

Package ‘msme’

July 23, 2025

Type Package

Title Functions and Datasets for “Methods of Statistical Model Estimation”

Version 0.5.4

Date 2025-04-27

Maintainer Andrew Robinson <apro@unimelb.edu.au>

Depends R (>= 2.10), MASS, lattice

Description Functions and datasets from Hilbe, J.M., and Robinson, A.P. 2013. Methods of Statistical Model Estimation. Chapman & Hall / CRC.

License GPL-3

LazyLoad yes

NeedsCompilation no

Author Joseph Hilbe [aut],
Andrew Robinson [aut, cre]

Repository CRAN

Date/Publication 2025-04-28 10:00:15 UTC

Contents

alrt	2
doll	3
hatvalues.msme	4
heart	5
irls	6
medpar	8
ml_g	10
ml_glm	11
ml_glm2	13
ml_glm3	15
nbinomial	17
plot.ml_g_fit	20
P__disp	21

residuals.msme	22
rwm5yr	23
summary.ml_g_fit	24
summary.msme	26
titanic	27
ufc	28

Index	30
--------------	-----------

alrt	<i>Function to compute asymptotic likelihood ratio test of two models.</i>
------	--

Description

This function computes the asymptotic likelihood ratio test of two models by comparing twice the difference in the log-likelihoods of the models with the Chi-squared distribution with degrees of freedom equal to the difference in the degrees of freedom of the models.

Usage

```
alrt(x1, x2, boundary = FALSE)
```

Arguments

x1	A fitted model as an object that logLik will work for.
x2	A fitted model as an object that logLik will work for.
boundary	A flag that reports whether a boundary correction should be made.

Value

out.tab	A data frame that summarizes the test.
jll.diff	The difference between the log-likelihoods.
df.diff	The difference between the degrees of freedom.
p.value	The p-value of the statistical test of the null hypothesis that there is no difference between the fit of the models.

Note

The function does not provide any checks for nesting, data equivalence, etc.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. *Methods of Statistical Model Estimation*. Chapman & Hall / CRC.

See Also[ml_glm](#), [ml_glm2](#)**Examples**

```
data(medpar)

ml.poi.1 <- ml_glm(los ~ hmo + white,
                  family = "poisson",
                  link = "log",
                  data = medpar)

ml.poi.2 <- ml_glm(los ~ hmo,
                  family = "poisson",
                  link = "log",
                  data = medpar)

alrt(ml.poi.1, ml.poi.2)
```

doll

Physician smoking and mortality count data

Description

The data are a record of physician smoking habits and the frequency of death by myocardial infarction, or heart attack.

Usage

```
data(doll)
```

Format

A data frame with 10 observations on the following variables.

age Ordinal age group

smokes smoking status

deaths count of deaths in category

pyears number of physician years in scope of data

a1 Dummy variable for age level 1

a2 Dummy variable for age level 2

a3 Dummy variable for age level 3

a4 Dummy variable for age level 4

a5 Dummy variable for age level 5

Details

The physicians were divided into five age divisions, with deaths as the response, person years (pyears) as the binomial denominator, and both smoking behavior (smokes) and agegroup (a1–a5) as predictors.

Source

Doll, R and A.B.Hill (1966). Mortality of British doctors in relation to smoking; observations on coronary thrombosis. In *Epidemiological Approaches to the Study of Cancer and Other Chronic Diseases*, W. Haenszel (ed), 19: 204–268. National Cancer Institute Monograph.

References

Hilbe, J., and A.P. Robinson. 2012. *Methods of Statistical Model Estimation*. Chapman & Hall / CRC.

Examples

```
data(doll)

i.glog <- irls(deaths ~ smokes + ordered(age),
              family = "binomial",
              link = "logit",
              data = doll,
              m = doll$pyears)
summary(i.glog)

glm.glog <- glm(cbind(deaths, pyears - deaths) ~
                smokes + ordered(age),
                data = doll,
                family = binomial)
coef(summary(glm.glog))
```

hatvalues.msme

Function to return the hat matrix of a msme-class model.

Description

This function uses QR decomposition to determine the hat matrix of a model given its design matrix X. It is specific to objects of class msme.

Usage

```
## S3 method for class 'msme'
hatvalues(model, ...)
```

Arguments

model A fitted model of class msme.
... other arguments, retained for compatibility with generic method.

Value

An $n \times n$ matrix of hat values, where n is the number of observations used to fit the model. Needed to standardize the residuals.

Note

Leverages can be obtained as the diagonal of the output. See the examples.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. *Methods of Statistical Model Estimation*. Chapman & Hall / CRC.

See Also

[hatvalues](#)

Examples

```
data(medpar)

ml.poi <- ml_glm(los ~ hmo + white,
                family = "poisson",
                link = "log",
                data = medpar)

str(diag(hatvalues(ml.poi)))
```

heart

Heart surgery outcomes for Canadian patients

Description

The data consists of Canadian patients who have either a Coronary Artery Bypass Graft surgery (CABG) or Percutaneous Transluminal Coronary Angioplasty (PTCA) heart procedure.

Usage

```
data(heart)
```

Format

A grouped binomial data frame with 15 observations.

death number of patients that died within 48 hours of hospital admission

cases number of patients monitored

anterior 1: anterior site damage heart attack; 0: other site damage

hcabg 1: previous CABG procedure; 0: previous PTCA procedure;

killip 1: normal heart; 2: angina; 3: minor heart blockage; 4: heart attack or myocardial infarction;

Details

The data are presented as a grouped binomial dataset, with each row representing a different combination of the predictor variables.

Source

National Canadian Registry of Cardiovascular Disease

References

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

```
data(heart)

heart.nb <- irls(death ~ anterior + hcabg + factor(killip),
               a = 0.0001,
               offset = log(heart$cases),
               family = "negBinomial", link = "log",
               data = heart)
```

 irls

Function to fit generalized linear models using IRLS.

Description

This function fits a wide range of generalized linear models using the iteratively reweighted least squares algorithm. The intended benefit of this function is for teaching. Its scope is similar to that of R's glm function, which should be preferred for operational use.

Usage

```
irls(formula, data, family, link, tol = 1e-06, offset = 0, m = 1, a = 1, verbose = 0)
```

Arguments

formula	an object of class <code>"formula"</code> (or one that can be coerced to that class): a symbolic description of the model to be fitted. (See the help for <code>'glm'</code> for more details).
data	a data frame containing the variables in the model.
family	a description of the error distribution to be used in the model. This must be a character string naming a family.
link	a description of the link function to be used in the model. This must be a character string naming a link function.
tol	an optional quantity to use as the convergence criterion for the change in deviance.
offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be 0 or a numeric vector of length equal to the number of cases.
m	the number of cases per observation for binomial regression.
a	the scale for negative binomial regression.
verbose	a flag to control the amount of output printed by the function.

Details

The containing package, `msme`, provides the needed functions to use the `irls` function to fit the Poisson, negative binomial (2), Bernoulli, and binomial families, and supports the use of the identity, log, logit, probit, complementary log-log, inverse, inverse^2 , and negative binomial link functions. All statistics are computed at the final iteration of the IRLS algorithm. The convergence criterion is the magnitude of the change in deviance. The object returned by the function is designed to be reported by the `print.glm` function.

Value

<code>coefficients</code>	parameter estimates.
<code>se.beta.hat</code>	standard errors of parameter estimates.
<code>model</code>	the final, weighted linear model.
<code>call</code>	the function call used to create the object.
<code>nobs</code>	the number of observations.
<code>eta</code>	the linear predictor at the final iteration.
<code>mu</code>	the estimated mean at the final iteration.
<code>df.residual</code>	the residual degrees of freedom.
<code>df.null</code>	the degrees of freedom for the null model.
<code>deviance</code>	the residual deviance.
<code>null.deviance</code>	a place-holder for the null deviance - returned as NA
<code>p.dispersion</code>	Pearson's Chi-squared statistic.
<code>pearson</code>	Pearson's deviance.

loglik	the maximized log-likelihood.
family	the chosen family.
X	the design matrix.
i	the number of iterations required for convergence.
residuals	the deviance residuals.
aic	Akaike's Information Criterion.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. *Methods of Statistical Model Estimation*. Chapman & Hall / CRC.

See Also

[glm](#), [ml_glm](#)

Examples

```
data(medpar)

irls.poi <- irls(los ~ hmo + white,
               family = "poisson",
               link = "log",
               data = medpar)
summary(irls.poi)

irls.probit <- irls(died ~ hmo + white,
                  family = "binomial",
                  link = "probit",
                  data = medpar)
summary(irls.probit)
```

medpar	<i>US national Medicare inpatient hospital database for Arizona patients.</i>
--------	---

Description

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

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

Source

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

References

Hilbe, Joseph M (2007, 2011), Negative Binomial Regression, Cambridge University Press
Hilbe, Joseph M (2009), Logistic Regression Models, Chapman & Hall/CRC
Hardin, JW and JM Hilbe (2001, 2007), Generalized Linear Models and Extensions, Stata Press

Examples

```
data(medpar)
glm1 <- glm(los ~ hmo + white + factor(type),
            family = poisson, data = medpar)
summary(glm1)
exp(coef(glm1))

m1.p <- ml_glm(los ~ hmo + white + factor(type),
              family = "poisson",
              link = "log",
              data = medpar)

summary(m1.p)
```

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

ml_g

Function to fit linear regression using maximum likelihood.

Description

This function demonstrates the use of maximum likelihood to fit ordinary least-squares regression models, by maximizing the likelihood as a function of the parameters. Only conditional normal errors are supported.

Usage

```
ml_g(formula, 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. (See the help for 'lm' for more details).
data	a data frame containing the variables in the model.

Details

This function has limited functionality compared with R's internal lm function, which should be preferred in general.

Value

fit	the output of optim.
X	the design matrix.
y	the response variable.
call	the call used for the function.
beta.hat	the parameter estimates.
se.beta.hat	estimated standard errors of the parameter estimates.
sigma.hat	the estimated conditional standard deviation of the response variable.

Note

We use least squares to get initial estimates, which is a pretty barbaric hack. But the purpose of this function is as a starting point, not to replace existing functions.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. Methods of Statistical Model Estimation. Chapman & Hall / CRC.

See Also

[lm](#)

Examples

```
data(ufc)
ufc <- na.omit(ufc)

ufc.g.reg <- ml_g(height.m ~ dbh.cm, data = ufc)

summary(ufc.g.reg)
```

ml_glm

A function to fit generalized linear models using maximum likelihood.

Description

This function fits generalized linear models by maximizing the joint log-likelihood, which is set in a separate function. Only single-parameter members of the exponential family are covered. The post-estimation output is designed to work with existing reporting functions.

Usage

```
ml_glm(formula, data, family, link, offset = 0, start = NULL, verbose =
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. (See the help for 'glm' for more details).
data	a data frame containing the variables in the model.
family	a description of the error distribution to be used in the model. This must be a character string naming a family.
link	a description of the link function to be used in the model. This must be a character string naming a link function.

offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be 0 or a numeric vector of length equal to the number of cases.
start	optional starting points for the parameter estimation.
verbose	logical flag affecting the detail of printing. Defaults to FALSE.
...	optional arguments to pass within the function.

Details

The containing package, `msme`, provides the needed functions to use the `ml_glm` function to fit the Poisson and Bernoulli families, and supports the use of the identity, log, logit, probit, and complementary log-log link functions. The object returned by the function is designed to be reported by the `print.glm` function.

Value

<code>fit</code>	the output of <code>optim</code> .
<code>X</code>	the design matrix.
<code>y</code>	the response variable.
<code>call</code>	the call used for the function.
<code>obs</code>	the number of observations.
<code>df.null</code>	the degrees of freedom for the null model.
<code>df.residual</code>	the residual degrees of freedom.
<code>deviance</code>	the residual deviance.
<code>null.deviance</code>	the residual deviance for the null model.
<code>residuals</code>	the deviance residuals.
<code>coefficients</code>	parameter estimates.
<code>se.beta.hat</code>	standard errors of parameter estimates.
<code>aic</code>	Akaike's Information Criterion.
<code>i</code>	the number of iterations required for convergence.

Note

This function is neither as comprehensive nor as stable as the inbuilt `glm` function. It is a lot easier to read, however.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. *Methods of Statistical Model Estimation*. Chapman & Hall / CRC.

See Also

[irls](#), [glm](#), [ml_glm2](#)

Examples

```
data(medpar)

ml.poi <- ml_glm(los ~ hmo + white,
                family = "poisson",
                link = "log",
                data = medpar)

ml.poi
summary(ml.poi)
```

ml_glm2

A function to fit generalized linear models using maximum likelihood.

Description

This function fits generalized linear models by maximizing the joint log-likelihood, which is set in a separate function. Two-parameter members of the exponential family are covered. The post-estimation output is designed to work with existing reporting functions.

Usage

```
ml_glm2(formula1, formula2 = ~1, data, family, mean.link, scale.link,
        offset = 0, start = NULL, verbose = FALSE)
```

Arguments

formula1	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the mean function for the model to be fitted. (See the help for 'glm' for more details).
formula2	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the scale function for the model to be fitted. (See the help for 'glm' for more details).
data	a data frame containing the variables in the model.
family	a description of the error distribution to be used in the model. This must be a character string naming a family.
mean.link	a description of the link function to be used for the mean in the model. This must be a character string naming a link function.
scale.link	a description of the link function to be used for the scale in the model. This must be a character string naming a link function.

offset	this can be used to specify an <i>_a priori_</i> known component to be included in the linear predictor during fitting. This should be 0 or a numeric vector of length equal to the number of cases.
start	optional starting points for the parameter estimation.
verbose	logical flag affecting the detail of printing. Defaults to FALSE.

Details

The containing package, *msme*, provides the needed functions to use the `ml_glm2` function to fit the normal and negative binomial (2), families, and supports the use of the identity and log link functions.

The object returned by the function is designed to be reported by the `print.glm` function.

Value

<code>fit</code>	the output of <code>optim</code> .
<code>loglike</code>	the maximized log-likelihood.
<code>X</code>	the design matrix.
<code>y</code>	the response variable.
<code>p</code>	the number of parameters estimated.
<code>rank</code>	the rank of the design matrix for the mean function.
<code>call</code>	the call used for the function.
<code>obs</code>	the number of observations.
<code>fitted.values</code>	estimated response variable.
<code>linear.predictor</code>	linear predictor.
<code>df.null</code>	the degrees of freedom for the null model.
<code>df.residual</code>	the residual degrees of freedom.
<code>pearson</code>	the Pearson Chi2.
<code>null.pearson</code>	the Pearson Chi2 for the null model.
<code>dispersion</code>	the dispersion.
<code>deviance</code>	the residual deviance.
<code>null.deviance</code>	the residual deviance for the null model.
<code>residuals</code>	the deviance residuals.
<code>presiduals</code>	the Pearson residuals.
<code>coefficients</code>	parameter estimates.
<code>se.beta.hat</code>	standard errors of parameter estimates.
<code>aic</code>	Akaike's Information Criterion.
<code>offset</code>	the offset used.
<code>i</code>	the number of iterations required for convergence.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. Methods of Statistical Model Estimation. Chapman & Hall / CRC.

See Also

[glm](#), [irls](#), [ml_glm](#),

Examples

```
data(medpar)
ml.nb2 <- ml_glm2(los ~ hmo + white,
                 formula2 = ~1,
                 data = medpar,
                 family = "negBinomial",
                 mean.link = "log",
                 scale.link = "inverse_s")
```

```
data(ufc)
```

```
ufc <- na.omit(ufc)
ml.g <- ml_glm2(height.m ~ dbh.cm,
                formula2 = ~ dbh.cm,
                data = ufc,
                family = "normal",
                mean.link = "identity",
                scale.link = "log_s")
```

```
summary(ml.g)
```

ml_glm3

A reduced maximum likelihood fitting function that omits null models.

Description

This function fits generalized linear models by maximizing the joint log-likelihood, which is set in a separate function. Null models are omitted from the fit. The post-estimation output is designed to work with existing reporting functions.

Usage

```
ml_glm3(formula, data, family, link, offset = 0, start = NULL, verbose = FALSE, ...)
```

Arguments

formula	an object of class <code>"formula"</code> (or one that can be coerced to that class): a symbolic description of the model to be fitted. (See the help for <code>'glm'</code> for more details).
data	a data frame containing the variables in the model.
family	a description of the error distribution to be used in the model. This must be a character string naming a family.
link	a description of the link function to be used in the model. This must be a character string naming a link function.
offset	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be 0 or a numeric vector of length equal to the number of cases.
start	optional starting points for the parameter estimation.
verbose	logical flag affecting the detail of printing. Defaults to <code>FALSE</code> .
...	other arguments to pass to the likelihood function, e.g. group structure.

Details

This function is essentially the same as `ml_glm`, but includes the `dots` argument to allow a richer set of model likelihoods to be fit, and omits computation of the null deviance. The function is presently set up to only fit the conditional fixed-effects negative binomial model.

Value

<code>fit</code>	the output of <code>optim</code> .
<code>X</code>	the design matrix.
<code>y</code>	the response variable.
<code>call</code>	the call used for the function.
<code>obs</code>	the number of observations.
<code>df.null</code>	the degrees of freedom for the null model.
<code>df.residual</code>	the residual degrees of freedom.
<code>deviance</code>	the residual deviance.
<code>null.deviance</code>	the residual deviance for the null model, set to <code>NA</code> .
<code>residuals</code>	the deviance residuals.
<code>coefficients</code>	parameter estimates.
<code>se.beta.hat</code>	standard errors of parameter estimates.
<code>aic</code>	Akaike's Information Criterion.
<code>i</code>	the number of iterations required for convergence.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. Methods of Statistical Model Estimation. Chapman & Hall / CRC.

See Also

[irls](#), [glm](#), [ml_glm](#)

Examples

```
data(medpar)
med.nb.g <- ml_glm3(los ~ hmo + white,
  family = "gNegBinomial",
  link = "log",
  group = medpar$provnum,
  data = medpar)
summary(med.nb.g)
```

nbinomial	<i>A function to fit negative binomial generalized linear models using maximum likelihood.</i>
-----------	--

Description

This function fits generalized linear models by maximizing the joint log-likelihood, which is set in a separate function. Two-parameter members of the negative binomial family are covered. The post-estimation output is designed to work with existing reporting functions.

Usage

```
nbinomial(formula1, formula2 = ~1, data, family="nb2", mean.link="log",
  scale.link="inverse_s", offset=0, start=NULL, verbose=FALSE)
```

Arguments

formula1	an object of class <code>"formula"</code> (or one that can be coerced to that class): a symbolic description of the mean function for the model to be fitted. (See the help for <code>'glm'</code> for more details).
formula2	an object of class <code>"formula"</code> (or one that can be coerced to that class): a symbolic description of the scale function for the model to be fitted. (See the help for <code>'glm'</code> for more details).
data	a data frame containing the variables in the model.
family	a description of the error distribution to be used in the model. This must be a character string naming a family.
mean.link	a description of the link function to be used for the mean in the model. This must be a character string naming a link function.

<code>scale.link</code>	a description of the link function be used for the scale in the model. This must be a character string naming a link function.
<code>offset</code>	this can be used to specify an <code>_a priori_</code> known component to be included in the linear predictor during fitting. This should be 0 or a numeric vector of length equal to the number of cases.
<code>start</code>	optional starting points for the parameter estimation.
<code>verbose</code>	logical flag affecting the detail of printing. Defaults to FALSE.

Details

The containing package, `msme`, provides the needed functions to use the `nbinomial` function to fit the negative binomial (2), families, and supports the use of the identity and log link functions.

The object returned by the function is designed to be reported by the `print.glm` function.

Value

<code>fit</code>	the output of <code>optim</code> .
<code>loglike</code>	the maximized log-likelihood.
<code>X</code>	the design matrix.
<code>y</code>	the response variable.
<code>p</code>	the number of parameters estimated.
<code>rank</code>	the rank of the design matrix for the mean function.
<code>call</code>	the call used for the function.
<code>obs</code>	the number of observations.
<code>fitted.values</code>	estimated response variable.
<code>linear.predictor</code>	linear predictor.
<code>df.null</code>	the degrees of freedom for the null model.
<code>df.residual</code>	the residual degrees of freedom.
<code>pearson</code>	the Pearson Chi2.
<code>null.pearson</code>	the Pearson Chi2 for the null model.
<code>dispersion</code>	the dispersion.
<code>deviance</code>	the residual deviance.
<code>null.deviance</code>	the residual deviance for the null model.
<code>residuals</code>	the deviance residuals.
<code>presiduals</code>	the Pearson residuals.
<code>coefficients</code>	parameter estimates.
<code>se.beta.hat</code>	standard errors of parameter estimates.
<code>aic</code>	Akaike's Information Criterion.
<code>offset</code>	the offset used.
<code>i</code>	the number of iterations required for convergence.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. *Methods of Statistical Model Estimation*. Chapman & Hall / CRC.

See Also

[glm](#), [irls](#), [ml_glm2](#),

Examples

```
data(medpar)

# TRADITIONAL NB REGRESSION WITH ALPHA

mynb1 <- nbinomial(los ~ hmo + white, data=medpar)
summary(mynb1)

# TRADITIONAL NB -- SHOWING ALL OPTIONS

mynb2 <- nbinomial(los ~ hmo + white,
                  formula2 = ~ 1,
                  data = medpar,
                  family = "nb2",
                  mean.link = "log",
                  scale.link = "inverse_s")
summary(mynb2)

# R GLM.NB - LIKE INVERTED DISPERSION BASED M

mynb3 <- nbinomial(los ~ hmo + white,
                  formula2 = ~ 1,
                  data = medpar,
                  family = "negBinomial",
                  mean.link = "log",
                  scale.link = "inverse_s")
summary(mynb3)

# R GLM.NB-TYPE INVERTED DISPERSION --THETA ; WITH DEFAULTS

mynb4 <- nbinomial(los ~ hmo + white, family="negBinomial", data =medpar)
summary(mynb4)

# HETEROGENEOUS NB; DISPERSION PARAMETERIZED

mynb5 <- nbinomial(los ~ hmo + white,
                  formula2 = ~ hmo + white,
                  data = medpar,
                  family = "negBinomial",
```

```
summary(mynb5)
      mean.link = "log",
      scale.link = "log_s")
```

plot.ml_g_fit

A plot method for objects of class ml_g_fit.

Description

This function provides a four-way plot for fitted models.

Usage

```
## S3 method for class 'ml_g_fit'
plot(x, ...)
```

Arguments

x the fitted model.
... other arguments, retained for compatibility with generic method.

Details

The function plots a summary. The output is structured to broadly match the default options of the plot.lm function.

Value

Run for its side effect of producing a plot object.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. Methods of Statistical Model Estimation. Chapman & Hall / CRC.

See Also

[ml_g](#)

Examples

```
data(ufc)
ufc <- na.omit(ufc)

ufc.g.reg <- ml_g(height.m ~ dbh.cm, data = ufc)

plot(ufc.g.reg)
```

P__disp	<i>A function to calculate Pearson Chi2 and its dispersion statistic following glm and glm.nb.</i>
---------	--

Description

This function calculates Pearson Chi2 statistic and the Pearson-based dispersion statistic. Values of the dispersion greater than 1 indicate model overdispersion. Values under 1 indicate underdispersion.

Usage

```
P__disp(x)
```

Arguments

x the fitted model.

Details

To be used following glm and glm.nb functions.

Value

pearson.chi2	Pearson Chi2 value.
dispersion	Pearson-based dispersion.

Author(s)

Joseph Hilbe and Andrew Robinson

References

Hilbe, J.M., and Robinson, A.P. 2013. Methods of Statistical Model Estimation. Chapman & Hall / CRC.

See Also

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

Examples

```
data(medpar)
mymod <- glm(los ~ hmo + white + factor(type),
            family = poisson,
            data = medpar)
P__disp(mymod)
```

residuals.msme

Function to produce residuals from a model of class msme.

Description

Function to produce deviance and standardized deviance residuals from a model of class msme.

Usage

```
## S3 method for class 'msme'
residuals(object, type = c("deviance", "standard"), ...)
```

Arguments

object	a model of class msme.
type	the type of residual requested. Defaults to deviance.
...	arguments to pass on. Retained for compatibility with generic method.

Details

Presently only deviance or standardized deviance residuals are computed.

Value

A vector of residuals.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. *Methods of Statistical Model Estimation*. Chapman & Hall / CRC.

Examples

```
data(medpar)

ml.poi <- ml_glm(los ~ hmo + white,
                family = "poisson",
                link = "log",
                data = medpar)

str(residuals(ml.poi))
```

rwm5yr

*German health registry for the years 1984-1988***Description**

German health registry for the years 1984-1988. Health information for years immediately prior to health reform.

Usage

```
data(rwm5yr)
```

Format

A data frame with 19,609 observations on the following 17 variables.

id patient ID (1=7028)
docvis number of visits to doctor during year (0-121)
hospvis number of days in hospital during year (0-51)
year year; (categorical: 1984, 1985, 1986, 1987, 1988)
edlevel educational level (categorical: 1-4)
age age: 25-64
outwork out of work=1; 0=working
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

rwm5yr is saved as a data frame. Count models typically use docvis as response variable. 0 counts are included

Source

German Health Reform Registry, years pre-reform 1984-1988,

References

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

Examples

```
data(rwm5yr)

glmrp <- glm(docvis ~ outwork + female + age + factor(edlevel),
             family = poisson, data = rwm5yr)
summary(glmrp)
exp(coef(glmrp))

ml_p <- ml_glm(docvis ~ outwork + female + age + factor(edlevel),
              family = "poisson",
              link = "log",
              data = rwm5yr)
summary(ml_p)
exp(coef(ml_p))

library(MASS)
glmrnb <- glm.nb(docvis ~ outwork + female + age + factor(edlevel),
                data = rwm5yr)
summary(glmrnb)
exp(coef(glmrnb))
## Not run:
library(gee)
mygee <- gee(docvis ~ outwork + age + factor(edlevel), id=id,
            corstr = "exchangeable", family=poisson, data=rwm5yr)
summary(mygee)
exp(coef(mygee))

## End(Not run)
```

summary.ml_g_fit

A summary method for objects of class ml_g_fit.

Description

This function provides a compact summary for fitted models.

Usage

```
## S3 method for class 'ml_g_fit'  
summary(object, dig = 3, ...)
```

Arguments

object	the fitted model.
dig	an optional integer detailing the number of significant digits for printing.
...	other arguments, retained for compatibility with generic method.

Details

The function prints out a summary and returns an invisible list with useful objects. The output is structured to match the print.summary.lm function.

Value

call	the call used to fit the model.
coefficients	a dataframe of estimates, standard errors, etc.
residuals	deviance residuals from the model.
aliased	included to match the print.summary.lm function. Lazily set to FALSE for all parameters.
sigma	the estimate of the conditional standard deviation of the response variable.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. Methods of Statistical Model Estimation. Chapman & Hall / CRC.

See Also

[ml_g](#)

Examples

```
data(ufc)  
ufc <- na.omit(ufc)  
  
ufc.g.reg <- ml_g(height.m ~ dbh.cm, data = ufc)  
  
summary(ufc.g.reg)
```

summary.msme	<i>A summary method for objects of class msme.</i>
--------------	--

Description

This function provides a compact summary for fitted models.

Usage

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

Arguments

object	the fitted model.
...	optional arguments to be passed through.

Details

The function prints out a summary and returns an invisible list with useful objects.

Value

call	the call used to fit the model.
coefficients	a dataframe of estimates, standard errors, etc.
deviance	deviance from the model fit.
null.deviance	deviance from the null model fit.
df.residual	residual degrees of freedom from the model fit.
df.null	residual degrees of freedom from the null model fit.

Author(s)

Andrew Robinson and Joe Hilbe.

References

Hilbe, J.M., and Robinson, A.P. 2013. *Methods of Statistical Model Estimation*. Chapman & Hall / CRC.

Examples

```
data(medpar)

ml.poi <- ml_glm(los ~ hmo + white,
                family = "poisson",
                link = "log",
                data = medpar)

summary(ml.poi)
```

titanic

Titanic passenger survival data

Description

Passenger survival data from 1912 Titanic shipping accident.

Usage

```
data(titanic)
```

Format

A data frame with 1316 observations on the following 4 variables.

survived 1=survived; 0=died

age 1=adult; 0=child

sex 1=Male; 0=female

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

Details

Titanic is saved as a data frame. Used to assess risk ratio; not standard count model; good binary response model.

Source

Found in many other texts

References

Hilbe, Joseph M (2007, 2011), Negative Binomial Regression, Cambridge University Press
Hilbe, Joseph M (2009), Logistic Regression Models, Chapman & Hall/CRC

Examples

```
data(titanic)

glm.lr <- glm(survived ~ age + sex + factor(class),
             family=binomial, data=titanic)
summary(glm.lr)
exp(coef(glm.lr))

glm.irls <- irls(survived ~ age + sex + factor(class),
               family = "binomial",
               link = "cloglog",
               data = titanic)
summary(glm.irls)
exp(coef(glm.irls))

glm.ml <- ml_glm(survived ~ age + sex + factor(class),
                family = "bernoulli",
                link = "cloglog1",
                data = titanic)
summary(glm.ml)
exp(coef(glm.ml))
```

ufc

Upper Flat Creek forest cruise tree data

Description

These are a subset of the tree measurement data from the Upper Flat Creek unit of the University of Idaho Experimental Forest, which was measured in 1991.

Usage

```
data(ufc)
```

Format

A data frame with 336 observations on the following 5 variables.

plot plot label

tree tree label

species species kbd with levels DF, GF, WC, WL

dbh.cm tree diameter at 1.37 m. from the ground, measured in centimetres.

height.m tree height measured in metres

Details

The inventory was based on variable radius plots with 6.43 sq. m. per ha. BAF (Basal Area Factor). The forest stand was 121.5 ha. This version of the data omits errors, trees with missing heights, and uncommon species. The four species are Douglas-fir, grand fir, western red cedar, and western larch.

Source

The data are provided courtesy of Harold Osborne and Ross Appelgren of the University of Idaho Experimental Forest.

References

Robinson, A.P., and J.D. Hamann. 2010. Forest Analytics with R: an Introduction. Springer.

Examples

```
data(ufc)

ufc <- na.omit(ufc)
m1.g <- ml_glm2(height.m ~ dbh.cm,
               formula2 = ~1,
               data = ufc,
               family = "normal",
               mean.link = "identity",
               scale.link = "log_s")

lm.g <- lm(height.m ~ dbh.cm,
           data = ufc)

m1.g
lm.g

summary(m1.g)
summary(lm.g)
```

Index

- * **~models**
 - alrt, 2
 - hatvalues.msme, 4
 - irls, 6
 - ml_g, 10
 - ml_glm, 11
 - ml_glm2, 13
 - ml_glm3, 15
 - nbinomial, 17
 - residuals.msme, 22
 - summary.ml_g_fit, 24
 - summary.msme, 26
- * **datasets**
 - doll, 3
 - heart, 5
 - medpar, 8
 - rwm5yr, 23
 - titanic, 27
 - ufc, 28
- * **htest**
 - plot.ml_g_fit, 20
- * **models**
 - plot.ml_g_fit, 20
- alrt, 2
- doll, 3
- glm, 8, 13, 15, 17, 19, 21
- glm.nb, 21
- hatvalues, 5
- hatvalues.msme, 4
- heart, 5
- irls, 6, 13, 15, 17, 19
- lm, 11
- medpar, 8
- ml_g, 10, 20, 25
- ml_glm, 3, 8, 11, 15, 17
- ml_glm2, 3, 13, 13, 19
- ml_glm3, 15
- nbinomial, 17
- P__disp, 21
- plot.ml_g_fit, 20
- residuals.msme, 22
- rwm5yr, 23
- summary.ml_g_fit, 24
- summary.msme, 26
- titanic, 27
- ufc, 28