

Package ‘LOGIT’

February 6, 2016

Type Package

Title Functions, Data and Code for Binary and Binomial Data

Version 1.3

Date 2015-09-28

Author Joseph M Hilbe <hilbe@asu.edu>, Rafael S. de Souza <rafael.2706@gmail.com>

Maintainer Rafael S. de Souza <rafael.2706@gmail.com>

Description Functions, data and code for Hilbe, J.M. 2015. Practical Guide to Logistic Regression, by Chapman and Hall/CRC.

Depends R (>= 3.0)

License GPL (>= 3)

Imports caret, pROC, ggplot2, reshape, e1071, MASS

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2016-02-06 11:20:21

R topics documented:

azheart	2
badhealth	3
confusion_stat	4
edrelig	5
hiv	6
hlGOF.test	7
HLTest	9
iQuantile	10
jhbayes	11
medpar	12
mylgg	14
P__disp	15
R84	16

ROCtest	17
rwm1984	19
titanicgrp	20
toOR	22
toRR	23
Index	25

azheart

azheart

Description

Random subset of 34 patients from the 1991 Arizona Medicare data for patients hospitalized subsequent to undergoing a CABG (DRGs 106, 107) or PTCA (DRG 112) cardiovascular procedure.

Usage

```
data(azheart)
```

Format

A data frame with 34 observations on the following 6 variables.

died 1=died as a result of surgery; 0=not died

procedure 1=CABG; 0=PTCA

age age of subject

gender 1=Male; 0=Female

los hospital length of stay

type 1=emerg/urgent admission; 0=elective admission

Details

azheart is saved as a data frame.

Source

Hilbe, Practical Guide to Logistic Regression, Chapman & Hall/CRC

References

Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC

Examples

```

library(LOGIT)
#library(COUNT)
data(azheart); attach(azheart)
table(los); table(procedure, type); table(los, died)
summary(los)
summary(mymod <- glm(died ~ procedure + type + los, family=binomial, data=azheart))
#modelfit(mymod)
summary(mymodq <- glm(died ~ procedure+ type + los, family=quasibinomial, data=azheart))
#modelfit(mymodq)
#library(sandwich)
#sqrt(diag(vcovHC(mymod, type="HC0")))
toOR(mymod)

```

badhealth

badhealth

Description

A study of 1127 patients in which only three variables have been selected. The data may be evaluated as a logistic or other binary response model with the binary variable "badh" as the response. It may also be modeled as a count model with "numvisit" (number of visits to a physician during the year) as the response. "age" is an adjustor, and should be centered or standardized when used in the model.

Usage

```
data(badhealth)
```

Format

A data frame with 1127 observations with 3 variables.

numvisit Number of visits to a physician during the year: 0 - 40

badh 0=patient evaluates self as in good health; 1=patient in bad health

age patient age: 20 - 60

Details

badhealth is saved as a data frame.

Source

Hilbe, Practical Guide to Logistic Regression, Chapman & Hall/CRC

References

Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC

Examples

```
library(LOGIT)
data(badhealth)
age.std <- scale(badhealth$age)
summary(myhealth<- glm(badh ~ numvisit + age.std, family=binomial, data=badhealth))
toOR(myhealth)
```

confusion_stat	<i>Display confusion or classification matrix following logistic regression using glm with binomial family.</i>
----------------	---

Description

Provides a confusion matrix of classification statistics following logistic regression.

Usage

```
confusion_stat(pred=pred, obs=obs)
```

Arguments

pred	Predicted values
obs	Observed values

Format

x The function has two arguments: predicted values, response values

Value

confusion matrix

Note

confusion_stat() must be loaded into memory in order to be effective. As a function in LOGIT, it is immediately available to a user.

Author(s)

Rafael de Souza, ELTE University, and Joseph M. Hilbe, Arizona State University

References

- Hilbe, Joseph M. (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.
 Hilbe, Joseph M. (2009), Logistic Regression Models, Chapman & Hall/CRC.

See Also

[glm](#)

Examples

```
library(MASS)
library(LOGIT)
data(R84)
R84$cage <- R84$age - mean(R84$age)
R84$cdoc <- R84$docvis - mean(R84$docvis)
mylogit <- glm(outwork ~ cdoc + female + kids + cage + factor(edlevel),
  family=binomial, data=R84)
mu <- predict(mylogit, type="response")
cutpoint<-ROCTest(mylogit, fold=10, type="Sensitivity")$cut
mu[mu>=cutpoint]<-1
mu[mu<cutpoint]<-0
confusion_stat(mu, R84$outwork)
```

 edrelig

edrelig

Description

A study of 601 subjects aimed at determining predictors for religiosity. Subjects self-identify as being religious or not. Predictors include gender, age, having children, and education level.

Usage

```
data(edrelig)
```

Format

A data frame with 601 observations of grouped data with 5 variables.

male 0=female; 1=male

age ages from 17.5 to 57

kids 0=not have children; 1=has children

educlevel 1=AA degree; 2=BA; 3=MA/PhD

religious 0=no; 1=yes

Details

edrelig is saved as a data frame.

Source

Hilbe, Practical Guide to Logistic Regression, Chapman & Hall/CRC.

References

Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.

Examples

```
library(LOGIT)
data(edrelig)
head(edrelig)
edrelig$cage <- edrelig$age - mean(edrelig$age)
isrelig<- glm(religious ~ male + cage + kids + factor(educlevel), family=binomial, data=edrelig)
summary(isrelig)
toOR(isrelig)
```

 hiv

hiv

Description

A study of 47 patients with similar profiles. Measurements on cd4 and cd8 levels are displayed for each panel of patients having identical predictor profiles. Both cd4 and cd8 have three levels. The data should be modeled as a grouped logistic model, but may also be modeled as a count model depending on what a research wishes to determine from the data.

Usage

```
data(hiv)
```

Format

A data frame with 11 observations of grouped data with 4 variables.

infec 1=Patient diagnosed with HIV; 0=not diagnosed with HIV

cases number of patients for each patient profile

cd4 3 levels: 0, 1, and 2

cd8 3 levels: 0, 1, and 2

Details

hiv is saved as a data frame.

Source

Hilbe, Practical Guide to Logistic Regression, Chapman & Hall/CRC.

References

Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.

Examples

```
# Not run
#data(hiv)
#table(hiv); hiv
#noinfec <- hiv$cases - hiv$infec
#myhiv<- glm(cbind(infec, noinfec) ~ factor(cd4) + factor(cd8), family=binomial, data=hiv)
#summary(myhiv)
#mymodq <- glm( cbind(infec, noinfec) ~ factor(cd4) + factor(cd8), family=quasibinomial, data=hiv)
#summary(mymodq)
#toOR(myhiv)
#End(Not run)

library(LOGIT)
data(hiv)
table(hiv); hiv
noinfec <- hiv$cases - hiv$infec
response <- cbind(hiv$infec, noinfec)
myhiv<- glm(response ~ factor(cd4) + factor(cd8), family=binomial, data=hiv)
summary(myhiv)

mymodq <- glm(response ~ factor(cd4) + factor(cd8), family=quasibinomial, data=hiv)
summary(mymodq)

#library(sandwich)
#sqrt(diag(vcovHC(myhiv, type="HC0")))
toOR(myhiv)
```

h1GOF.test

Display Hosmer-Lemeshow statistic and table of probabilities following logistic regression using glm with binomial family.

Description

Provides a Hosmer-Lemeshow statistic and table following logistic regression.

Usage

```
h1GOF.test(observed, predicted, breaks = 15)
```

Arguments

observed	response variable
predicted	predicted statistic
breaks	breaks or groups

Format

x The function has three arguments: observed term, predicted values, # groups

Details

hlGOF.test is a post-estimation function for logistic regression, following the use of glm(). Usage displays a table of observed vs predicted groups and an overall H-L goodness-of-fit statistic. The test is originally from Hilbe (2009).

Value

numeric

Note

hlGOF.test must be loaded into memory in order to be effective. As a function in LOGIT, it is immediately available to a user. My thanks to Prof. Robert LaBudde for the initial version of this function.

Author(s)

Joseph M. Hilbe, Arizona State University, Robert LaBudde, Institute for Statistical Education (Statistics.com), provided initial code for this function for Hilbe, Logistic Regression Models, text.

References

Hilbe, J. M. (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.

Hilbe, J. M. (2009), Logistic Regression Models, Chapman & Hall/CRC.

See Also

[glm](#)

Examples

```
library(MASS)
library(LOGIT)
data(medpar)
mylogit <- glm( died ~ los + white + hmo, family=binomial, data=medpar)
summary(mylogit)
medpar2 <- na.omit(medpar)
hlGOF.test(medpar2$died, predict(mylogit,medpar2, type="response"), breaks=12)
```

HLTest	<i>Display Hosmer-Lemeshow statistic and table of probabilities following logistic regression using glm with binomial family.</i>
--------	---

Description

Provides a Hosmer-Lemeshow statistic and table following logistic regression.

Usage

```
HLTest(obj, g)
```

Arguments

obj	model name
g	number of groups

Format

x The function has two arguments: model name, number of groups

Details

HLTest is a post-estimation function for logistic regression, following the use of glm(). Usage displays a table of observed vs predicted groups and an overall H-L goodness-of-fit statistic.

Value

list

Note

HLTest must be loaded into memory in order to be effective. As a function in LOGIT, it is immediately available to a user. My thanks to Bilger and Loughlin for the use of their function.

Author(s)

Adapted from Loughlin, T.M. in Bilder and Loughlin, 2015

References

- Hilbe, J. M. (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.
Bilder, C.R. and Loughlin, T.M. (2015), Analysis of Categorical Data with R, Chapman & Hall/CRC.
Hilbe, J. M. (2009), Logistic Regression Models, Chapman & Hall/CRC.
Hosmer, D.W., Lemeshow, S, and Sturdivant, R.X (2013), Applied Logistic Regression, 3rd ed, Wiley.

See Also[glm](#)**Examples**

```
library(MASS)
library(LOGIT)
data(medpar)
mylogit <- glm( died ~ los + white + hmo, family=binomial, data=medpar)
grp10 <- HLTtest(obj=mylogit, g=10)
cbind(grp10$observed, round(grp10$expect, digits = 1))
grp10
```

*iQuantile**iQuantile*

Description

Produces indices in x of percentile steps by $1/\text{breaks}$ and respective percentiles.

Usage

```
iQuantile(x, breaks=15)
```

Arguments

x	list
breaks	number of breaks

Format

x The function has two arguments: a list and number of breaks.

Details

iQuantile is used internally by the *hlGOF.test*.

Value

numeric

Author(s)

Joseph M. Hilbe, Arizona State University.

References

Hilbe, J. M. (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.
Hilbe, J. M. (2009), Logistic Regression Models, Chapman & Hall/CRC.

Examples

```
library(LOGIT)
mod <- rnorm(100,0,1)
iQuantile(mod)
```

jhbayes

Two functions to provide better JAGS model output

Description

For use with JAGS from within the R environment. Provides a nicer model output than comes with the default JAGS output.

Usage

```
#source("jhbayes.r")
```

Arguments

x Variable arguments based on model

Details

Load jhbayes.r prior to running JAGS model. MyBUGSOutput and uNames functions will then be in memory. From Alain Zuur support files on highstat.com.

Note

jhbayes.r must be loaded into memory in order to be effective. Use the source function or paste into R editor. Code is 23 lines in length.

Author(s)

Alain F. Zuur, Highlands Statistics, UK. highstat@highstat.com Joseph M. Hilbe, Arizona State University, and Jet Propulsion Laboratory, California Institute of technology hilbe@asu.edu

References

- Hilbe, Joseph M. (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC, page 137-143.
- Zuur, A.F., Hilbe, J.M., and Ieno, E.N. (2013), A Beginner's Guide to GLM and GLMM with R: a frequentist and Bayesian perspective for ecologists, Highlands.

Examples

```
#library(R2jags)
#library(LOGIT)
#data(medpar)
#JAGS code with J0 as MCMC algorithm
# out <- J0$BUGS$output
# myB <- MyBUGSOutput(out, c(uNames("beta", K), "LogL", "AIC", "BIC"))
# round(myB, 4)
```

 medpar

medpar

Description

The US national Medicare inpatient hospital database is referred to as the Medpar data, which is prepared yearly from hospital filing records. Medpar files for each state are also prepared. The full Medpar data consists of 115 variables. The national Medpar has some 14 million records, with one record for each hospitalization. The data in the medpar file comes from 1991 Medicare files for the state of Arizona. The data are limited to only one diagnostic group (DRG 112). Patient data have been randomly selected from the original data.

Usage

```
data(medpar)
```

Format

A data frame with 1495 observations on the following 10 variables.

```
los length of hospital stay
hmo Patient belongs to a Health Maintenance Organization, binary
white Patient identifies themselves as Caucasian, binary
died Patient died, binary
age Patient age range, categorical
age80 Patient age 80 and over, binary
type Type of admission, categorical
type1 Elective admission, binary
type2 Urgent admission, binary
type3 Elective admission, binary
provnum Provider ID
```

Details

medpar is saved as a data frame. Count models use los as response variable. 0 counts are structurally excluded. The data is also used to predict death as well as to understand the predictors which bear on the death of a patient while in the hospital following surgery.

Source

1991 National Medpar data, National Health Economics & Research Co.

References

Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.

Hilbe, Joseph M (2014), Modeling Count Data, Cambridge University Press.

Hilbe, Joseph M (2007, 2011), Negative Binomial Regression, Cambridge University Press.

Hilbe, Joseph M (2009), Logistic Regression Models, Chapman & Hall/CRC.

first used in Hardin, JW and JM Hilbe (2001, 2007), Generalized Linear Models and Extensions, Stata Press.

Examples

```
# library(MASS) if not automatically loaded

# medpar in both LOGIT and COUNT packages
library(LOGIT)
data(medpar)
glmb <- glm( died ~ los + hmo + white + factor(type), family=binomial, data=medpar)
summary(glmb)
toOR(glmb)

library(LOGIT)
data(medpar)
summary(glmnb <- glm( died ~ los + hmo + white + factor(type),
                    family=binomial(link=probit), data=medpar))

library(LOGIT) # or library(COUNT)
data(medpar)
medpar$los<-as.numeric(medpar$los)
glmnb <- glm(los ~ hmo + white + factor(type), family=poisson, data=medpar)
summary(glmnb)
exp(coef(glmnb))
toRR(glmnb)

#library(COUNT) # nbinomial in both COUNT and msme packages
#data(medpar)
#nb2 <- nbinomial(los ~ hmo + white + factor(type), data=medpar)
#summary(nb2)
#exp(coef(nb2))

#library(LOGIT) # or library(COUNT)
#data(medpar)
#glmnb <- glm.nb(los ~ hmo + white + factor(type), data=medpar)
#summary(glmnb)
#exp(coef(glmnb))
#toRR(glmnb)
```

`mylgg``mylgg`

Description

The data has 11 grouped observations and 6 variables. Grouped subset of medpar data.

Usage

```
data(mylgg)
```

Format

A data frame with 11 observations with the following 6 variables.

`white` 0=identified as non-white; identified as white

`hmo` 0=patient not an HMO member; 1=patient member of HMO

`type` Type of admission: 1=elective;2=urgent; 3=emergency

`alive` # patients alive per patient profile

`dead` # patients died within 48 hrs admit per patient profile

`m` # patients in each patient profile (same predictor values)

Details

`mylgg` is saved as a data frame. Used to assess odds ratios and predict survival following surgery

Source

Subset of medpar data, grouped format.

References

Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.

Hilbe, Joseph M (2009), Logistic Regression Models, Chapman & Hall/CRC.

Examples

```
library(MASS) # if not automatically loaded
# LOGISTIC REGRESSION
library(LOGIT)
data(mylgg)
mylgg
summary(lg <- glm(cbind(alive, dead) ~ white + hmo + factor(type),
                  family=binomial, data=mylgg))
toOR(lg)
P__disp(lg)
```

P__disp	<i>Display Pearson Chi2 and associated dispersion statistic following following use of glm.</i>
---------	---

Description

Following the `glm()` function with a grouped binomial or poisson family, or `glm.nb()`, `P__disp()` displays the Pearson Chi2 statistic and related dispersion statistic. Values of the dispersion greater than 1.0 indicate possible overdispersion; values under 1.0 indicate possible underdispersion.

Usage

```
P__disp(x)
```

Arguments

x glm object

Format

x The only argument is the name of the fitted glm or glm.nb function model

Details

`P__disp` is a post-estimation function, following the use of `glm()` or `glm.nb()`. Appropriate with grouped binomial or Poisson glm families.

Value

Pearson Chi2 Pearson Chi2 statistic

Dispersion Pearson dispersion: Chi2/dof

Note

`P__disp` must be loaded into memory in order to be effective. As a function in LOGIT, it is immediately available to a user.

Author(s)

Joseph M. Hilbe, Arizona State University, and Jet Propulsion Laboratory, California Institute of technology

References

Hilbe, Joseph M. (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC. Hilbe, Joseph M. (2014), Modeling Count Data, Cambridge University Press

See Also[glm](#)**Examples**

```

library(MASS)
library(LOGIT)
data(titanicgrp)
class03 <- factor(titanicgrp$class, levels=c("3rd class", "2nd class", "1st class"))
died <- titanicgrp$cases - titanicgrp$survive
grptit <- glm( cbind(survive, died) ~ age+sex+class03, family=binomial,
data=titanicgrp)
summary(grptit)
P__disp(grptit)

```

R84

*R84***Description**

German health reform data for the year 1984. Subset of a multiyear registry evaluating differences in physician provider utilization prior to and after health reform legislation in the late 1980s. R84 is identical to rwn1984.

Usage

```
data(R84)
```

Format

A data frame with 3,874 observations on the following 15 variables.

```

outwork out of work=1; 0=working
docvis number of visits to doctor during year (0-121)
hospvis number of days in hospital during year (0-51)
edlevel educational level (categorical: 1-4)
age age: 25-64
female female=1; 0=not female
married married=1; 0=not married
kids have children=1; no children=0
hhninc household yearly income in marks (in Marks)
educ years of formal education (7-18)
self self-employed=1; not self employed=0
edlevel1 (1/0) not high school graduate
edlevel2 (1/0) high school graduate
edlevel3 (1/0) university/college
edlevel4 (1/0) graduate school

```


Details

R84 is saved as a data frame. The data is typically used to model docvis, which is a count variable. It also may be used to model "outwork", a variable indicating if a patient is out-of-work. "outwork" is a binary variable which can be used as the response of a logistic or other binary response model. R84 is identical to rwm1984.

Source

German Health Reform Registry for the year 1984, in Hilbe and Greene (2007)

References

Hilbe, Joseph M (2009), Logistic Regression Models, Chapman & Hall/CRC.
 Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.
 Hilbe, Joseph M (2011), Negative Binomial Regression, 2nd ed., Cambridge University Press.
 Hilbe, Joseph M (2014), Modeling Count Data, Cambridge University Press.
 Hardin & Hilbe (2013), Generalized Linear Models & Extensions, 3rd ed, Stata Press.

Examples

```
# library(MASS)  if not automatically loaded
library(LOGIT)
# library(COUNT) R84 also in COUNT package, but not toOR or P_disp
data(R84)
# center both docvis and age
R84$cage <- R84$age - mean(R84$age)
R84$cdoc <- R84$docvis - mean(R84$docvis)
glmrp <- glm(outwork ~ cdoc + female + kids + cage + factor(edlevel),
             family=binomial, data=R84)
summary(glmrp)
exp(coef(glmrp))
toOR(glmrp)
```

ROCTest

Display ROC curve and related AUC statistic, or sensitivity-specificity plot of glm with binomial family.

Description

Provides two options following the glm() function with binomial family. 1: Sensitivity-specificity plot with optimal cut point statistic 2: ROC plot with Area Under Curve (AUC) statistic

Usage

```
ROCTest(model= model, fold=10, type=c("ROC", "Sensitivity"))
```

Arguments

model	model name
fold	number of k-folds
type	type of plot

Format

x The function has three arguments: modelname, folds, type of plot

Details

ROCtest is a post-estimation function for logistic regression, following the use of glm(). Options to display a sensitivity-specificity plot or ROC curve are available.

Value

plot

Note

ROCtest() must be loaded into memory in order to be effective. As a function in LOGIT, it is immediately available to a user.

Author(s)

Rafael de Souza, ELTE, Hungary, Joseph M. Hilbe, Arizona State University.

References

Hilbe, Joseph M. (2016), Practical Guide to Logistic Regression, Chapman & Hall/CRC. Hilbe, Joseph M. (2009), Logistic Regression Models, Chapman & Hall/CRC.

See Also

[glm](#)

Examples

```
library(MASS)
library(LOGIT)
data(R84)
R84$cage <- R84$age - mean(R84$age)
R84$cdoc <- R84$docvis - mean(R84$docvis)
mylogit <- glm(outwork ~ cdoc + female + kids + cage + factor(edlevel),
family=binomial, data=R84)
summary(mylogit)
ROCtest(mylogit, fold=10, type="Sensitivity")
ROCtest(mylogit, fold=10, type="ROC")
```

*rwm1984**rwm1984*

Description

German health reform data for the year 1984. Subset of a multiyear registry evaluating differences in physician provider utilization prior to and after health reform legislation in the late 1980s.

Usage

```
data(rwm1984)
```

Format

A data frame with 3,874 observations on the following 15 variables.

outwork out of work=1; 0=working

docvis number of visits to doctor during year (0-121)

hospviz number of days in hospital during year (0-51)

edlevel educational level (categorical: 1-4)

age age: 25-64

female female=1; 0=male

married married=1; 0=not married

kids have children=1; no children=0

hhninc household yearly income in marks (in Marks)

educ years of formal education (7-18)

self self-employed=1; not self employed=0

edlevel1 (1/0) not high school graduate

edlevel2 (1/0) high school graduate

edlevel3 (1/0) university/college

edlevel4 (1/0) graduate school

Details

rwm1984 is saved as a data frame. The data is typically used to model docvis, which is a count variable. It also may be used to model "outwork", a variable indicating if a patient is out-of-work. "outwork" is a binary variable which can be used as the response of a logistic or other binary response model.

Source

German Health Reform Registry for the year 1984, in Hilbe and Greene (2007)

References

- Hardin & Hilbe (2013), Generalized Linear Models & Extensions, 3rd ed, Stata Press.
- Hilbe, Joseph M (2009), Logistic Regression Models, Chapman & Hall/CRC.
- Hilbe, Joseph M (2011), Negative Binomial Regression, 2nd ed., Cambridge University Press.
- Hilbe, Joseph M (2014), Modeling Count Data, Cambridge University Press.
- Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.

Examples

```
# library(MASS)  if not automatically loaded
library(LOGIT)
# library(COUNT)  rwm1984 also in COUNT package, but not toOR or P_disp
data(rwm1984)
# center both docvis and age
rwm1984$cage <- rwm1984$age - mean(rwm1984$age)
rwm1984$cdoc <- rwm1984$docvis - mean(rwm1984$docvis)
glmpr <- glm(outwork ~ cdoc + female + kids + cage + factor(edlevel),
             family=binomial, data=rwm1984)
summary(glmpr)
exp(coef(glmpr))
toOR(glmpr)
```

titanicgrp

titanicgrp

Description

The data is an grouped version of the 1912 Titanic passenger survival log,

Usage

```
data(titanicgrp)
```

Format

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

survive number of passengers who survived

cases number of passengers with same pattern of covariates

age 1=adult; 0=child

sex 1=male; 0=female

class ticket class 1= 1st class; 2= second class; 3= third class

Details

titanicgrp is saved as a data frame. Used to assess risk ratios

Source

Found in many other texts

References

Hilbe, Joseph M (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.
 Hilbe, Joseph M (2014), Modeling Count Data, Cambridge University Press.
 Hilbe, Joseph M (2007, 2011), Negative Binomial Regression, Cambridge University Press.
 Hilbe, Joseph M (2009), Logistic Regression Models, Chapman & Hall/CRC.

Examples

```
library(MASS) # if not automatically loaded

# LOGISTIC REGRESSION
library(LOGIT)
data(titanicgrp)
tg <- titanicgrp
head(tg)
tg$died <- tg$cases - tg$survive
summary(mylr <- glm( cbind(survive, died) ~ age + sex + factor(class),
                    family=binomial, data=tg))

toOR(mylr)
P__disp(mylr)

# SCALED LOGISTIC REGRESSION
summary(myqr <- glm( cbind(survive, died) ~ age + sex + factor(class),
                    family=quasibinomial, data=tg))

toOR(myqr)

# POISSON REGRESSION
# library(COUNT)
data(titanicgrp)
titanicgrp$class <- as.factor(titanicgrp$class)
titanicgrp$logcases <- log(titanicgrp$cases)
glmpr <- glm(survive ~ age + sex + class + offset(logcases), family= poisson, data=titanicgrp)
summary(glmpr)
exp(coef(glmpr))

#lcases <- log(titanicgrp$cases)
#nb2o <- nbinomial(survive ~ age + sex + factor(class),
#                  formula2 =~ age + sex,
#                  offset = lcases,
#                  mean.link="log",
#                  scale.link="log_s",
#                  data=titanicgrp)
```

```
#summary(nb2o)
#exp(coef(nb2o))
```

toOR	<i>Display logistic coefficient table as odds ratios and associated statistics.</i>
------	---

Description

Following the glm command, toOR() displays a table of odds ratios and related statistics including exponentiated model confidence intervals.

Usage

```
toOR(object)
```

Arguments

object name of the fitted glm function model

Format

object The only argument is the name of the fitted glm function model value

or odds ratio of predictor

delta Model standard error using delta method

zscore z-statistic

pvalue probability-value based on normal distribution

exp.loci Exponentiated lower model confidence interval

exp.upci Exponentiated upper model confidence interval

Details

toOR is a post-estimation function, following the use of glm().

Value

list

Note

toOR must be loaded into memory in order to be effective. As a function in LOGIT, it is immediately available to a user.

Author(s)

Joseph M. Hilbe, Arizona State University, and Jet Propulsion Laboratory, California Institute of technology

References

Hilbe, Joseph M. (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.

See Also

[glm](#)

Examples

```
library(MASS)
library(LOGIT)
data(medpar)
mylogit <- glm(died ~ los + white + hmo, family=binomial, data=medpar)
summary(mylogit)
toOR(mylogit)
```

toRR	<i>Display count model coefficient table as incidence rate ratios and associated statistics.</i>
------	--

Description

Following the `glm` or `glm.nb` commands, `toRR()` displays a table of incidence rate ratios and related statistics including exponentiated model confidence intervals.

Usage

```
toRR(object)
```

Arguments

`object` name of the fitted `glm` function model

Format

object The only argument is the name of the fitted `glm` or `glm.nb` function model

or incidence rate ratio of predictor

delta Model standard error using delta method

zscore z-statistic

pvalue probability-value based on normal distribution

exp.loci Exponentiated lower model confidence interval

exp.upci Exponentiated upper model confidence interval

Details

`toRR` is a post-estimation function, following the use of `glm()` with the Poisson or `negative.binomial` families, and following `glm.nb()`.

Value

list

Note

toRR must be loaded into memory in order to be effective. As a function in LOGIT, it is immediately available to a user.

Author(s)

Joseph M. Hilbe, Arizona State University, and Jet Propulsion Laboratory, California Institute of technology

References

Hilbe, Joseph M. (2015), Practical Guide to Logistic Regression, Chapman & Hall/CRC.

Hilbe, Joseph M. (2014), Modeling Count Data, Cambridge University Press.

See Also

[glm](#), [glm.nb](#)

Examples

```
library(MASS)
library(LOGIT)
data(medpar)
medpar$los<-as.numeric(medpar$los)
mypoi <- glm(los ~ white + hmo + factor(age80), family=poisson, data=medpar)
summary(mypoi)
toRR(mypoi)
```


Index

*Topic **datasets**

azheart, [2](#)
badhealth, [3](#)
edrelig, [5](#)
hiv, [6](#)
medpar, [12](#)
mylgg, [14](#)
R84, [16](#)
rwm1984, [19](#)
titanicgrp, [20](#)

*Topic **models**

confusion_stat, [4](#)
h1GOF.test, [7](#)
HLTest, [9](#)
iQuantile, [10](#)
jhbayes, [11](#)
P__disp, [15](#)
ROCTest, [17](#)
toOR, [22](#)
toRR, [23](#)

azheart, [2](#)

badhealth, [3](#)

confusion_stat, [4](#)

edrelig, [5](#)

glm, [5](#), [8](#), [10](#), [16](#), [18](#), [23](#), [24](#)

glm.nb, [24](#)

hiv, [6](#)

h1GOF.test, [7](#)

HLTest, [9](#)

iQuantile, [10](#)

jhbayes, [11](#)

medpar, [12](#)

mylgg, [14](#)

P__disp, [15](#)

R84, [16](#)

ROCTest, [17](#)

rwm1984, [19](#)

titanicgrp, [20](#)

toOR, [22](#)

toRR, [23](#)