of observations in the data.

`nRand`
: number of random effects.

`nComp`
: number of random components.

`nRandComp`
: number of random effects in each random component of the model.

`namesRand`
: names of the random components.

`deviance`
: deviance of the model.

`df`
: degrees of freedom of the model.

`null.deviance`
: null-deviance, deviance of the null model. The null model will only include an intercept as the estimation of the probability parameter of the conditinal beta-binomial distribution.

null.df	degrees of freedom of the null model.
Goodness.of.fit	p-value of the goodness of fit test.
balanced	if the conditional beta-binomial response variable is balanced it returns "yes", otherwise "no".
m	maximum score number in each beta-binomial observation.
conv	convergence of the methodology. If the algorithm has converged it returns "yes", otherwise "no".

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Breslow N. E. & Calyton D. G. (1993): Approximate Inference in Generalized Linear Mixed Models, *Journal of the American Statistical Association*, **88**, 9-25

Lee Y. & Nelder J. A. (1996): Hierarchical generalized linear models, *Journal of the Royal Statistical Society. Series B*, **58**, 619-678

Najera-Zuloaga J., Lee D.-J. & Arostegui I. (2017): Comparison of beta-binomial regression model approaches to analyze health related quality of life data, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280217690413

See Also

The multiroot and uniroot functions of the R-package rootSolve for the general Newton-Raphson algorithm.

BBmm.

Examples

```
set.seed(14)

# Defining the parameters
k <- 100
m <- 10
phi <- 0.5
beta <- c(1.5,-1.1)
sigma <- 0.5

# Simulating the covariate and random effects
x <- runif(k,0,10)
X <- model.matrix(~x)
z <- as.factor(rBI(k,4,0.5,2))
Z <- model.matrix(~z-1)
u <- rnorm(5,0,sigma)
```

```
# The linear predictor and simulated response variable
eta <- beta[1]+beta[2]*x+crossprod(t(Z),u)
p <- 1/(1+exp(-eta))
y <- rBB(k,m,p,phi)
dat <- data.frame(cbind(y,x,z))
dat$z <- as.factor(dat$z)

# Apply the model
model <- BBmm(fixed.formula = y~x,random.formula = ~z,m=m,data=dat)
sum.model <- summary(model)
```

`summary.BBreg` *Summarizes a BBreg class model.*

Description

`summary.BBreg` si the BBreg specific method for the generic function `summary` which summarizes objects returned by modelling functions.

Usage

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

Arguments

`object`	a BBreg class model.
`...`	for extra arguments.

Details

`summary.BBreg` summarizes all the relevant information about the estimation of the parameters in a BBreg class model.

The function performs statistical significance hypothesis about the estimated regression parameters based on the normal distribution of the estimates. It also performs a goodness of fit test based on the difference between the calculated deviance of the model and the null deviance or deviance of the null model, which it is suppose to follow a Chi-square distribution with degrees of freedom equal to the difference in degrees of freedom of the models.

Value

`summary.BBreg` returns an object of class "`summary.BBreg`".

`coefficients`	a table with all the relevant information about the significance of the regression coefficients of the model. It includes the estimations, the standard errors of the estimations, the test-statistics and the p-values.

psi.table	a table which includes the estimation and standard errors of the logarithm of the dispersion parameter of the conditional beta-binomial distribution.
deviance	deviance of the model.
df	degrees of freedom of the model.
null.deviance	null-deviance, deviance of the null model.
null.df	degrees of freedom of the null model.
Goodness.of.fit	p-value of the goodness of fit test.
iter	number of iterations in the estimation method.
X	the model matrix.
y	the dependent variable in the model.
balanced	if the response variable is balanced it returns "yes", otherwise "no".
m	number of trials in each binomial observation.
nObs	number of observations.
m	number of trials in each observation.
balanced	if the response beta-binomial variable is balanced it returns "yes", otherwise "no".
conv	convergence of the methodology. If the algorithm has converged it returns "yes", otherwise "no".

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Forcina A. & Franconi L. (1988): Regression analysis with Beta-Binomial distribution, *Revista di Statistica Applicata*, **21**, 7-12

Najera-Zuloaga J., Lee D.-J. & Arostegui I. (2017): Comparison of beta-binomial regression model approaches to analyze health related quality of life data, *Statistical Methods in Medical Research*, DOI: 10.1177/0962280217690413

Examples

```
# We simulate a covariate, fix the paramters of the beta-binomial
# distribution and simulate a response variable.

# Then we apply the model, and try to get the same values.
set.seed(18)
k <- 1000
m <- 10
x <- rnorm(k,5,3)
```

```
beta <- c(-10,2)
p <- 1/(1+exp(-(beta[1]+beta[2]*x)))
phi <- 1.2

y <- rBB(k,m,p,phi)

model <- BBreg(y~x,m)
sum.model <- summary(model)
```

---

summary.BImm *Summarizes a BImm class model.*

Description

`summary.BImm` si the BImm specific method for the generic function `summary` which summarizes objects returned by modelling functions.

Usage

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

Arguments

`object`	a BImm class model.
`...`	for extra arguments.

Details

`summary.BImm` summarizes all the relevant information about the estimation of the parameters in a BImm class model.

The function performs statistical significance hypothesis about the estimated fixed parameters based on the normal distribution of the estimates.

Value

`summary.BImm` returns an object of class "`summary.BImm`".

`fixed.coefficients`
: a table with all the relevant information about the significance of the fixed effects of the model. It includes the estimations, the standard errors of the estimations, the test-statistics and the p-values.

`random.coef`
: predicted random effects of the regression.

`sigma.table`
: a table which inlcudes the estimation and standard errors of the parameters which the variance-covariance matrix of the random effects consists of.

`fitted.values`
: the fitted mean values of the probability parameter of the conditional beta-binomial distribution.

`residuals`	residuals of the model.
`deviance`	deviance of the model.
`df`	degrees of freedom of the model.
`nRand`	number of random effects.
`nComp`	number of random components.
`nRandComp`	number of random components in each random effect of the model.
`namesRand`	names of the random components.
`iter`	number of iterations in the estimation method.
`nObs`	number of observations in the data.
`y`	dependent response variable in the model.
`X`	model matrix of the fixed effects.
`Z`	model matrix of the random effects.
`balanced`	if the conditional binomial response variable is balanced it returns "yes", otherwise "no".
`m`	number of trials in each binomial observation.
`conv`	convergence of the methodology. If the algorithm has converged it returns "yes", otherwise "no".

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Breslow N. E. & Calyton D. G. (1993): Approximate Inference in Generalized Linear Mixed Models, *Journal of the American Statistical Association*, **88**, 9-25

McCulloch C. E. & Searle S. R. (2001): Generalized, Linear, and Mixed Models, *Jhon Wiley & Sons*

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

See Also

BImm

Examples

```
set.seed(5)
# Fixing parameters for the simulation:
nObs <- 1000
m <- 10
beta <- c(1.5,-1.1)
```

```
sigma <- 0.8

# Simulating the covariate:
x <- runif(nObs,-5,5)

# Simulating the random effects:
z <- as.factor(rBI(nObs,5,0.5,2))
u <- rnorm(6,0,sigma)

# Getting the linear predictor and probability parameter.
X <- model.matrix(~x)
Z <- model.matrix(~z-1)
eta <- beta[1]+beta[2]*x+crossprod(t(Z),u)
p <- 1/(1+exp(-eta))

# Simulating the response variable
y <- rBI(nObs,m,p)

# Apply the model
model <- BImm(fixed.formula = y~x,random.formula = ~z,m=m)
sum.model <- summary(model)
```

`summary.BIreg` *Summarizes a BIreg class model.*

Description

`summary.BIreg` si the BIreg specific method for the generic function `summary` which summarizes objects returned by modelling functions.

Usage

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

Arguments

`object`	a BIreg class model.
`...`	for extra arguments.

Details

`summary.BIreg` summarizes all the relevant information about the estimation of the parameters in a BIreg class model.

The function performs statistical significance hypothesis about the estimated regression parameters based on the normal distribution of the estimates. It also performs a goodness of fit test based on the difference between the calculated deviance of the model and the null deviance or deviance of the null model, which it is suppose to follow a Chi-square distribution with degrees of freedom equal to the difference in degrees of freedom of the models.

Value

`summary.BIreg` returns an object of class "`summary.BIreg`".

`coefficients`	a table with all the relevant information about the significance of the regression coefficients of the model. It includes the estimations, the standard errors of the estimations, the test-statistics and the p-values.
`phi`	the estimated value of the dispersion parameter. If `disp` FALSE, then the estimated value is 1.
`deviance`	the deviance of the model.
`df`	the degrees of freedom of the model.
`null.deviance`	the deviance for the null model. The null model will include only an intercept.
`df.null`	the degrees of freedom for the null model.
`Goodness.of.fit`	p-value of the goodness of fit test.
`iter`	number of iterations in the IWLS method.
`conv`	convergence of the methodology. If the algorithm has converged it returns "yes", otherwise "no".
`X`	the model matrix.
`y`	the dependent variable in the model.
`balanced`	if the response variable is balanced it returns "yes", otherwise "no".
`m`	the number of trials in each observation.
`nObs`	number of observations.
`balanced`	if the response binomial variable is balanced it returns "yes", otherwise "no".

Author(s)

J. Najera-Zuloaga

D.-J. Lee

I. Arostegui

References

Pawitan Y. (2001): In All Likelihood: Statistical Modelling and Inference Using Likelihood, *Oxford University Press*

Williams D. A. (1982): Extra-Binomial Variation in Logistic Linear Regression, *Journal of the Royal Statistical Society. Series C*, **31**, 144-148

See Also

`BIreg`, `BIiwls`

Examples

```
set.seed(1234)
# We simulate a covariable and construct the outcome variable applying
# an inverse logit link function on it.

m <- 10
k <-100
covariate <- rnorm(k,2,0.5)
beta <- c(-6,4)
p <- 1/(1+exp(-(beta[1]+beta[2]*covariate)))
outcome <- rBI(k,m,p)

model <- BIreg(outcome~covariate,m,disp=FALSE)
sum.model <- summary(model)
```

Index