Package ‘binomTools’

February 19, 2015

Type Package
Title Performing diagnostics on binomial regression models
Version 1.0-1
Date 2011-08-03
Author Rune Haubo B Christensen and Merete K Hansen
Maintainer Merete K Hansen <mkh@imm.dtu.dk>
Description This package provides a range of diagnostic methods for
binomial regression models.
License GPL (>= 3)
LazyLoad yes
Repository CRAN
Date/Publication 2011-08-09 11:36:34
NeedsCompilation no

R topics documented:

beetles ................................................................. 2
doppels .................................................................... 2
exact.deletion ............................................................ 3
group ................................................................. 4
halfnorm ............................................................... 6
HLtest.Rsq ......................................................... 8
profile ............................................................... 9
Residuals ............................................................. 12
Rsq.glm ............................................................... 14
serum ................................................................. 16

Index 17
**Description**

Data from a study examining the response of confused flour beetles to increasing concentrations of gaseous carbon disulphide. After exposure for five hours the exact concentration of carbon disulphide was determined and the number of dead flour beetles were recorded. For each concentration of carbon disulphide duplicate batches of beetles were used.

**Usage**

data(beetles)

**Format**

A data frame with 16 observations on the following 4 variables.

- **conc** Concentration of carbon disulphide (mg/l)
- **rep** Replicate number
- **y** Number of deaths in each dose group
- **n** Total number of beetles in each dose group

**Source**


**References**


---

**empLogit**

*Calculates the empirical logit transform*

**Description**

The empirical logit transform allows for a tolerance such that infinity is not returned when the argument is zero or one.

**Usage**

empLogit(x, eps = 1e-3)
**exact.deleteion**

**Arguments**

- **x**: numerical vector for which the empirical logit transform is desired
- **eps**: numerical scalar; a tolerance to prevent infinite values

**Value**

the empirical logit transform of x

**Author(s)**

Rune Haubo B Christensen

**Examples**

```r
## The function is currently defined as
## function (x, eps = 1e-3) log((eps + x)/(1 - x + eps))
## Lifted from example(predict.glm):
ldose <- rep(0:5, 2)
numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
sex <- factor(rep(c("M", "F"), c(6, 6)))
SF <- cbind(numdead, numalive=20-numdead)
## budworm.lg <- glm(SF ~ sex*ldose, family=binomial)
## summary(budworm.lg)

empLogit(numdead/20)

## Possible usage:
## Explorotive interaction plot:
interaction.plot(ldose, sex, empLogit(numdead/20))
```

---

**exact.deleteion**

**Exact deletion residuals**

**Description**

Function to derive exact values of deletion (leave-one-out) residuals for binomial regression models

**Usage**

```r
exact.deleteion(object)
```

**Arguments**

- **object**: An object of class glm with a binomial family
Details

The $i$th deletion residual is calculated subtracting the deviances when fitting a linear logistic model to the full set of $n$ observations and fitting the same model to a set of $n - 1$ observations excluding the $i$th observation, for $i = 1, \ldots, n$. This gives rise to $n+1$ fitting processes and may be computationally heavy for large data sets.

Approximations to the deletion residuals, as described in Williams (1987), are provided by `rstudent`. Inconsistency regarding the terminology implies that the deletion residuals are called different names in the literature, including likelihood residuals, studentized residuals, externally studentized residuals, deleted studentized residuals and jack-knife residuals. Conversely, some of these terms refer to different types of residuals.

Value

A vector with exact deletion residuals

Author(s)

Merete K Hansen

References


See Also

`Residuals`, `rstudent`

Examples

```r
data(beetles)
beetles.glm <- glm(cbind(y, n-y) ~ log(conc), family=binomial, data=beetles)
exact.deletion(beetles.glm)
```

---

**group**

*Group observations in a binomial glm*

Description

This function groups the observations in a binomial glm based on the covariate structure. This can make it possible to assess goodness-of-fit in some models fitted to binary observations.
group

Usage

## S3 method for class 'glm'
group(object, eval = TRUE, ind = NULL, ...)

Arguments

- **object** a binomial glm object
- **eval** should the new glm-model be evaluated?
- **ind** an indicator for which rows to keep. If this is not specified the grouping structure is based on the covariate structure in the model.
- **...** currently not used

Details

The residual deviance and residual Pearson deviance are not meaningful measures of goodness-of-fit if the expected frequencies under the model are small (say less than five).

If `eval = TRUE` it is tested whether the estimated coefficients are identical up to three significant digits and a warning is issued if this is not the case. This should be the case in well-behaved situations but may not happen in cases of complete separation.

Value

A list with components

- **newCall** the new call
- **newData** a data frame with the aggregated data set
- **oldData** a data frame with the original data set
- **oldN** the number of rows (cases / observations) in the original data set
- **newN** the number of rows (cases / observations) in the aggregated data set
- **oldObject** the original fitted model
- **newObject** if `eval = TRUE` the new fitted model object, otherwise empty

Author(s)

Rune Haubo B Christensen

References

Examples

```r
## Lifted from example(predict.glm):
ldose <- rep(0:5, 2)
numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
sex <- factor(rep(c("M", "F"), c(6, 6)))
SF <- cbind(numdead, numalive=20-numdead)
## budworm.lg <- glm(SF ~ sex*ldose, family=binomial)
## summary(budworm.lg)
dat <- data.frame(SF=SF, sex, ldose)
dat[10, 1:2] <- rep(5, 2)
dat[13,] <- dat[10,]
rm(SF, sex, ldose)
SF <- as.matrix(dat[,1:2])
dat <- dat[,-(1:2)]
dat <- as.data.frame(cbind(SF, dat))

summary(m0 <- glm(SF ~ sex*ldose, binomial, dat))

## Various types of grouping:
(ind <- c(1:12, 10))
g <- group(m0, ind=ind, eval=TRUE)
g <- group(m0, eval=FALSE)
g <- group(m0, eval=TRUE)

## The correct GOF-test from the residual deviance is given by:
g$newObject
```

Description

halfnorm produces a half normal plot of the residuals with simulated envelopes useful for model evaluation and detection of outliers.

Usage

```r
halfnorm(object, resType = c("approx.deletion", "exact.deletion",
"standard.deviance", "standard.pearson", "deviance",
"pearson", "working", "response", "partial"), env = T,
nSIM = 20, plot = T, identify = F, n = 2)
```

Arguments

- `object` An object of class glm with a binomial family
- `resType` The type of residual used in the plot
halfnorm

env Logical for whether envelopes are simulated
nsim Number of simulations used for the envelopes
plot Logical for whether the points should be plotted. If plot = \( F \) a list is returned
identify Logical for whether it should be possible to identify points interactively. Ignored if plot = \( F \)
n How many points should be identified. Ignored if identify = \( F \)

Details

Absolute values of the residuals are used in a half normal plot that otherwise corresponds to a regular normal probability plot.

Residuals from a binomial glm are not necessarily uncorrelated and normally distributed and may accordingly deviate from a straight line even if the fitted model is true. If the fitted model is true the optional simulated envelopes are likely to contain the absolute residuals.

The different types of residuals are described in Residuals

Value

If plot = \( T \) a plot is produced. Otherwise a list of the residuals and their expected values are returned

Author(s)

Merete K Hansen

References


See Also

Residuals, identify

Examples

```r
## Halfnormal plot with simulated envelopes
data(beetles)
beetles glm <- glm(cbind(y, n-y) ~ conc, family=binomial, data=beetles)
halfnorm(beetles glm, restype='pearson')

## Not run:
## Halfnormal plot with simulated envelopes
## Two points are interactively identified when they are selected with the mouse
halfnorm(beetles glm, restype='deviance', identify = T, n = 2)

## End(Not run)
```
Description
Goodness-of-fit tests for GLMs for binary data including the Hosmer-Lemeshow decile test and X-squared test with normal approximation.

Usage

```r
## S3 method for class 'Rsq'
HLtest(object, method = c("deciles", "fixed"),
       decile.type = 8, ...)

## S3 method for class 'HLtest.Rsq'
print(x, digits = getOption("digits"), ...)

## S3 method for class 'Rsq'
X2GOFtest(x, ...)

## S3 method for class 'X2GOFtest.Rsq'
print(x, ...)
```

Arguments
- object: An Rsq object
- x: An HLtest.Rsq or an X2GOFtest.Rsq object
- method: The type of Hosmer-Lemeshow test to be performed. The "deciles" method should be more accurate (Hosmer et al, 1997)
- decile.type: The quantile computation method; see `quantile` for details
- digits: the desired number of printed digits
- ...: currently not used

Details
These tests are known to have very low power. They are only appropriate when the fitted frequencies are very low and when the covariate pattern dictates strictly binary observations.

Value
For HLtest.Rsq an object of class HLtest.Rsq with components
- expected: the expected frequencies in the 2 x 10 entries
- observed: the observed frequencies in the 2 x 10 entries
- resid: Pearson residuals
X2  the Pearson X-squared statistic
p.value  the p-value for the goodness-of-fit test
method  the method used for the test

For X2GOFtest an object of class X2GOFtest with components
p.value  the p-value for the goodness-of-fit test
z.score  the standardized z-score for the goodness-of-fit test
RSS  the residual sums of squares term
X2  the pearson chi-squared statistic

Author(s)
Rune Haubo B Christensen

References

Examples
```r
## Lifted from example(predict.glm):
ldose <- rep(0:5, 2)
numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
sex <- factor(rep(c("M", "F"), c(6, 6)))
SF <- cbind(numdead, numalive=20-numdead)
budworm.lg <- glm(SF ~ sex*ldose, family=binomial)
## summary(budworm.lg)

(Rsq.budworm <- Rsq(budworm.lg))

HLtest(Rsq.budworm)
HLtest(Rsq.budworm, method="fixed")
X2GOFtest(Rsq.budworm)
```

**Description**
Generate and plot the profile likelihoods for each parameter in a binomial regression model
Usage

```r
# S3 method for class 'glm'
profile(fitted, which.par, alpha = 0.005, max.steps = 50,
nsteps = 8, step.warn = 5, trace = F, ...)

# S3 method for class 'profile.glm'
plot(x, which.par, likelihood = TRUE,
    log = FALSE, relative = TRUE, approx = TRUE, conf.int = TRUE,
    level = 0.95, n = 100, fig = TRUE, ylim = NULL, ...)
```

Arguments

- **fitted**: An object of class `glm` with a binomial family
- **x**: An object of class `profile.glm`
- **which.par**: A numeric or character vector with the parameters to be profiled. If missing all parameters are profiled
- **alpha**: The likelihood is profiled in approximately the 100*(1-alpha)% confidence region
- **likelihood**: Logical for whether the profile likelihood or likelihood root should be plotted
- **log**: Logical for whether the profile likelihood should be plotted on log-scale. Ignored if `likelihood` = FALSE
- **relative**: Logical for whether the profile likelihood or log-likelihood should be plotted on a relative or absolute scale. Ignored if `likelihood` = FALSE
- **approx**: Logical for whether a quadratic approximation should be included in the plot
- **conf.int**: Logical for whether a confidence interval should be included in the plot
- **level**: A scalar or numerical vector indicating the confidence level(s) to be included in the plot. Ignored if `conf.int` = FALSE
- **n**: How many points to employ in the spline interpolation of the profile likelihood
- **fig**: Logical for whether the profile likelihood should be plotted. If `fig` = FALSE the list of points from the spline interpolation is returned
- **ylim**: The limits of the y-axis in the plot
- **trace**: Logical for whether progress should be printed to the screen during the profiling process
- **nsteps**: Number of profiling steps to take in each direction for each parameter. The number is approximate since the step size is determined according to a quadratic approximation to the profile log-likelihood, hence, the deviation of the value of `nsteps` to the actual number of steps performed is influenced by the degree of irregularity of the profile likelihood
- **max.steps**: The maximum number of profiling steps in each direction for each parameter. A warning is issued if the number of `max.steps` is reached
- **step.warn**: A warning is issued if the actual number of steps in either direction does not exceed the number of `step.warn`
- **...**: Additional arguments passed to other methods
Details

lrroot returned by profile is the signed square-root of the usual profile likelihood

\[ \text{sgn}(\theta - \hat{\theta})\sqrt{2(l(\hat{\theta}) - l(\theta))} \]

where \( \theta \) is the parameter being profiled and \( \hat{\theta} \) is the maximum likelihood estimate of \( \theta \). The appertaining par.vals is a vector of \( \theta \) values in an appropriate range around \( \hat{\theta} \).

The logical argument likelihood in plot controls if the profile likelihood or the likelihood root should be plotted.

Value

For profile: a list of class profile.glm with a range of parameter values and lrroot statistics for each parameter in which.par

For plot: if \( \text{fig} = \text{FALSE} \) a list with plotting points and confidence interval(s) for each parameter in which.par is returned. If \( \text{fig} = \text{TRUE} \) the list is returned invisibly.

Note

The implementation of these functions are largely inspired by profile.glm from the MASS package and profile.clm from the ordinal package. This work is a direct extension of profile from MASS with an extended set of warnings. The main difference, though, is in the plotting functionality, which enables plot of the usual profile likelihood and log-likelihood and the optional inclusion of confidence interval(s).

Author(s)

Merete K Hansen

References


See Also

glm, profile.glm, plot.profile

Examples

data(serum)
serum glm <- glm(cbind(y, n-y) ~ dose, family=binomial, data=serum)
pr <- profile(serum glm)
plot(pr)
Residuals

Residuals from a binomial regression model

Description

Function to extract residuals from a binomial regression model

Usage

Residuals(object, type = c("approx.deletion", "exact.deletion",
"standard.deviance", "standard.pearson", "deviance",
"pearson", "working", "response", "partial"))

Arguments

object An object of class glm with a binomial family

type The type of residuals to be returned. Default is approx.deletion residuals

Details

A considerable terminology inconsistency regarding residuals is found in the litterature, especially concerning the adjectives standardized and studentized. Here, we use the term standardized about residuals divided by \(\sqrt{1 - h_i}\) and avoid the term studentized in favour of deletion to avoid confusion. See Hardin and Hilbe (2007) p. 52 for a short discussion of this topic.

The objective of Residuals is to enhance transparency of residuals of binomial regression models in R and to uniformise the terminology. With the exception of exact.deletion all residuals are extracted with a call to rstudent, rstandard and residuals from the stats package (see the description of the individual residuals below).

- response: response residuals
  \[ y_i - \hat{y}_i \]
  The response residuals are also called raw residuals
  The residuals are extracted with a call to residuals.

- pearson: Pearson residuals
  \[ X_i = \frac{y_i - n_i \hat{p}_i}{\sqrt{n_i \hat{p}_i (1 - \hat{p}_i)}} \]
  The residuals are extracted with a call to residuals.

- standard.pearson: standarded Pearson residuals
  \[ r_{P,i} = \frac{X_i}{\sqrt{1 - h_i}} = \frac{y_i + n_i \hat{p}_i}{\sqrt{n_i \hat{p}_i (1 - \hat{p}_i)(1 - h_i)}} \]
  where \(X_i\) are the Pearson residuals and \(h_i\) are the hatvalues obtainable with hatvalues.
  The standardized Pearson residuals have many names including studentized Pearson residuals, standardized residuals, studentized residuals, internally studentized residuals.
  The residuals are extracted with a call to rstandard.
Residuals

- deviance: deviance residual
  The deviance residuals are the signed square roots of the individual observations to the overall deviance
  \[ d_i = sgn(y_i - \hat{y}_i) \sqrt{2y_i \log \frac{y_i}{\hat{y}_i}} + 2(n_i - y_i) \log \frac{n_i - \hat{y}_i}{n_i - y_i} \]
  The residuals are extracted with a call to `residuals`.

- standard.deviance: standardized deviance residuals
  \[ r_{D,i} = \frac{d_i}{\sqrt{1 - h_i}} \]
  where \( d_i \) are the deviance residuals and \( h_i \) are the hatvalues that can be obtained with `hatvalues`.
  The standardized deviance residuals are also called studentized deviance residuals.
  The residuals are extracted with a call to `rstandard`.

- approx.deletion: approximate deletion residuals
  \[ sgn(y_i - \hat{y}_i) \sqrt{h_i r_{P,i}^2 + (1 - h_i) r_{D,i}^2} \]
  where \( r_{P,i} \) are the standardized Pearson residuals, \( r_{D,i} \) are the standardized deviance residuals and \( h_i \) are the hatvalues that is obtained with `hatvalues`.
  The approximate deletion residuals are approximations to the exact deletion residuals (see below) as suggested by Williams (1987).
  The approximate deletion residuals are called many different names in the litterature including likelihood residuals, studentized residuals, externally studentized residuals, deleted studentized residuals and jack-knife residuals.
  The residuals are extracted with a call to `rstudent`.

- exact.deletion: exact deletion residuals
  The \( i \)th deletion residual is calculated subtracting the deviances when fitting a linear logistic model to the full set of \( n \) observations and fitting the same model to a set of \( n - 1 \) observations excluding the \( i \)th observation, for \( i = 1, ..., n \). This gives rise to \( n + 1 \) fitting processes and may be computationally heavy for large data sets.

- working: working residuals
  The difference between the working response and the linear predictor at convergence
  \[ r_{W,i} = (y_i - \hat{y}_i) \frac{\partial \hat{y}_i}{\partial \mu_i} \]
  The residuals are extracted with a call to `residuals`.

- partial: partial residuals
  \[ r_{W,i} + x_{ij} \hat{\beta}_j \]
  where \( j = 1, ..., p \) and \( p \) is the number of predictors. \( x_{ij} \) is the \( i \)th observation of the \( j \)th predictor and \( \hat{\beta}_j \) is the \( j \)th fitted coefficient.
  The residuals are useful for making partial residuals plots. They are extracted with a call to `residuals`
Value

A vector of residuals

Author(s)

Merete K Hansen

References


Examples

```r
data(serum)
serum(glm<glm(cbind(y, n-y) ~ log(dose), family = binomial, data = serum)
Residuals(serum glm, type='standard.deviance')
```

---

<table>
<thead>
<tr>
<th>Rsq.glm</th>
<th>R-squared measures for binomial GLMs</th>
</tr>
</thead>
</table>

Description

This function computes the R-squared measures for binomial GLMs proposed by Tjur (2010) "Coefficients of determination in logistic regression models - a new proposal: The coefficient of discrimination".

Usage

```r
## S3 method for class 'glm'
Rsq(object, ...)

## S3 method for class 'Rsq'
print(x, digits = getOption("digits"), ...)

## S3 method for class 'Rsq'
plot(x, which=c("hist", "ecdf", "ROC"), ...)
```
Arguments

- **object**: a binomial `glm` object
- **x**: an Rsq object
- **which**: the desired plot: histograms, empirical cumulative distribution functions or ROC (receiver operating characteristic) curve
- **digits**: the desired number of printed digits
- **...**: currently not used

Details

The plot method has the following options:

- "hist": Two histograms with ten bins of the fitted probabilities are plotted on top of each other; the upper one for \( y = 0 \) and the lower one for \( y = 1 \).
- "ecdf": Two ecdf curves; one for \( y = 0 \) and one for \( y = 1 \)
- "ROC": The (empirical) ROC curve

Value

Rsq.glm returns an object of class Rsq. The plot and print methods returns the Rsq objects invisibly.

Author(s)

Rune Haubo B Christensen

References


See Also

A `HLtest` (Hosmer and Lemeshow test) method exists for Rsq objects.

Examples

```r
## Lifted from example(predict.glm):
ldose <- rep(0:5, 2)
numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
sex <- factor(rep(c("M", "F"), c(6, 6)))
SF <- cbind(numdead, numalive=20-numdead)
budworm.lg <- glm(SF ~ sex*ldose, family=binomial)
## summary(budworm.lg)

(Rsq.budworm <- Rsq(budworm.lg))

plot(Rsq.budworm, "hist") ## or simply 'plot(Rsq.budworm)
plot(Rsq.budworm, "ecdf")
plot(Rsq.budworm, "ROC")
```
Description

Data from a study examining the protective effect of a serum co-administered in increasing doses with an infecting dose of a pneumococci culture. Each dose group consisted of 40 mice \( (n) \) and the number of deaths caused by pneumonia was recorded \( (y) \).

Usage

data(serum)

Format

A data frame with 5 observations on the following 3 variables.

- **dose**: Dose of the anti-pneumococcus serum administered
- **y**: Number of deaths in each dose group
- **n**: Total number of mice in each dose group

Source


References

Index

*Topic datasets
  beetles, 2
  serum, 16
*Topic dplot
  halfnorm, 6
*Topic iplot
  halfnorm, 6
*Topic models
  exact.deletion, 3
group, 4
  halfnorm, 6
  profile, 9
  Residuals, 12
  Rsq.glm, 14
*Topic regression
  profile, 9
*Topic tests
  group, 4
  HLtest.Rsq, 8
  Rsq.glm, 14
*Topic utilities
  empLogit, 2
  plot.Rsq(Rsq.glm), 14
  print.HLtest.Rsq(HLtest.Rsq), 8
  print.Rsq(Rsq.glm), 14
  print.X2GOFtest.Rsq(HLtest.Rsq), 8
  profile, 9
    profile.clm, 11
    profile.glm, 11
  profile.glm(profile), 9
  quantile, 8
  Resid(Residuals), 12
  Residuals, 4, 7, 12
  residuals, 12, 13
  Rsq(Rsq.glm), 14
  Rsq.glm, 14
  rstandard, 12, 13
  rstudent, 4, 12, 13
  serum, 16
  X2GOFtest(HLtest.Rsq), 8

beetles, 2
empLogit, 2
exact.deletion, 3
glm, 11, 15
group, 4
halfnorm, 6
hatvalues, 12, 13
HLtest, 15
HLtest(HLtest.Rsq), 8
HLtest.Rsq, 8
identify, 7
plot.profile, 11
plot.profile.glm(profile), 9