Package 'HRQoL'

February 22, 2016

Title Health Related Quality of Life Analysis

Version 1.0

Author Josu Najera <jnajera@bcamath.org>, Dae-Jin Lee <dlee@bcamath.org>

Maintainer Josu Najera <jnajera@bcamath.org>

Description Statistical analysis of Health Related Quality of Life data provided by the Short-Form-36 Health Survey

License What license is it under?

Imports fmsb, car, RColorBrewer, matrixcalc, rootSolve

NeedsCompilation no

R topics documented:

`HRQoL-package` *Health Related Quality of Life Analysis*

Description

The main aim of this package is to offer tools for dealing with the Health Related Quality of Life (HRQoL) analysis- This package provides functions based on different regression models for the analysis of outcome from HRQoL survey data.

The tools for measuring the HRQoL have been developed in questionaries. One of the most commonly used is the Short-Form-36 Health Survey (SF-36), which we have based on in this package.

The SF-36 provides information about the HRQoL divided in eight domains, each one related with an aspect of the HRQoL. This domains are bounded between 0 and 100 and can only take some values. The most natural way for the analysis of these domains is considering them distributed as a binomial random variable and recoding them from 0 to n. This package provides a function to perform the ideal recodification of the domains based on *Arostegui et al. (2013)*, `SF36rec`. It also give the choice, through `HRQoLplot`, to plot the HRQoL of the subjects in an radar/spider plot.

Several instruments for measuring HRQoL have been developed in form of questionnaires, some of them in a generic way and other for specific diseases. The SF-36 has 36 items that are reduced to 8 health dimensions, it was developed and validated by Ware et al. (Ware et al., 1993) and it has been translated and validated into many languages, including Spanish (Alonso et al., 1995). The health domains provided by the SF-36 are bounded between 0 and 100 and can only take some values. The most natural way of analyzing these domains is considering them as binomial distributed and recoding them from 0 to n. This package provides a function, `SF36rec`, to perform the ideal recodification of the domains based on Arostegui et al. (2013) proposal.

The recoded health domains have a binomial distribution, however, the relationship between the mean and variance the binomial distribution assumes is not usually met. Consequently, if we want to analyze the given dimensions as response variables we have to use some modelization approaches that will deal with over-dispersion problems. This packages provides different regression and estimation methodologies based on different approaches to modelize the over-dispersion:

1. Binomial distribution with dispersion parameter approach, which inserts a parameter, called dispersion parameter, in the variance of the response variable. That way the relationship between the expectation and variance of the response variable is more flexible and it takes into account the over variability that exists in the model.
2. Beta-binomial modeliztion approach, where conditioned on some beta distributed random variables, θ, the response variable in binomial with probability parameter θ. This distribution softens the mean and variance restriction, reaching different distribution shapes.

   Arostegui et al. (2006) showed that the beta-binomial distribution is adequate to HRQoL data. Moreover, the beta binomial regression (BRR) approach has been proposed in the literature, not only to detect significant predictors of HRQoL when SF-36 is used, but also to analyze and interpret the effect of several explanatory variables on HRQoL (Arostegui et al., 2010). Comparison of the BBR approach with other commonly used modelling approaches in the same context showed that the beta-binomial distribution was a good option to account for over-dispersion in HRQoL analysis and it offered convenient interpretation of the resutls (Arostegui et al., 2013).
3. Generalized Linear Mixed Model (GLMM) approach, which are the extension of GLMs (Generalized Linear Models) to the inclusion of normal distributed random effects. This package is focus on the analysis of HRQoL domains, so the GLMM framework is focus on the binomial distribution of the response variable. This methodology is a very widely used approach to analyze binomial repsonses in different framework, even in HRQoL.

Finally, apart form the previously defined modeling approaches, the package provides also some especific plots to draw HRQoL in different populations, where each dimension of the SF-36 is ploted in a scatter plot considering different characteristics of the given population.

References

Alonso, J., Prieto, L., and Ant\'on, J. M. (1995). La versi\'on espa\~nola del sf-36 health survey (cuestionario de salud sf–36): Un instrumento para la medida de los resultados cl\'inicos. *Medicina cl\'inica*, (104):771–776.

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*, 40:3, 563-583

Arostegui I., Nunez-Antón V. & Quintana J. M. (2010): Statistical approches to analyse patient-reported outcome as response variables: An application to health-related quality of life, *Statistical Methods in Medical Research*, 21, 189-214

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

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

Fahrmeir L. & Tutz G. (2001): Multivariate Modelling Based on Generalized Linear Mixed Models, *Springer Series in Statistics*

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

Goldsmith, S. B. (1972). The status of health status indicators. *Health Services Reports*, (87):212–220.

Jiang J. (2007): Linear and Generalized Linear Mixed Models and Their Applications, *Springer Science + Business Media*

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*

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

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

Examples

```
set.seed(123)
#  Number of observations
k <- 100
#  Maximun number of score in the binomial trials:
n <- 10
#  Probability:
p <- 0.5
#  Dispersion parameter:
phi <- 2
#  We simulate a overdispersed random variable following the beta-binomial distribution:
y <- rBB(k,n,p,phi)

#  We calculate the mle of the parameters using the BIest function,
#   binomial estimation with overdispersion:
est <- BIest(y,n,disp=TRUE)
est
est.p <- est[1]
est.phi <- est[5]

# If we plot it:
hist(y,col="grey",breaks=seq(-0.5,10.5,1),probability = TRUE)
lines(c(0:n),dBI(n,est.p,est.phi),col="red",lwd=4)
```

```
# Now we are going to calculate the mle of the parameters using
# the BBest function, beta-binomial:
out <- BBest(y,n)$coef
out
out.p <- out[2]
out.phi <- out[3]
# If we plot it:
hist(y,col="grey",breaks=seq(-0.5,10.5,1),probability = TRUE)
lines(c(0:n),dBB(0:n,n,out.p,out.phi),col="red",lwd=4)

#  Perform a regression:
x <- rnorm(100,2,2)
# Binomial with overdispersion distribution:
BIreg(y~x,10,disp=TRUE)
# Beta-binomial regression:
BBreg(y~x,n)
```

BB *The Beta-Binomial Distribution*

Description

Density and random generation for the beta-binomial distribution.

Usage

```
dBB(y,n,p,phi)
rBB(k,n,p,phi)
```

Arguments

n	the maximum score of the beta-binomial trials.
k	the number of simulations.
y	the number of successes in n beta-binomial trials.
p	the probabilily parameter of the beta-binomial distribution.
phi	the dispersion parameter of the beta-binomial distribution.

Details

The beta-binomial distribution is defined as a mixture between a binomial distribution and a beta distribution. It assumes that conditioned on some random components θ, with beta distribution with parameters p/ϕ and $p/(1-\phi)$, the response variable y follows a binomial distribution with probability parameter θ,

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

The expectation, variance and density function of this distribution can be explicitly calculated:

$$E[y] = np,$$

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

Consequently, p is called the probability parameter and ϕ is called the dispersion parameter of the beta-binomial distribution. Hence, the response variable y follows a beta-binomial distribution of parameters n, p and ϕ,

$$y \sim BB(n,p,\phi).$$

Value

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

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

Author(s)

Josu Najera

Dae-Jin Lee

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
n <- 10     #maximum number of scores in the binomial trials
p <- 0.4    #probability parameter of the beta-binomial distribution
phi <- 1.8  #dispersion parameter of the beta-binomial distribution

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

# Show the histogram of the generated variable,
#with dBB() function we fit a beta-binomial distribution:
hist(bb,col="grey",breaks=seq(-0.5,n+0.5,1),probability=TRUE,main="Histogram",xlab="beta-binomial random va
lines(c(0:n),dBB(0:n,n,p,phi),col="red",lwd=4)
```

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

Description

This function performs the estimation of the beta-binomial distribution parameters, p and ϕ, by the method of moments. The variance of the estimations is performed using maximum likelihood by the analytically determined beta-binomial density function.

Usage

```
BBest(y,n)
```

Arguments

y	the outcome variable that is supposed to follow a beta-binomial distribution.
n	the maximum score of the beta-binomial trials.

Details

The method of moments assumes that the response variable y follows a beta-binomial distribution and using the moments of first and second order, i.e., the mean and expectation of the beta-binomial distribution gets the estimation of the parameters.

$$E[y] = np$$

$$Var[y] = np(1-p)[1+(n-1)\phi/(1+\phi)]$$

The dispersion parameter, ϕ, of the beta-binomial distribution can only be positive, as the beta-binomial distribution models the overdispersion. So if the estimated ϕ parameter by the method of moments is negative, meaning that there exists underdispersion, the beta-binomial model will not be useful and the function will print an error message, sugesting the use of the binomial distribution.

The variance of the estimations is calculated by maximum likelihood by the second derivative of the density function of the beta-binomial distribution, using the Fisher score matrix, and remplacing the estmiated values of the parameters in it.

Value

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

n	the maximum score of the Binomial trials.
p	the estimated probability parameter.
phi	the estimated dispersion parameter.
vcov	the variance and covariance matrix of the parameters of the beta-binomial distribution

Author(s)

Josu Najera

Dae-Jin Lee

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 beta-binomial distributed observations.
n <- 10     # maximum score of the Binomial trials
k <- 1000   # number of simulated observations
p <- 0.7    # probability parameter of the beta-binomial distribution
phi <- 1.6  # dispersion parameter of the beta-binomial distribution

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

```
# Estimation
est <- BBest(y,n)
print(est)

# Summary of the results
summary(est)
```

BBreg *Fitting Beta-Binomial Regression Models*

Description

BBreg is used to fit beta-binomial regression models, where the estimation is performed using maximum likelihood.

Usage

```
BBreg(formula,n,data)
```

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.
n	the maximum score of the binomial trials.
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).

Details

The probability parameter of the beta-binomial distribution p is connected to a vector of regression parameters by means of a logit link function model. The relationship between the covariables $X_1, \ldots, X_t$ and the probablity parameter of each observation p_i, $i = 1, \ldots, m$ where m is the number of observations, is defined as

$$p_i = exp(x_i'\beta)/(1 + exp(x_i'\beta)),$$

where β is a $(t+1) \times 1$ vector of regression parameter, and x_i is the ith row of a full design matrix X composed by the covariables.

The marginal density of the beta-binomial distribution can be explicity calculated, so if we replace the previous equation on the probability parameter of the density function we can perform estimations by maximum likelihood in the distribution parameters. Forcina and Franconi (1988) presented an iterative weighted least square method to perfom estimation on the both, regression parameters and, the dispersion parameter of the beta-binomial distribution.

If the estimation of the dispersion parameter ϕ in the iterative weigthes least squares method is lower than $0.05/[(n-1) - 0.05)]$, where n is the number of trials, the function will use the usual logistic regression based on a binomial distribution, as a lower estimation of the dispersion parameter than the given value means that there is no dispersion problem in the model.

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.

The generic accessor functions coefficients, fitted.values and residuals can be used to extract various useful features of the value returned by BBreg.

coefficients the estimated value of the regression coefficients.

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

phi.coefficient the logarithm of the estimated value of the dispersion parameter of a beta-binomial distribution model.

phi.var the variance of the logarithm of the estimated dispersion parameter.

fitted.values the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.

residuals the working residuals, that is the residuals in the final iteration of the IWLS fit.

df degrees of freedom of the model.

deviance the deviance of the model, i.e., minus twice the maximized log-likelihood.

iter number of iterations in the iterative weighted least squares method.

X the model matrix.

y the dependent variable in the model.

n the maximum score of the Binomial trials.

noObs number of observations in the data frame.

call the matched call.

formula the formula supplied.

Author(s)

Josu Najera

Dae-Jin Lee

References

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

Examples

```
# We will generate a beta-binomial response variable fixing some values of
# the regression parameters and simualting a random effect. Then we are going
# to proof that we reach the same values.

# We generate the outcome variable fromm a simulated covariable.
set.seed(11)
k <- 100
n <- 10
x <- rnorm(k,5,3)

# We calculate the probability parameter as in the proposed methodology.
```

```
p <- 1/(1+exp(-(2*x-10)))
phi <- 1.2

# We simualte a beta-binomial variable for those parameters.
y <- rBB(k,n,p,phi)

# We perform the beta binomial regression, reaching very similar values
BBreg(y~x,n)
```

BI *The Binomial distribution with Dispersion Parameter*

Description

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

Usage

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

Arguments

k	number of simulations.
n	the maximum score of the binomial trials.
p	the probability of scoring a success in each binomial trial.
phi	the dispersion parameter of the binomial distribution. If `phi=1`, then the simple binomial model will be performed.

Details

The inclusion of a dispersion parameter in the variance equation of the binomial distribution softens the relationship that is expected between the mean and variance in binomial models,

$$E[y] = np, \quad Var[y] = \phi np(1-p).$$

The density function of this binomial model is calculated considering as an expontial family, where the density function has the following form

$$f(y) = \frac{ylog(p/(1-p)) + nlog(1-p)}{\phi} + c(y, \phi),$$

where $c(y, \phi)$ is a function that is approximated by the deviance of the model.

Value

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

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

Author(s)

Josu Najera

Dae-Jin Lee

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
n <- 10
p <- 0.765
phi <- 4.35

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

#density function
d <- dBI(n,p,phi)
d
```

BIest *Estimation of the parameters of a binomial distribution*

Description

This function calculates the maximum likelihood estimation of the probability parameter in a binomial distribution. It also returns the standard errors of the estimator, and the upper and lower bounds of the interval of confidence. There is the option to include a dispersion parameter in the model, which is estimated by the method of moments.

Usage

```
BIest(y,n,disp=FALSE)
```

Arguments

`y`	the variable under the binomial distribution assumption we want to model.
`n`	the maximum score of the binomial trials.
`disp`	if TRUE the dispersion parameter will be added to the model. By default it uses the FALSE option.

Details

The estimation of the probability parameter is calculated by maximum likelihood considering the binomial distribution as a general exponential family distribution. The log-likelihood is derived and equalized to zero, reaching the solution that maximizes the log-likelihood. It also returns the standard error of the estimator, which is obtained by the Fisher information formula, i.e., the second derivate of the log-likelihood inserting the mle.

On the other hand, once the mle of the probability parameter has been found out, the dispersion parameter is estimate by the method of moments by the following formula,

$$Var[Y] = \phi np(1-p)$$

We replace the estimated value of the probability parameter and the calculated variance of the variable in the formula reaching the estimated value of the dispersion parameter.

Value

`p`	the maximum likelihood estimation of the probability parameter.
`se`	the standard errors of the mle.
`low.ic`	the lower bound of the intervale of confidence of the mle.
`up.ic`	the upper bound of the intervale of confidence of the mle.
`phi`	if the disp option is TRUE, it returns the estimated value of the dispersion parameter. This estimation is realized with the method of moments, and influences in the standard error of the mle.

Author(s)

Josu Najera Zuloaga

Dae-Jin Lee

References

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

Examples

```
set.seed(9)
# We simulated the binomial data with some parameters and then
# we are going to try to reach the same estimations.
n <- 10            # the maximum score of the Binomial trials
k <- 100           # number of simulations
p <- 0.654         # probability parameter
y <- rbinom(k,n,p)  # simulations

# without overdispersion
BIest(y,n) #no overdispersion by default

# with overdispersion
BIest(y,n,TRUE)
```

BIMreg *Generalized Linear Mixed Models (GLMMs) for binomial data*

Description

This model provides the inclusion of random effects in the linear predictor of a logistic regression. The random components are assumed with a normal distribution of zero expectation. This type of model can be used to modelize binomial data over time or binomial hierarchical models.

Usage

```
BIMreg(formula,n,random,data)
```

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.
`n`	the maximum score of the binomial trials.
`random`	a vector containing the names of the random components in the data frame, the names must be between quotes.
`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).

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 distribution, allowing the inclusion of random components in the linear predictor,

$$logit(p) = X\beta + Zu,$$

where p is the probability parameter, X a full rank matrix composed by the covariables, β the regression coefficients, Z the design matrix for the random effects ang u are the random components. These random effects are independent and has a normal distribution with the same variance and mean 0.

The model estimates the fixed effects and the variance of the random effects, and predicts those random effects for each observation.

The estimation of the parameters is done by likelihood approximation, via iteartive weighted least squares method. The estimation is performed in two steps: (i) fixed and random parameters are calculated for some given random component variances (ii) variances of the random effects are calculated for some given regression and random coefficients. The estimation approach iterates between (i) and (ii) until convergence is obtained.

Value

`BImmreg` returns an object of class `"BIMreg"`.

The function `summary` (i.e., `summary.BIMreg`) can be used to obtain or print a summary of the results, `coef` can be used to print the regression coefficients of the model, `fitted` (i.e., `fitted.BIMreg`) to print the fitted values and `residuals` (i.e., `residuals.BIMreg`) to get the residuals.

`coefficients` the estimated value of the fixed coefficients in the regression.

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

`random.coef` the predicted random effects of the regression. when more than one random effect is included the prediction of all the random coefficients is given, following the order defined in parameter `random`.

`random.var` the variance of each random effect.

`deviance` the deviance of the model based on the approximated log-likelihood.

`fitted.values` the fitted mean values of the probability parameter of the conditioned binomial distribution, obtained by transforming the linear predictor by the inverse of the link function.

`residuals` the residuals of the model.

`random.se` the standard error of each random effect, the esquare of the variance of the random effects.

`working` the final working vector of the iterative weighted least square method.

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

`nObs` the number of observations in the data.

`nrand` the number of random components in the model.

`cluster` the number of levels in each random component.

`y` the dependent variable in the model.

`X` the model matrix of the fixed effects.

`Z` the model matrix of the random effects.

`D` the variance and covariance matrix of the random effects.

`n` the maximum score of the conditioned binomial trials.

`W` the final weights matrix in the iterative weighted least square method.

`call` the matched call.

`formula` the formula supplied.

Author(s)

Josu Najera

Dae-Jin Lee

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)
# Creating the dependent variable with a simulated covariable:
nObs <- 500
x <- rnorm(nObs,4,1)
id1 <- c(kronecker(seq(1,5),rep(1,100)))
id2 <- c(kronecker(seq(1,10),rep(1,50)))


p <- 1/(1+exp(-(5-5*x+kronecker(rnorm(5,0,0.5),rep(1,100))+kronecker(rnorm(10,0,1.2),rep(1,50)))))
y <- rbinom(nObs,10,p)
dat <- data.frame(cbind(y,x,id1,id2))
dat$id1 <- as.factor(dat$id1)
dat$id2 <- as.factor(dat$id2)

#Estimating the mixed model for one random component.
mm1 <- BIMreg(y~x,10,c("id1"),dat)
mm1
summary(mm1)

#Estimating the mixed model for two random components.
mm2 <- BIMreg(y~x,10,c("id1","id2"),dat)
mm2
summary(mm2)
```

BIreg *Fit a logistic regression model*

Description

BIreg is used to fit a logistic regression model, i.e., it conects the probability parameter of a binomial distribution with some given covariables by a logit link function. There is the option to include a dispersion parameter which will be estimated by the bias-corrected method of moments.

Usage

```
BIreg(formula,n,data,disp=FALSE)
```

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.
n	the maximum score of the Binomial trials.
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 the dispersion parameter will be estimated. Default mode is set to FALSE.

Details

The estimation of the regression parameters is done via maximum likelihood, considering the likelihood of the binomial response variable in the general form of a exponential family

$$f(y) = \frac{ylog(p/(1-p)) + nlog(1-p)}{\phi} + c(y,\phi).$$

The proabability parametern p has been connected with the given covariables ,$X_1,. . . ,X_t$, by the logit link function as,

$$logit(p) = log(\frac{p}{1-p}) = x_i'\beta,$$

where β are the regression coefficients and x_i is the ith row of a full rank design matrix X composed by the given covariables. Remplacing the prevous formula in the likelihodd of the response variable y, and by an iterative weigthed least squares algorithm, (`IWLS`), we get the estimations of the regression parameters.

If the `disp` option is TRUE, the dispersion parameter is going to be done using the bias-corrected method of moments, that is the same as the method of moments, but divided by the number of observation minus the number of parameters instead of only divided by the number of observations.

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.

The generic accessor functions `coefficients`, `fitted.values` and `residuals` can be used to extract various useful features of the value returned by `BIreg`.

`coefficients` the estimated value of the regression coefficients.

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

`phi` if the disp option is TRUE, it returns the estimated value of the dispersion parameter; the estimation is realized with the profile likelihood of the dispersion parameter, using the bias-corrected estimation. If the disp option is FALSE, the estimated value will be 1.

`fitted.values` the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.

`residuals` the working residuals, that is the residuals in the final iteration of the IWLS fit.

`deviance` the deviance of the model, i.e., minus twice the maximized log-likelihood.

`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 residual degrees of freedom for the null model.

`iter` number of iterations in the iterative weighted least squares method.

`X` the model matrix.

`y` the dependent variable in the model.

`n` the maximum score of the Binomial trials.

`noObs` number of observations in the data frame.

`call` the matched call.

`formula` the formula supplied.

Author(s)

Josu Najera

Dae-Jin Lee

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*, 31, 144-148

See Also

Iterative weighted least squares method function: `IWLS`

Examples

```
set.seed(1234)
# We simulate a covariable and we are going to construct the outcome
# variable usign the logit link function and the created covariable.

n <- 10                       # maximum score of the Binomial trials
k <-100                       # number of observations
covariable <- rnorm(k,2,0.5)  # the covariable

p <- 1/(1+exp(-(-6+4*covariable)))
outcome <- rbinom(k,n,p)

# without dispersion parameter
bat <- BIreg(outcome~covariable,n,disp=FALSE)
summary(bat)

# considering dispersion parameter
bi <- BIreg(outcome~covariable,n,disp=TRUE)
summary(bi)
```

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

Description

This function creates a spider plot with all the 8 dimensions of the SF-36 questionnaire indicating the HRQoL score of each observation.

Usage

```
HRQoLplot(data,legend)
```

Arguments

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

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

`legend` logical parameter, if TRUE the legend with the name of the rows of the data will appear. FALSE has been taken by default.

Details

The `HRQoLplot` has a data frame object as input, and an optional `legend` logical input. The data frame must have observations of the HRQoL taken by the SF-36 and the columns of the data frame will refer to each domain of the SF-36, hence the data frame must have eight columns. Each axis of the plot refers to an especific HRQoL domain, so the order of the data frame columns must be as it is showed in the arguments section. The HRQoL score of each observation (row) of the data frame is drawn with a line of a different color in the plot.

Author(s)

Josu Najera Zuloaga

Dae-Jin Lee

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

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(4,3,20,20,8,9,20,13)
k=3
p=runif(8,0,1)
phi <- runif(8,1,3)
dat <- data.frame(
  RP=rBB(k,n[1],p[1],phi[1]),
  RE=rBB(k,n[2],p[2],phi[2]),
  PF=rBB(k,n[3],p[3],phi[3]),
  VT=rBB(k,n[4],p[4],phi[4]),
  SF=rBB(k,n[5],p[5],phi[5]),
  BP=rBB(k,n[6],p[6],phi[6]),
  GH=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)
```

IWLS *Iterative Weighted Least Squares*

Description

IWLS gives a method to estimate the coefficients of logistic regressions by maximum likelihood.

Usage

```
IWLS(y,X,n)
```

Arguments

y	the dependent variable of the model.
X	the model matrix.
n	the maximum score of the binomial trials.

Details

The iterative weighted least squares (IWLS) is a general algorithm to find the maximum likelihood estimations (mle) and standard deviations in generalized linear mixed 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, IWLS, 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 and the the maximum score of the binomial trials.

Value

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

Author(s)

Josu Najera

Dae-Jin Lee

References

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

Examples

```
#we are going to create a variable and a matrix model
k=1000                   #number of observations
n=10                     #the maximum score of the binomial trials
y <- rbinom(k,n,0.87)    #dependent variable
x1 <- rnorm(k,1,50)      #covariable 1
x2 <- rnorm(k,30,9)      #covariable 2
X <- cbind(1,x1,x2)      #model matrix

IWLS(y,X,n)
```

SF36rec *Short Form-36 Helath Survey (SF-36) Recode*

Description

The SF36rec function recodes the Short Form-36 Health Survey (SF-36) dimensions into binomial data, based on *Arostegui et al. (2013).*

Usage

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

Arguments

x The dimension that is going to be recoded, it must be numeric and bounded between 0 and 100.

k An integer from 1 to 8, indicating the SF-36 dimension that x belongs. These are the dimensions depending on the k value:
k=1 -> Role Physical
k=2 -> Role Emotional
k=3 -> Physical Functioning
k=4 -> Vitality
k=5 -> Social Functioning
k=6 -> Body Pain
k=7 -> General Health
k=8 -> Mental Health

Details

Each observation of the HRQoL measured by the SF-36 is divided in eight domains, and each domain refers to an especific aspect of the HRQoL: *Role Physical, Role Emotional, Physical Functioning, Vitality, Social Functioning, Body Pain, General Health and Mental Health.* This domains are bounded between 0 and 100, and each domain can only take some values. The usual way of analyzing the domains is by the linear regression, but it has been proved that it is not correct *Arostegui et al. (2006).* The most natural way of analyzing the HRQoL data, and, thus, the eight domains, is by considering them binomially distributed *Arostegui et al. (2006).* The recodification to the most suitable form is evaluated in *Arostegui et al. (2013).* This function recodes the domains based on the method that is proposed in that article.

The form to use this function, is by inserting one by one each domain, and by defining with the input k which domain are we inserting, get the recodification to the binomial form of that domain.

Value

The dimension recoded to binomial data.

Author(s)

Josu Najera Zuloaga

Dae-Jin Lee

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*, 40:3, 563-583

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

```
set.seed(2)
#We simulate a variable bounded between 0 and 100, and that only can take some values.
BodyPain <- rnorm(1000,50,15)
k <- 6  #Because the domain we are inserting is the Body Pain
BodyPain.rec <- SF36rec(BodyPain,k)
```

Index