Package ‘glmtoolbox’

October 10, 2023

Type   Package
Title  Set of Tools to Data Analysis using Generalized Linear Models
Version 0.1.9
Description Set of tools to the statistical analysis of data using: (1) normal linear models; (2) generalized linear models; (3) negative binomial regression models as alternative to the Poisson regression models under the presence of overdispersion; (4) beta-binomial and random-clumped binomial regression models as alternative to the binomial regression models under the presence of overdispersion; (5) Zero-inflated and zero-altered regression models to deal with zero-excess in count data; (6) generalized estimating equations for cluster correlated data.
License GPL-2 | GPL-3
Imports methods, stats, utils, graphics, numDeriv, Rfast, splines, Formula, MASS, statmod
Suggests aplore3, ISLR, pscl, GLMsData
Encoding UTF-8
LazyData false
RoxygenNote 7.2.3
NeedsCompilation no
Author Luis Hernando Vanegas [aut, cre],
     Luz Marina Rondón [aut],
     Gilberto A. Paula [aut]
Maintainer Luis Hernando Vanegas <lhvanegasp@unal.edu.co>
Repository CRAN
Date/Publication 2023-10-10 02:50:03 UTC

R topics documented:

adjR2 ................................................................. 4
adjR2.glm ............................................................ 4
adjR2.lm .............................................................. 5
advertising ............................................................ 6
AGPC ............................................................... 7
R topics documented:

amenorrhea .......................................................... 9
anova.glmgee ....................................................... 10
anova.overglm ...................................................... 11
anova.zeroinflation ............................................... 13
anova2 .............................................................. 14
aucuba ............................................................... 16
bladder ............................................................... 16
BoxTidwell .......................................................... 17
BoxTidwell(glm) .................................................... 18
BoxTidwell.lm ....................................................... 20
brains ................................................................. 22
cellular .............................................................. 23
cholecystectomy .................................................... 24
CIC ................................................................. 25
confint2 ............................................................. 26
cooks.distance.glmgee ........................................ 28
cooks.distance.overglm ......................................... 30
cooks.distance.zeroinflation ................................... 31
coupons ............................................................. 33
depression ........................................................... 34
dfbeta.glmgee ...................................................... 35
dfbeta.overglm ..................................................... 37
dfbeta.zeroinflation .............................................. 38
dilution ............................................................... 40
dfbeta ............................................................... 40
envelope .............................................................. 40
envelope(glm) ....................................................... 41
envelope.lm .......................................................... 43
envelope.overglm .................................................. 46
envelope.zeroinflation .......................................... 48
estequa ............................................................. 50
estequa(glm) .......................................................... 50
estequa.glmgee ..................................................... 51
estequa.overglm .................................................... 53
estequa.zeroinflation ............................................ 54
fabric ............................................................... 55
FisherScoring ....................................................... 56
GHYC ............................................................... 57
glmgee ............................................................. 58
gnmgee ............................................................. 63
GUIDE .............................................................. 67
gvif ................................................................. 68
gvif(glm) ............................................................. 68
gvif.lm ............................................................... 70
gvif.overglm ........................................................ 71
hltest ............................................................... 72
ldh ................................................................. 74
leverage ............................................................. 75
leverage.glmgee .................................................... 75
R topics documented:

liver ................................................................. 77
localInfluence .................................................. 78
localInfluence.glm ............................................. 78
localInfluence.glmgee .......................................... 79
localInfluence.overglm ....................................... 81
mammary ........................................................... 83
Oranges ............................................................. 84
orobanche .......................................................... 84
ossification .......................................................... 85
overglm ............................................................. 86
PAC ................................................................. 90
pipeline ............................................................. 91
predict.glmgee .................................................. 92
QIC ................................................................. 94
rabbits ............................................................. 95
races ............................................................... 96
residuals.glmgee .................................................. 97
residuals.overglm ............................................... 98
residuals.zeroinflation ....................................... 100
residuals2 .......................................................... 101
richness ............................................................. 102
rinse ............................................................... 103
RJC ................................................................. 104
ROCc ............................................................... 105
Seizures ........................................................... 106
SGPC ............................................................... 107
shelflife ............................................................ 109
skincancer ......................................................... 110
spruces ............................................................. 111
Steel ............................................................... 112
stepCriterion ..................................................... 112
stepCriterion.glm .............................................. 113
stepCriterion.glmgee .......................................... 115
stepCriterion.lm ............................................... 117
stepCriterion.overglm ....................................... 119
swimmers ......................................................... 121
Trajan ............................................................ 123
uti ................................................................. 124
vcov.glmgee ..................................................... 125
vdtest ............................................................. 126
vdtest.glm ....................................................... 127
vdtest.lm .......................................................... 128
wglmgee ........................................................... 130
zero.excess ........................................................ 134
zeroalt ............................................................. 135
zeroinf ............................................................. 139

Index 143
adjR2

Adjusted R-squared

Description
Computes the adjusted R-squared

Usage
adjR2(..., digits, verbose)

Arguments
... one of several model fit objects.
digits an (optional) integer value indicating the number of decimal places to be used.
verbose an (optional) logical indicating if should the report of results be printed.

Value
A matrix with the values of the adjusted R-squared for all model fit objects.

adjR2.glm

Adjusted R-squared in Generalized Linear Models

Description
Computes the adjusted deviance-based R-squared in generalized linear models.

Usage
## S3 method for class 'glm'
adjR2(..., digits = 4, verbose = TRUE)

Arguments
... one or several objects of the class glm, which are obtained from the fit of generalized linear models.
digits an (optional) integer value indicating the number of decimal places to be used. By default, digits is set to be 4.
verbose an (optional) logical indicating if should the report of results be printed. By default, verbose is set to be TRUE.
Details

The deviance-based R-squared is computed as $R^2 = 1 - \frac{Deviance}{Null.Deviance}$. Then, the adjusted deviance-based R-squared is computed as $1 - \frac{n-1}{n-p}(1 - R^2)$, where $p$ is the number of parameters in the linear predictor and $n$ is the sample size.

Value

a matrix with the following columns

| Deviance | value of the residual deviance, |
| R-squared | value of the deviance-based R-squared, |
| df | number of parameters in the linear predictor, |
| adj.R-squared | value of the adjusted deviance-based R-squared, |

Examples

```r
### Example 1: Fuel efficiency of cars
Auto <- ISLR::Auto
fit1 <- glm(mpg ~ horsepower*weight, family=Gamma(inverse), data=Auto)
fit2 <- update(fit1, formula=mpg ~ horsepower*weight*cylinders)
fit3 <- update(fit1, family=Gamma(log))
fit4 <- update(fit2, family=Gamma(log))
fit5 <- update(fit1, family=inverse.gaussian(log))
fit6 <- update(fit2, family=inverse.gaussian(log))

AIC(fit1,fit2,fit3,fit4,fit5,fit6)
BIC(fit1,fit2,fit3,fit4,fit5,fit6)
adjR2(fit1,fit2,fit3,fit4,fit5,fit6)
```

## adjR2.lm

**Adjusted R-squared in Normal Linear Models**

**Description**

Extracts the adjusted R-squared in normal linear models.

**Usage**

```r
# S3 method for class 'lm'
adjR2(..., digits = 4, verbose = TRUE)
```
Arguments

... one or several objects of the class \textit{lm}, which are obtained from the fit of normal linear models.

digits an (optional) integer value indicating the number of decimal places to be used. By default, \texttt{digits} is set to be 4.

verbose an (optional) logical indicating if should the report of results be printed. By default, \texttt{verbose} is set to be \texttt{TRUE}.

Details

The R-squared is computed as \( R^2 = 1 - \frac{RSS}{\text{Null.RSS}} \). Then, the adjusted R-squared is computed as \( 1 - \frac{n-1}{n-p}(1 - R^2) \), where \( p \) is the number of parameters in the linear predictor and \( n \) is the sample size.

Value

a matrix with the following columns

- \textit{RSS} value of the residual sum of squares,
- \textit{R-squared} value of the R-squared,
- \textit{df} number of parameters in the linear predictor,
- \textit{adj.R-squared} value of the adjusted R-squared,

Examples

%%%% Example 1: Fuel efficiency of cars

\begin{verbatim}
fit1 <- lm(mpg ~ log(hp) + log(wt) + qsec, data=mtcars)
fit2 <- lm(mpg ~ log(hp) + log(wt) + qsec + log(hp)*log(wt), data=mtcars)
fit3 <- lm(mpg ~ log(hp)*log(wt)*qsec, data=mtcars)

AIC(fit1,fit2,fit3)
BIC(fit1,fit2,fit3)
adjR2(fit1,fit2,fit3)
\end{verbatim}

Description

The advertising data set consists of sales of that product in 200 different markets. It also includes advertising budgets for the product in each of those markets for three different media: TV, radio, and newspapers.
Usage

```r
data(advertising)
```

Format

A data frame with 200 rows and 4 variables:

- **TV** a numeric vector indicating the advertising budget on TV.
- **radio** a numeric vector indicating the advertising budget on radio.
- **newspaper** a numeric vector indicating the advertising budget on newspaper.
- **sales** a numeric vector indicating the sales of the interest product.

Source

https://www.statlearning.com/s/Advertising.csv

References


Examples

```r
data(advertising)
breaks <- with(advertising,quantile(radio,probs=c(0:3)/3))
labels <- c("low","mid","high")
advertising2 <- within(advertising,radioC <- cut(radio,breaks,labels,include.lowest=TRUE))
dev.new()
with(advertising2,plot(TV,sales,pch=16,col=as.numeric(radioC)))
legend("topleft",legend=c("low","mid","high"),fill=c(1:3),title="Radio",bty="n")
```

---

**AGPC**  
AGPC for Generalized Estimating Equations

Description

Computes the Akaike-type penalized Gaussian pseudo-likelihood criterion (AGPC) for one or more objects of the class glmgee.

Usage

```r
AGPC(..., k = 2, verbose = TRUE, digits = 2)
```
Arguments

... one or several objects of the class glmgee.

k an (optional) non-negative value giving the magnitude of the penalty. By default, k is set to be 2.

verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.

digits an (optional) integer indicating the number of digits to print.

Details

If k is set to be 0 then the AGPC reduces to the Gaussian pseudo-likelihood criterion (GPC), proposed by Carey and Wang (2011), which corresponds to the logarithm of the multivariate normal density function.

Value

A data.frame with the values of the gaussian pseudo-likelihood, the number of parameters in the linear predictor plus the number of parameters in the correlation matrix, and the value of AGPC for each glmgee object in the input.

References


See Also

QIC, CIC, RJC, GHYC, SGPC

Examples

####### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces

data(spruces)

mod1 <- size ~ poly(days,4) + treat

fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces)

fit2 <- update(fit1, corstr="AR-M-dependent")

fit3 <- update(fit1, corstr="Stationary-M-dependent(2)"")

fit4 <- update(fit1, corstr="Exchangeable")

AGPC(fit1, fit2, fit3, fit4)

####### Example 2: Treatment for severe postnatal depression

data(depression)

mod2 <- depressd ~ visit + group

fit1 <- glmgee(mod2, id=subj, family=binomial(logit), data=depression)

fit2 <- update(fit1, corstr="AR-M-dependent")
amenorrhea

fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
AGPC(fit1, fit2, fit3, fit4)

##### Example 3: Treatment for severe postnatal depression (2)
mod3 <- dep ~ visit*group
fit1 <- glmgee(mod3, id=subj, family=gaussian(identity), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Exchangeable")
AGPC(fit1, fit2, fit3)

---

amenorrhea

**Description**

A total of 1151 women completed menstrual diaries. The diary data were used to generate a binary sequence for each woman, indicating whether or not she had experienced amenorrhea (the absence of menstrual bleeding for a specified number of days) on the day of randomization and three additional 90-day intervals. The goal of this trial was to compare the two treatments (100 mg or 150 mg of depot-medroxyprogesterone acetate (DMPA)) in terms of how the rates of amenorrhea change over time with continued use of the contraceptive method.

**Usage**

data(amenorrhea)

**Format**

A data frame with 4604 rows and 4 variables:

- **ID** a numeric vector indicating the woman’s ID.
- **Dose** a factor with two levels: "100mg" for treatment with 100 mg injection; and "150mg" for treatment with 150 mg injection.
- **Time** a numeric vector indicating the number of 90-day intervals since the trial began.
- **amenorrhea** a numeric vector indicating the amenorrhea status (1 for amenorrhea; 0 otherwise).

**References**


Examples

```r
data(amenorrhea)
dev.new()
amenorrhea2 <- aggregate(amenorrhea ~ Time + Dose, mean, data=amenorrhea, na.rm=TRUE)
barplot(100*amenorrhea ~ Dose+Time, data=amenorrhea2, beside=TRUE, col=c("blue","yellow"), ylab="%")
legend("topleft", legend=c("100 mg","150 mg"), fill=c("blue","yellow"), title="Dose", bty="n")
```

### anova.glmege

Comparison of nested Generalized Estimating Equations

Description

Allows to compare nested generalized estimating equations using the Wald and generalized score tests.

Usage

```r
## S3 method for class 'glmgee'
anova(
  object,
  ...
  test = c("wald", "score"),
  verbose = TRUE,
  varest = c("robust", "df-adjusted", "model", "bias-corrected")
)
```

Arguments

- **object**: an object of the class `glmgee`.
- **...**: another objects of the class `glmgee` which are obtained from the fit of generalized estimating equations.
- **test**: an (optional) character string indicating the required test. The available options are: Wald ("wald") and generalized score ("score") tests. By default, `test` is set to be "wald".
- **verbose**: an (optional) logical switch indicating if should the report of results be printed. By default, `verbose` is set to be TRUE.
- **varest**: an (optional) character string indicating the type of estimator which should be used to the variance-covariance matrix of the interest parameters in the Wald test. The available options are: robust sandwich-type estimator ("robust"), degrees-of-freedom-adjusted estimator ("df-adjusted"), bias-corrected estimator ("bias-corrected"), and the model-based or naive estimator ("model"). By default, `varest` is set to be "robust". See `vcov.glmege`.
Value
A matrix with three columns which contains the following:

- **Chi**: The value of the statistic of the test.
- **df**: The number of degrees of freedom.
- **Pr(>Chi)**: The \( p \)-value of the test computed using the Chi-square distribution.

References


Examples

```
###### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod <- size ~ poly(days,4)
fit1 <- glmgee(mod, id=tree, family=Gamma(log), data=spruces, corstr="AR-M-dependent")
fit2 <- update(fit1, . ~ . + treat)
fit3 <- update(fit2, . ~ . + poly(days,4):treat)
anova(fit1,fit2,fit3,test="wald")
anova(fit3,test="wald")
```

```
###### Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ group
fit1 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent", data=depression)
fit2 <- update(fit1, . ~ . + visit)
fit3 <- update(fit2, . ~ . + group:visit)
anova(fit1,fit2,fit3,test="score")
anova(fit3,test="score")
```

Description
Allows to compare nested models for regression models based on the negative binomial, beta-binomial, and random-clumped binomial distributions, which are alternatives to the Poisson and binomial regression models under the presence of overdispersion. The comparisons are performed by using the Wald, score, gradient or likelihood ratio tests.
Usage

## S3 method for class 'overglm'

anova(object, ..., test = c("wald", "lr", "score", "gradient"), verbose = TRUE)

Arguments

object

an object of the class overglm.

...  
another objects of the class overglm.

test

an (optional) character string which allows to specify the required test. The available options are: Wald ("wald"), Rao’s score ("score"), likelihood ratio ("lr") and Terrell’s gradient ("gradient") tests. By default, test is set to be "wald".

verbose

an (optional) logical indicating if should the report of results be printed. By default, verbose is set to be TRUE.

Value

A matrix with the following three columns:

Chi  
The value of the statistic of the test,

Df   
The number of degrees of freedom,

Pr(>Chi)  
The p-value of the test-type test computed using the Chi-square distribution.

References


Examples

## Example 1: Self diagnosed ear infections in swimmers
data(swimmers)
fit1 <- over glm(infections ~ frequency + location + age + gender, family="nb1(log)", data=swimmers)
anova(fit1, test="wald")
anova(fit1, test="score")
anova(fit1, test="lr")
anova(fit1, test="gradient")

## Example 2: Agents to stimulate cellular differentiation
data(cellular)
fit2 <- over glm(cbind(cells,200-cells) ~ tnf*ifn, family="bb(logit)", data=cellular)
anova(fit2, test="wald")
anova(fit2, test="score")
anova(fit2, test="lr")
anova(fit2, test="gradient")
anova.zeroinflation

Description

Allows to compare nested models for regression models used to deal with zero-excess in count data. The comparisons are performed by using the Wald, score, gradient or likelihood ratio tests.

Usage

```r
## S3 method for class 'zeroinflation'
anova(
  object,
  ..., 
  test = c("wald", "lr", "score", "gradient"),
  verbose = TRUE,
  submodel = c("counts", "zeros")
)
```

Arguments

- `object` an object of the class `zeroinflation`.
- `...` another objects of the class `zeroinflation`.
- `test` an (optional) character string which allows to specify the required test. The available options are: Wald ("wald"), Rao’s score ("score"), likelihood ratio ("lr") and Terrell’s gradient ("gradient") tests. By default, test is set to be "wald".
- `verbose` an (optional) logical indicating if should the report of results be printed. By default, verbose is set to be TRUE.
- `submodel` an (optional) character string which allows to specify the model: "counts" or "zeros". By default, submodel is set to be "counts".

Value

A matrix with the following three columns:

- Chi: The value of the statistic of the test,
- Df: The number of degrees of freedom,
- Pr(>Chi): The p-value of the test `test` computed using the Chi-square distribution.

References


Examples

####### Example 1: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit1 <- zeroinf(art ~ fem + kid5 + ment | ment, family="nbl1(log)", data = bioChemists)
anova(fit1,test="wald")
anova(fit1,test="lr")
anova(fit1,test="score")
anova(fit1,test="gradient")

fit2 <- zeroalt(art ~ fem + kid5 + ment, family="nbl1(log)", data = bioChemists)
anova(fit2,submodel="zeros",test="wald")
anova(fit2,submodel="zeros",test="lr")
anova(fit2,submodel="zeros",test="score")
anova(fit2,submodel="zeros",test="gradient")

---

anova2

Comparison of nested Generalized Linear Models

Description

Allows to compare nested generalized linear models using Wald, score, gradient, and likelihood ratio tests.

Usage

anova2(
  object,
  ..., 
  test = c("wald", "lr", "score", "gradient"),
  verbose = TRUE 
)

Arguments

object an object of the class glm which is obtained from the fit of a generalized linear model.

... another objects of the class glm which are obtained from the fit of generalized linear models.

test an (optional) character string indicating the required type of test. The available options are: Wald ("wald"), Rao’s score ("score"), Terrell’s gradient ("gradient"), and likelihood ratio ("lr") tests. By default, test is set to be "wald".

verbose an (optional) logical indicating if should the report of results be printed. By default, verbose is set to be TRUE.
Details

The Wald, Rao’s score and Terrell’s gradient tests are performed using the expected Fisher information matrix.

Value

A matrix with three columns which contains the following:

- **Chi**: The value of the statistic of the test.
- **Df**: The number of degrees of freedom.
- **Pr(>Chi)**: The $p$-value of the test computed using the Chi-square distribution.

References


Examples

```r
## Example 1
Auto <- ISLR::Auto
fit1 <- glm(mpg ~ weight, family=inverse.gaussian("log"), data=Auto)
fit2 <- update(fit1, . ~ . + horsepower)
fit3 <- update(fit2, . ~ . + horsepower:weight)
anova2(fit1, fit2, fit3, test="lr")
anova2(fit1, fit2, fit3, test="score")
anova2(fit1, fit2, fit3, test="wald")
anova2(fit1, fit2, fit3, test="gradient")

## Example 2
burn1000 <- aplore3::burn1000
mod <- death ~ age + tbsa + inh_inj
fit1 <- glm(mod, family=binomial("logit"), data=burn1000)
fit2 <- update(fit1, . ~ . + inh_inj + age*inh_inj + tbsa*inh_inj)
anova2(fit1, fit2, test="lr")
anova2(fit1, fit2, test="score")
anova2(fit1, fit2, test="wald")
anova2(fit1, fit2, test="gradient")

## Example 3
data(aucuba)
fit <- glm(lesions ~ 1 + time, family=poisson("log"), data=aucuba)
anova2(fit, test="lr")
anova2(fit, test="score")
anova2(fit, test="wald")
anova2(fit, test="gradient")
```
### aucuba

**Lesions of Aucuba mosaic virus**

**Description**

The investigators counted the number of lesions of *Aucuba mosaic* virus developing after exposure to X rays for various times.

**Usage**

```r
data(aucuba)
```

**Format**

A data frame with 7 rows and 2 variables:

- **time**: a numeric vector giving the minutes of exposure.
- **lesions**: a numeric vector giving the counts of lesions, in hundreds.

**References**


**Examples**

```r
data(aucuba)
dev.new()
barplot(lesions ~ time, col="red", data=aucuba)
```

---

### bladder

**Bladder cancer in mice**

**Description**

Female mice were continuously fed dietary concentrations of 2-Acetylaminofluorene (2-AAF), a carcinogenic and mutagenic derivative of fluorene. Serially sacrificed, dead or moribund mice were examined for tumors and deaths dates recorded. These data consist of the incidences of bladder neoplasms in mice observed during 33 months.

**Usage**

```r
data(bladder)
```
BoxTidwell

Format

A data frame with 8 rows and 3 variables:

- **dose** a numeric vector giving the dose, in parts per $10^4$, of 2-AAF.
- **exposed** a numeric vector giving the number of mice exposed to each dose of 2-AAF.
- **cancer** a numeric vector giving the number of mice with bladder cancer for each dose of 2-AAF.

References


See Also

liver

Examples

```r
data(bladder)
dev.new()
barplot(100*cancer/exposed ~ dose, beside=TRUE, data=bladder, col="red",
      xlab="Dose of 2-AAF", ylab="% of mice with bladder cancer")
```

BoxTidwell  

*Box-Tidwell transformations*

Description

Computes the Box-Tidwell power transformations of the predictors in a regression model.

Usage

```r
BoxTidwell(
  object,
  transf,
  epsilon = 1e-04,
  maxiter = 30,
  trace = FALSE,
  digits = getOption("digits") - 2,
  ...
)
```
Arguments

- **object**: a model fit object.
- **transf**: an one-sided formula giving the predictors that are candidates for transformation.
- **epsilon**: an (optional) numerical value. If the maximum relative change in coefficients is less than `epsilon`, then convergence is declared. By default, `epsilon` is set to be 0.0001.
- **maxiter**: an (optional) positive integer value indicating the maximum number of iterations. By default, `maxiter` is set to be 30.
- **trace**: an (optional) logical indicating if should the record of iterations be printed. By default, `trace` is set to be FALSE.
- **digits**: an (optional) integer value indicating the number of decimal places to be used.
- **...**: further arguments passed to or from other methods.

Value

Two matrices with the values of marginal and omnibus tests.

### BoxTidwell.glm

**Box-Tidwell transformations in Generalized Linear Models**

**Description**

Computes the Box-Tidwell power transformations of the predictors in a generalized linear model.

**Usage**

```r
## S3 method for class 'glm'
BoxTidwell(
  object,
  transf,
  epsilon = 1e-04,
  maxiter = 30,
  trace = FALSE,
  digits = getOption("digits") - 2,
  ...
)
```

**Arguments**

- **object**: an object of the class `glm`.
- **transf**: an one-sided formula giving the quantitative predictors that are candidates for transformation.
epsilon an (optional) numerical value. If the maximum relative change in coefficients is less than \( \epsilon \), then convergence is declared. By default, \( \epsilon \) is set to be 0.0001.

maxiter an (optional) positive integer value indicating the maximum number of iterations. By default, \( \text{maxiter} \) is set to be 30.

trace an (optional) logical indicating if should the record of iterations be printed. By default, \( \text{trace} \) is set to be FALSE.

digits an (optional) integer value indicating the number of decimal places to be used.

... further arguments passed to or from other methods.

Value

a list list with components including

marginal a matrix with estimates and standard errors of the estimated powers, as well as the statistic and the p-value of the Wald test to assess the hypothesis \( H_0 : \tau = 1 \) versus \( H_1 : \tau \neq 1 \),

omnibus a matrix with the statistic and the p-value of the Wald test for null hypothesis that all powers are 1.

References


See Also

*BoxTidwell.lm*

Examples

```r
# Example 1: Skin cancer in women
data(skincancer)
fit1 <- glm(cases ~ age + city, offset=log(population), family=poisson(log), data=skincancer)
AIC(fit1)
BoxTidwell(fit1, transf= ~ age)
fit1 <- update(fit1, formula=cases ~ I(age^(-1/2)) + city)
AIC(fit1)

# Example 3: Gas mileage
data(Auto, package="ISLR")
fit3 <- glm(mpg ~ horsepower + weight, family=inverse.gaussian(log), data=Auto)
AIC(fit3)
BoxTidwell(fit3, transf= ~ horsepower + weight)
fit3 <- update(fit3, formula=mpg ~ I(horsepower^(-1/3)) + weight)
AIC(fit3)

# Example 4: Advertising
data(advertising)
```
fit4 <- glm(sales ~ TV + radio, family=gaussian(log), data=advertising)
AIC(fit4)
BoxTidwell(fit4, transf=~TV)
fit4 <- update(fit4, formula=sales ~ I(TV^{1/10}) + radio)
AIC(fit4)
fit4 <- update(fit4, formula=sales ~ I(TV^{1/10})*radio)
AIC(fit4)

BoxTidwell.lm  Box-Tidwell transformations in Normal Linear Models

Description
Computes the Box-Tidwell power transformations of the predictors in a normal linear model.

Usage
## S3 method for class 'lm'
BoxTidwell(
  object,
  transf,
  epsilon = 1e-04,
  maxiter = 30,
  trace = FALSE,
  digits = getOption("digits") - 2,
  ...
)

Arguments

- **object**
  an object of the class `lm`.

- **transf**
  an one-sided formula giving the quantitative predictors that are candidates for transformation.

- **epsilon**
  an (optional) numerical value. If the maximum relative change in coefficients is less than `epsilon`, then convergence is declared. By default, `epsilon` is set to be 0.0001.

- **maxiter**
  an (optional) positive integer value indicating the maximum number of iterations. By default, `maxiter` is set to be 30.

- **trace**
  an (optional) logical indicating if should the record of iterations be printed. By default, `trace` is set to be FALSE.

- **digits**
  an (optional) integer value indicating the number of decimal places to be used.

- **...**
  further arguments passed to or from other methods.

Value
a list list with components including
marginal a matrix with estimates and standard errors of the estimated powers, as well as the statistic and the p-value of the Wald test to assess the hypothesis \( H_0: \tau = 1 \) versus \( H_1: \tau \neq 1 \),

omnibus a matrix with the statistic and the p-value of the Wald test for null hypothesis that all powers are 1,

References


See Also

BoxTidwell.glm

Examples

###### Example 1: Hill races in Scotland

data(races)
fit1 <- lm(rtime ~ distance + cclimb, data=races)
AIC(fit1)
BoxTidwell(fit1, transf= ~ distance + cclimb)
fit1 <- update(fit1, formula=rtime ~ distance + I(cclimb^2))
AIC(fit1)

###### Example 2: Gasoline yield

data(mtcars)
fit2 <- lm(mpg ~ hp + wt + am, data=mtcars)
AIC(fit2)
BoxTidwell(fit2, transf= ~ hp + wt)
fit2 <- update(fit2, formula=mpg ~ log(hp) + log(wt) + am)
AIC(fit2)

###### Example 3: New York Air Quality Measurements

data(airquality)
fit3 <- lm(log(Ozone) ~ Solar.R + Wind + Temp, data=airquality)
AIC(fit3)
BoxTidwell(fit3, transf= ~ Solar.R + Wind + Temp)
fit3 <- update(fit3, formula=log(Ozone) ~ log(Solar.R) + Wind + Temp)
AIC(fit3)

###### Example 4: Heat capacity of hydrobromic acid

data(heatcap)
fit4 <- lm(Cp ~ Temp, data=heatcap)
AIC(fit4)
BoxTidwell(fit4, transf= ~ Temp)
fit4 <- update(fit4, formula=Cp ~ I(Temp^5))
AIC(fit4)

###### Example 5: Age and Eye Lens Weight of Rabbits in Australia

data(rabbits)
fit5 <- lm(log(wlens) ~ age, data=rabbits)
AIC(fit5)
BoxTidwell(fit5, transf= ~ age)
fit5 <- update(fit5, formula=log(wlens) - I(age^(-1/3)))
AIC(fit5)

brains

Mammal brain and body weights

Description

These data correspond to the (average) body weight and the (average) brain weight for sixty-two species of mammals.

Usage

data(brains)

Format

A data frame with 62 rows and 3 variables:

Specie a character string giving the species name.

BrainWt a numeric vector indicating the average brain weight, in grams.

BodyWt a numeric vector indicating the average body weight, in kilograms.

References


Examples

data(brains)
xlab <- "log(Body Weight)"
ylab <- "log(Brain Weight)"
dev.new()
with(brains, plot(log(BodyWt),log(BrainWt),pch=20,xlab=xlab,ylab=ylab))
**Description**

In a biomedical study of the immuno-activating ability of two agents, TNF (tumor necrosis factor) and IFN (interferon), to induce cell differentiation, the number of cells that exhibited markers of differentiation after exposure to TNF and IFN was recorded. At each of the 16 TNF/INF dose combinations, 200 cells were examined. The main question is whether the two agents stimulate cell differentiation synergistically or independently.

**Usage**

data(cellular)

**Format**

A data frame with 16 rows and 3 variables:

- **cells** a numeric vector giving the number of cells that exhibited markers of differentiation after exposure to the dose of the two agents
- **tnf** a numeric vector giving the dose (U/ml) of TNF
- **ifn** a numeric vector giving the dose (U/ml) of IFN

**References**


**Examples**

data(cellular)
dev.new()
barplot(100*cells/200 ~ ifn + tnf, beside=TRUE, data=cellular, col=terrain.colors(4),
       xlab="Dose of TNF", ylab="% of cells with markers of differentiation")
legend("topleft", legend=c("0","4","20","100"), fill=terrain.colors(4),
       title="Dose of IFN", bty="n")
Description

Inflation of the abdomen during laparoscopic cholecystectomy (removal of the gallbladder) separates the liver from the diaphragm and strains the attachments that connect both. This strain is felt as a referred shoulder pain. Suction to remove residual gas may reduce shoulder pain. There were 22 subjects randomized in the active group (with abdominal suction) and 19 subjects randomized in the control group (without abdominal suction). After laparoscopic surgery, patients were asked to rate their shoulder pain on a visual analog scale morning and afternoon for three days after the operation (a total of six different times). The scale was coded into five ordered categories where a pain score of 1 indicated "low pain" and a score of 5 reflected "high pain".

Usage

data(cholecystectomy)

Format

A data frame with 246 rows and 7 variables:

- **id**: a numeric vector with the identifier of the patient.
- **treatment**: a factor indicating the treatment received by the patient: abdominal suction ("A") and placebo ("P").
- **gender**: a factor indicating the gender of the patient: female ("F") and male ("M").
- **age**: a numeric vector indicating the age of the patient, in years.
- **time**: a numeric vector indicating the occasion the patient was asked to rate their shoulder pain after the laparoscopic surgery: integers from 1 to 6.
- **pain**: a numeric vector indicating the shoulder pain rated by the patient on a scale coded into five ordered categories, where 1 indicated "low pain" and 5 reflected "high pain".
- **pain2**: a numeric vector indicating the shoulder pain rated by the patient and coded as 1 for the two first categories of pain and 0 for other cases.

References

Examples

data(cholecystectomy)
out <- aggregate(pain2 ~ treatment + time, data=cholecystectomy, mean)
dev.new()
barplot(100*pain2 ~ treatment + time, beside=TRUE, data=out, xlab="Time",
        col=c("yellow","blue"), ylab="% of patients with low pain")
legend("topleft", c("Placebo","Abdominal suction"), fill=c("yellow","blue"),
       title="Treatment", cex=0.9, bty="n")

CIC

Correlation Information Criterion for Generalized Estimating Equations

Description

Computes the Correlation Information Criterion (CIC) for one or more objects of the class glmgee.

Usage

CIC(..., verbose = TRUE, digits = 3)

Arguments

... one or several objects of the class glmgee.
verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.
digits an (optional) integer indicating the number of digits to print.

Value

A data.frame with the values of the CIC for each glmgee object in the input.

References


See Also

QIC, GHYC, RJC, AGPC, SGPC
Examples

Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces

```r
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
CIC(fit1, fit2, fit3, fit4)
```

Example 2: Treatment for severe postnatal depression

```r
data(depression)
mod2 <- depressd ~ visit + group
fit1 <- glmgee(mod2, id=subj, family=binomial(logit), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
CIC(fit1, fit2, fit3, fit4)
```

Example 3: Treatment for severe postnatal depression (2)

```r
mod3 <- dep ~ visit*group
fit1 <- glmgee(mod3, id=subj, family=gaussian(identity), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Exchangeable")
CIC(fit1, fit2, fit3)
```

---

**confint2**

*Confidence Intervals for Generalized Linear Models*

**Description**

Computes confidence intervals based on Wald, likelihood-ratio, Rao’s score or Terrell’s gradient tests for a generalized linear model.

**Usage**

```r
confint2(
  model, 
  level = 0.95, 
  test = c("wald", "lr", "score", "gradient"),
  digits = 5,
  verbose = TRUE
)
```

**Arguments**

- **model**: an object of the class `glm`. 
level an (optional) value indicating the required confidence level. By default, level is set to be 0.95.

test an (optional) character string indicating the required type of test. The available options are: Wald ("wald"), Rao’s score ("score"), Terrell’s gradient ("gradient"), and likelihood ratio ("lr") tests. By default, test is set to be "wald".

digits an (optional) integer value indicating the number of decimal places to be used. By default, digits is set to be 5.

verbose an (optional) logical indicating if should the report of results be printed. By default, verbose is set to be TRUE.

Details
The approximate 100(1-level)% confidence interval for $\beta$ based on the test test is the set of values of $\beta_0$ for which the hypothesis $H_0: \beta = \beta_0$ versus $H_1: \beta \neq \beta_0$ is not rejected at the approximate significance level of 100(1-level)%.

Value
A matrix with so many rows as parameters in the linear predictor and two columns: "Lower limit" and "Upper limit".

References


Examples

##### Example 1: Fuel consumption of automobiles
Auto <- ISLR::Auto
fit1 <- glm(mpg ~ weight*horsepower, family=inverse.gaussian("log"), data=Auto)
confint2(fit1, test="lr")
confint2(fit1, test="score")

##### Example 2: Patients with burn injuries
burn1000 <- aplore3::burn1000
burn1000 <- within(burn1000, death <- factor(death, levels=c("Dead","Alive")))
fit2 <- glm(death ~ age*inh_inj + tbsa*inh_inj, family=binomial("logit"), data=burn1000)
confint2(fit2, test="lr")
confint2(fit2, test="gradient")
Description

Produces an approximation, better known as the one-step approximation, of the Cook's distance, which is aimed to measure the effect on the estimates of the parameters in the linear predictor of deleting each cluster/observation in turn. This function also can produce a cluster/observation-index plot of the Cook's distance for all parameters in the linear predictor or for some subset of them (via the argument coefs).

Usage

```r
## S3 method for class 'glmgee'
cooks.distance(
  model,
  method = c("Preisser-Qaqish", "full"),
  level = c("clusters", "observations"),
  plot.it = FALSE,
  coefs,
  identify,
  varest = c("robust", "df-adjusted", "model", "bias-corrected"),
  ...
)
```

Arguments

- **model**: an object of class `glmgee`.
- **method**: an (optional) character string indicating the method of calculation for the one-step approximation. The options are: the one-step approximation described by Preisser and Qaqish (1996) in which the working-correlation matrix is assumed to be known ("Preisser-Qaqish"); and the "authentic" one-step approximation ("full"). By default, method is set to be "Preisser-Qaqish".
- **level**: an (optional) character string indicating the level for which the Cook's distance is required. The options are: cluster-level ("clusters") and observation-level ("observations"). By default, level is set to be "clusters".
- **plot.it**: an (optional) logical indicating if the plot of Cook's distance is required or just the data matrix in which that plot is based. By default, plot.it is set to be FALSE.
- **coefs**: an (optional) character string which (partially) match with the names of some of the parameters in the linear predictor.
- **identify**: an (optional) integer indicating the number of clusters to identify on the plot of Cook's distance. This is only appropriate if plot.it=TRUE.
varest  
an (optional) character string indicating the type of estimator which should be used to the variance-covariance matrix of the interest parameters. The available options are: robust sandwich-type estimator ("robust"), degrees-of-freedom-adjusted estimator ("df-adjusted"), bias-corrected estimator ("bias-corrected"), and the model-based or naive estimator ("model"). By default, varest is set to be "robust".

...  

further arguments passed to or from other methods. If plot.it=TRUE then ... may be used to include graphical parameters to customize the plot. For example, col, pch, cex, main, sub, xlab, ylab.

Details

The Cook’s distance consists of the distance between two estimates of the parameters in the linear predictor using a metric based on the (estimate of the) variance-covariance matrix. For the cluster-level, the first one set of estimates is computed from a dataset including all clusters/observations, and the second one is computed from a dataset in which the i-th cluster is excluded. To avoid computational burden, the second set of estimates is replaced by its one-step approximation. See the dfbeta.glmgee documentation.

Value

A matrix as many rows as clusters/observations in the sample and one column with the values of the Cook’s distance.

References


Examples

```r
# Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces, corstr="AR-M-dependent")

### Cook’s distance for all parameters in the linear predictor
cooks.distance(fit1, method="full", plot.it=TRUE, col="red", lty=1, lwd=1, cex=0.8,
col.lab="blue", col.axis="blue", col.main="black", family="mono")

### Cook’s distance for the parameter associated to the variable 'treat'
cooks.distance(fit1, coef="treat", method="full", plot.it=TRUE, col="red", lty=1,
lwd=1, col.lab="blue", col.axis="blue", col.main="black", cex=0.8)

# Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ visit + group
```
```r
fit2 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent", data=depression)

### Cook's distance for all parameters in the linear predictor
cooks.distance(fit2, method="full", plot.it=TRUE, col="red", lty=1, lwd=1, cex=0.8, 
col.lab="blue", col.axis="blue", col.main="black", family="mono")

### Cook's distance for the parameter associated to the variable 'group'
cooks.distance(fit2, coef="group", method="full", plot.it=TRUE, col="red", lty=1, 
lwd=1, col.lab="blue", col.axis="blue", col.main="black", cex=0.8)
```

---

**cooks.distance.overglm**

*Cook's Distance for alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion*

**Description**

Produces an approximation, better known as the *one-step approximation*, of the Cook's distance, which is aimed to measure the effect on the estimates of the parameters in the linear predictor of deleting each observation in turn. This function also can produce an index plot of the Cook's distance for all parameters in the linear predictor or for some subset of them (via the argument coefs).

**Usage**

```r
## S3 method for class 'overglm'
cooks.distance(model, plot.it = FALSE, coefs, identify, ...)
```

**Arguments**

- `model` an object of class `overglm`.
- `plot.it` an (optional) logical indicating if the plot is required or just the data matrix in which that plot is based. By default, `plot.it` is set to be `FALSE`.
- `coefs` an (optional) character string which (partially) match with the names of some model parameters.
- `identify` an (optional) integer indicating the number of individuals to identify on the plot of the Cook's distance. This is only appropriate if `plot.it=TRUE`.
- `...` further arguments passed to or from other methods. If `plot.it=TRUE` then ... may be used to include graphical parameters to customize the plot. For example, `col`, `pch`, `cex`, `main`, `sub`, `xlab`, `ylab`.

**Details**

The Cook's distance consists of the *distance* between two estimates of the parameters in the linear predictor using a metric based on the (estimate of the) variance-covariance matrix. The first one set of estimates is computed from a dataset including all individuals, and the second one is computed from a dataset in which the *i*-th individual is excluded. To avoid computational burden, the second set of estimates is replaced by its *one-step approximation*. See the `dfbeta.overglm` documentation.
Value

A matrix as many rows as individuals in the sample and one column with the values of the Cook’s distance.

Examples

### Example 1: Self diagnosed ear infections in swimmers
```r
data(swimmers)
fit1 <- overglm(infections ~ frequency + location, family="nb1(log)", data=swimmers)

### Cook's distance for all parameters in the linear predictor
cooks.distance(fit1, plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", col.main="black", family="mono", cex=0.8)

### Cook's distance just for the parameter associated with 'frequency'
cooks.distance(fit1, plot.it=TRUE, coef="frequency", col="red", lty=1, lwd=1, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)

### Example 2: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit2 <- overglm(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)

### Cook's distance for all parameters in the linear predictor
cooks.distance(fit2, plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", col.main="black", family="mono", cex=0.8)

### Cook's distance just for the parameter associated with 'fem'
cooks.distance(fit2, plot.it=TRUE, coef="fem", col="red", lty=1, lwd=1, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)

### Example 3: Agents to stimulate cellular differentiation
```r
data(cellular)
fit3 <- overglm(cbind(cells,200-cells) ~ tnf + ifn, family="bb(logit)", data=cellular)

### Cook's distance for all parameters in the linear predictor
cooks.distance(fit3, plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", col.main="black", family="mono", cex=0.8)

### Cook's distance just for the parameter associated with 'tnf'
cooks.distance(fit3, plot.it=TRUE, coef="tnf", col="red", lty=1, lwd=1, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)
Description

Produces an approximation, better known as the one-step approximation, of the Cook’s distance, which is aimed to measure the effect on the estimates of the parameters in the linear predictor of deleting each observation in turn. This function also can produce an index plot of the Cook’s distance for all parameters in the linear predictor or for some subset of them (via the argument `coefs`).

Usage

```r
# S3 method for class 'zeroinflation'
cooks.distance(model,
               submodel = c("counts", "zeros", "full"),
               plot.it = FALSE,
               coefs,
               identify,
               ...
)
```

Arguments

- `model` an object of class `zeroinflation`.
- `submodel` an (optional) character string which allows to specify the model: "counts", "zeros" or "full". By default, `submodel` is set to be "counts".
- `plot.it` an (optional) logical indicating if the plot is required or just the data matrix in which that plot is based. By default, `plot.it` is set to be FALSE.
- `coefs` an (optional) character string which (partially) match with the names of some model parameters.
- `identify` an (optional) integer indicating the number of individuals to identify on the plot of the Cook’s distance. This is only appropriate if `plot.it`=TRUE.
- `...` further arguments passed to or from other methods. If `plot.it`=TRUE then ... may be used to include graphical parameters to customize the plot. For example, `col`, `pch`, `cex`, `main`, `sub`, `xlab`, `ylab`.

Details

The Cook’s distance consists of the distance between two estimates of the parameters in the linear predictor using a metric based on the (estimate of the) variance-covariance matrix. The first one set of estimates is computed from a dataset including all individuals, and the second one is computed from a dataset in which the $i$-th individual is excluded. To avoid computational burden, the second set of estimates is replaced by its one-step approximation. See the `dfbeta.zeroinflation` documentation.

Value

A matrix as many rows as individuals in the sample and one column with the values of the Cook’s distance.
### Example 1: Self diagnosed ear infections in swimmers

data(swimmers)
fit <- zeroinf(infections ~ frequency + location, family="nb1(log)", data=swimmers)

### Cook's distance for all parameters in the "counts" model
cooks.distance(fit, submodel="counts", plot.it=TRUE, col="red", lty=1, lwd=1,
               col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)

### Cook's distance for all parameters in the "zeros" model
cooks.distance(fit, submodel="zeros", plot.it=TRUE, col="red", lty=1, lwd=1,
               col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)

---

### Description

The market research department of a soft drink manufacturer is investigating the effectiveness of a price discount coupon on the purchase of a two-litre beverage product. A sample of 5500 customers received coupons for varying price discounts between 5 and 25 cents. The main objective of the analysis is to determine if the price discount affects the proportion of redeemed coupons after one month.

### Usage

data(coupons)

### Format

A data frame with 11 rows and 3 variables:

- **discounts** a numeric vector indicating the price discount, in cents.
- **costumers** a numeric vector indicating the number of customers who received coupons.
- **redeemed** a numeric vector indicating the number of redeemed coupons.

### References


### Examples

dev.new()
data(coupons)
barplot(100*redeemed/costumers ~ discounts, data=coupons, xlab="Discount price",
ylab="(%) Redeemed coupons", col="blue")
Description

These data arose from a study on the efficacy of oestrogen given transdermally for the treatment of severe postnatal depression. Women with major depression were randomly assigned to a placebo control group or an oestrogen patch group. Prior to the treatment all women were assessed by self-rated depressive symptoms on the Edinburgh Postnatal Depression Scale (EPDS). EPDS data were collected monthly for six months once treatment began. Higher EDPS scores are indicative of higher depression levels.

Usage

data(depression)

Format

A data frame with 427 rows and 5 variables:

- subj a numeric vector giving the identifier of each woman.
- group a factor giving the received treatment: "placebo" or "oestrogen".
- visit a numeric vector giving the number of months since the treatment began, where -1 indicates the pretreatment assessment of the EPDS.
- dep a numeric vector giving the value of the EDPS.
- depressd a numeric vector coded as 1 when the value of the EDPS is greater than or equal to 11 and coded as 0 in other cases.

Source


References


Examples

data(depression)
dev.new()
boxplot(dep ~ visit, data=subset(depression,group="placebo"), at=c(0:6) - 0.2, col="yellow", boxwex=0.3, xaxt="n", ylim=range(na.omit(depression$dep)), xlab="Months since the treatment began", ylab="EDPS")
boxplot(dep ~ visit, data=subset(depression,group="oestrogen"), add=TRUE, at=c(0:6) + 0.2, col="blue", boxwex=0.3, xaxt="n")
axis(1, at=c(0:6), labels=c(-1,1:6))
dfbeta.glmgee

Description

Produces an approximation, better known as the one-step approximation, of the effect on the parameter estimates of deleting each cluster/observation in turn. This function also can produce an index plot of the Dfbeta Statistic for some parameters via the argument coefs.

Usage

```r
## S3 method for class 'glmgee'
dfbeta(model,
       level = c("clusters", "observations"),
       method = c("Preisser-Qaqish", "full"),
       coefs,
       identify,
       ...)
```

Arguments

- `model`: an object of class glmgee.
- `level`: an (optional) character string indicating the level for which the Dfbeta statistic is required. The options are: cluster-level ("clusters") and observation-level ("observations"). By default, level is set to be "clusters".
- `method`: an (optional) character string indicating the method of calculation for the one-step approximation. The options are: the one-step approximation described by Preisser and Qaqish (1996) in which the working-correlation matrix is assumed to be known ("Preisser-Qaqish"); and the "authentic" one-step approximation ("full"). By default, method is set to be "Preisser-Qaqish".
- `coefs`: an (optional) character string which (partially) match with the names of some parameters in the linear predictor.
- `identify`: an (optional) integer indicating the number of clusters/observations to identify on the plot of the Dfbeta statistic. This is only appropriate if coefs is specified.
- `...`: further arguments passed to or from other methods. If coefs is specified then ... may be used to include graphical parameters to customize the plot. For example, col, pch, cex, main, sub, xlab, ylab.
Details

The *one-step approximation* (with the method "full") of the estimates of the parameters in the linear predictor of a GEE when the *i*-th cluster is excluded from the dataset is given by the vector obtained as the result of the first iteration of the fitting algorithm of that GEE when it is performed using: (1) a dataset in which the *i*-th cluster is excluded; and (2) a starting value which is the solution to the same GEE but based on the dataset including all clusters.

Value

A matrix with so many rows as clusters/observations in the sample and so many columns as parameters in the linear predictor. For clusters, the *i*-th row of that matrix corresponds to the difference between the estimates of the parameters in the linear predictor using all clusters and the *one-step approximation* of those estimates when the *i*-th cluster is excluded from the dataset.

References


Examples

#### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces

```r
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), corstr="AR-M-dependent", data=spruces)
dfbs1 <- dfbeta(fit1, method="full", coefs="treat", col="red", lty=1, lwd=1, col.lab="blue",
    col.axis="blue", col.main="black", family="mono", cex=0.8, main="treat")
```

### Calculation by hand of dfbeta for the tree labeled by "N1T01"

```r
onestep1 <- glmgee(mod1, id=tree, family=Gamma(log), corstr="AR-M-dependent",
data=spruces, start=coef(fit1), subset=c(tree!="N1T01"), maxit=1)
coef(fit1)-coef(onestep1)
dfbs1[rownames(dfbs1)=="N1T01",]
```

#### Example 2: Treatment for severe postnatal depression

```r
data(depression)
mod2 <- depressd ~ visit + group
fit2 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent",
data=depression)
dfbs2 <- dfbeta(fit2, method="full", coefs="group", col="red", lty=1, lwd=1, col.lab="blue",
    col.axis="blue", col.main="black", family="mono", cex=0.8, main="group")
```

### Calculation by hand of dfbeta for the woman labeled by "18"

```r
onestep2 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent",
data=depression, start=coef(fit2), subset=c(subj!=18), maxit=1)
```
dfbeta.overglm

Example 3: Treatment for severe postnatal depression (2)

```r
mod3 <- dep ~ visit*group
fit3 <- glmgee(mod3, id=subj, family=gaussian(identity), corstr="AR-M-dependent", data=depression)

dfbs3 <- dfbeta(fit3, method="full", coefs="visit:group", col="red", lty=1, lwd=1, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8, main="visit:group")
```

### Calculation by hand of dfbeta for the woman labeled by "18"

```r
onestep3 <- glmgee(mod3, id=subj, family=gaussian(identity), corstr="AR-M-dependent", data=depression, start=coef(fit3), subset=c(subj!=18), maxit=1)

coeffit3-coef(onestep3)
dfbs3[rownames(dfbs3)==18,]
```

---

dfbeta.overglm

**Dfbeta statistic for alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion.**

**Description**

Produces an approximation, better known as the *one-step approximation*, of the effect on the parameter estimates of deleting each individual in turn. This function also can produce an index plot of the Dfbeta statistic for some parameter chosen via the argument `coefs`.

**Usage**

```r
## S3 method for class 'overglm'

dfbeta(model, coefs, identify, ...)
```

**Arguments**

- `model` an object of class `overglm`.
- `coefs` an (optional) character string which (partially) match with the names of some model parameters.
- `identify` an (optional) integer indicating the number of individuals to identify on the plot of the Dfbeta statistic. This is only appropriate if `coefs` is specified.
- `...` further arguments passed to or from other methods. If `plot.it=TRUE` then `...` may be used to include graphical parameters to customize the plot. For example, `col, pch, cex, main, sub, xlab, ylab`.

---
Details

The one-step approximation of the estimates of the parameters when the \(i\)-th individual is excluded from the dataset consists of the vector obtained as result of the first iteration of the Newton-Raphson algorithm when it is performed using: (1) a dataset in which the \(i\)-th individual is excluded; and (2) a starting value which is the estimate of the same model but based on the dataset including all individuals.

Value

A matrix with so many rows as individuals in the sample and so many columns as parameters in the linear predictor. The \(i\)-th row of that matrix corresponds to the difference between the estimates of the parameters in the linear predictor using all individuals and the one-step approximation of those estimates when the \(i\)-th individual is excluded from the dataset.

References


Examples

##### Example 1: Self diagnozed ear infections in swimmers

data(swimmers)
fit1 <- overglm(infections ~ frequency + location, family="nb1(log)", data=swimmers)
dfbeta(fit1, coefs="frequency", col="red", lty=1, lwd=1, col.lab="blue",
       col.axis="blue", col.main="black", family="mono", cex=0.8, main="frequency")

##### Example 2: Article production by graduate students in biochemistry PhD programs

bioChemists <- pscl::bioChemists
fit2 <- overglm(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)
dfbeta(fit2, coefs="fem", col="red", lty=1, lwd=1, col.lab="blue",
       col.axis="blue", col.main="black", family="mono", cex=0.8, main="fem")

##### Example 3: Agents to stimulate cellular differentiation

data(cellular)
fit3 <- overglm(cbind(cells,200-cells) ~ tnf + ifn, family="bb(logit)", data=cellular)
dfbeta(fit3, coefs="tnf", col="red", lty=1, lwd=1, col.lab="blue",
       col.axis="blue", col.main="black", family="mono", cex=0.8, main="tnf")

dfbeta.zeroinflation  *Dfbeta statistic for Regression Models to deal with Zero-Excess in Count Data*

Description

Produces an approximation, better known as the one-step approximation, of the effect on the parameter estimates of deleting each individual in turn. This function also can produce an index plot of the Dfbeta statistic for some parameter chosen via the argument coefs.
Usage

```r
## S3 method for class 'zeroinflation'
dfbeta(model, submodel = c("counts", "zeros"), coefs, identify, ...)```

Arguments

- `model` an object of class `zeroinflation`.
- `submodel` an (optional) character string which allows to specify the model: "counts" or "zeros". By default, `submodel` is set to be "counts".
- `coefs` an (optional) character string which (partially) match with the names of some model parameters.
- `identify` an (optional) integer indicating the number of individuals to identify on the plot of the Dfbeta statistic. This is only appropriate if `coefs` is specified.
- `...` further arguments passed to or from other methods. If `plot.it=TRUE` then `...` may be used to include graphical parameters to customize the plot. For example, `col`, `pch`, `cex`, `main`, `sub`, `xlab`, `ylab`.

Details

The one-step approximation of the estimates of the parameters when the i-th individual is excluded from the dataset consists of the vector obtained as result of the first iteration of the Newton-Raphson algorithm when it is performed using: (1) a dataset in which the i-th individual is excluded; and (2) a starting value which is the estimate of the same model but based on the dataset including all individuals.

Value

A matrix with so many rows as individuals in the sample and so many columns as parameters in the linear predictor. The i-th row of that matrix corresponds to the difference between the estimates of the parameters in the linear predictor using all individuals and the one-step approximation of those estimates when the i-th individual is excluded from the dataset.

References


Examples

```r
## Example 1: Self diagnosed ear infections in swimmers
data(swimmers)
fit <- zeroinf(infections ~ frequency + location, family="nb1(log)", data=swimmers)
dfbeta(fit, submodel="counts", coefs="frequency", col="red", lty=1, lwd=1,
col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)
dfbeta(fit, submodel="zeros", coefs="location", col="red", lty=1, lwd=1,
col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)
```
**dilution**  
*Description*

These data are counts of virus particles at 5 different dilutions. There are 4 replicate counts at each dilution except the last for which there are 5 counts. The aim is to estimate the expected number of virus particles per unit volume.

**Usage**

```r
data(dilution)
```

**Format**

A data frame with 21 rows and 2 variables:

- **Count** a numeric vector indicating the count of virus particles.
- **Dilution** a numeric vector indicating the dilution volume.

**Source**


**Examples**

```r
data(dilution)
xlab <- "Dilution volume"
 ylab <- "Count of virus particles"
dev.new()
with(dilution, plot(Dilution, Count, pch=20, xlab=xlab, ylab=ylab))
```

---

**envelope**  
*Description*

Generic function for building a normal QQ-plot with simulated envelope of residuals obtained from a fitted model.

**Usage**

```r
envelope(object, ...)
```
envelope.glm

Arguments

object a fitted model object.

... further arguments passed to or from other methods.

Value

A matrix with the simulated envelope and, optionally, a plot of it.

envelope.glm Normal QQ-plot with simulated envelope of residuals in GLMs

Description

Produces a normal QQ-plot with simulated envelope of residuals for generalized linear models.

Usage

## S3 method for class 'glm'
envelope(
  object,
  rep = 25,
  conf = 0.95,
  type = c("quantile", "deviance", "pearson"),
  standardized = FALSE,
  plot.it = TRUE,
  identify,
  ...
)

Arguments

object an object of the class glm.

rep an (optional) positive integer which allows to specify the number of replicates which should be used to build the simulated envelope. By default, rep is set to be 25.

conf an (optional) value in the interval (0,1) indicating the confidence level which should be used to build the pointwise confidence intervals, which form the envelope. By default, conf is set to be 0.95.

type a character string indicating the type of residuals which should be used. The available options are: randomized quantile ("quantile"), deviance ("deviance") and pearson ("pearson") residuals. By default, type is set to be "quantile".

standardized an (optional) logical switch indicating if the residuals should be standardized by dividing by the square root of \((1 - h)\), where \(h\) is a measure of leverage. By default, standardized is set to be FALSE.
plot.it

an (optional) logical switch indicating if the normal QQ-plot with simulated envelope of residuals is required or just the data matrix in which it is based. By default, plot.it is set to be TRUE.

identify

an (optional) positive integer indicating the number of individuals to identify on the QQ-plot with simulated envelope of residuals. This is only appropriate if plot.it=TRUE.

... further arguments passed to or from other methods. If plot.it=TRUE then ...

may be used to include graphical parameters to customize the plot. For example, col, pch, cex, main, sub, xlab, ylab.

Details

The simulated envelope is built by simulating rep independent realizations of the response variable for each individual, which is accomplished taking into account the following: (1) the model assumption about the distribution of the response variable; (2) the estimates of the parameters in the linear predictor; and (3) the estimate of the dispersion parameter. The interest model is re-fitted rep times, as each time the vector of observed responses is replaced by one of the simulated samples. The type-type residuals are computed and then sorted for each replicate, so that for each $i = 1, 2, ..., n$, where $n$ is the number of individuals in the sample, there is a random sample of size rep of the $i$-th order statistic of the type-type residuals. Therefore, the simulated envelope is composed of the quantiles $(1 - \text{conf})/2$ and $(1 + \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$.

Value

A matrix with the following four columns:

- **Lower limit**  
  the quantile $(1 - \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$.

- **Median**  
  the quantile 0.5 of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$.

- **Upper limit**  
  the quantile $(1 + \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$.

- **Residuals**  
  the observed type-type residuals.

References


envelope.lm

See Also

envelope.lm, envelope.overglm

Examples

##### Example 1:
burn1000 <- aplore3::burn1000
burn1000 <- within(burn1000, death <- factor(death, levels=c("Dead","Alive")))
fit1 <- glm(death ~ age*inh_inj + tbsa*inh_inj, family=binomial("logit"), data=burn1000)
envelope(fit1, rep=50, conf=0.95, type="pearson", col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)

##### Example 2: Fuel consumption of automobiles
Auto <- ISLR::Auto
fit2 <- glm(mpg ~ horsepower*weight, family=inverse.gaussian("log"), data=Auto)
envelope(fit2, rep=50, conf=0.95, type="pearson", col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)

##### Example 3: Skin cancer in women
data(skincancer)
fit3 <- glm(cases ~ offset(log(population)) + city + age, family=poisson, data=skincancer)
envelope(fit3, rep=100, conf=0.95, type="quantile", col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)

##### Example 4: Self diagnosed ear infections in swimmers
data(swimmers)
fit4 <- glm(infections ~ frequency + location, family=poisson(log), data=swimmers)
envelope(fit4, rep=100, conf=0.95, type="quantile", col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)

##### Example 5: Agents to stimulate cellular differentiation
data(cellular)
fit5 <- glm(cbind(cells,200-cells) ~ tnf + ifn, family=binomial(logit), data=cellular)
envelope(fit5, rep=100, conf=0.95, type="quantile", col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)

envelope.lm

Normal QQ-plot with simulated envelope of residuals for normal linear models

Description

Produces a normal QQ-plot with simulated envelope of residuals obtained from the fit of a normal linear model.
Usage

```r
## S3 method for class 'lm'
envelope(
  object,
  rep = 100,
  conf = 0.95,
  type = c("external", "internal"),
  plot.it = TRUE,
  identify,
  ...
)
```

Arguments

- **object**: an object of the class `lm`.
- **rep**: an (optional) positive integer indicating the number of replicates which should be used to build the simulated envelope. By default, `rep` is set to be 100.
- **conf**: an (optional) value in the interval (0,1) indicating the confidence level which should be used to build the pointwise confidence intervals, which form the envelope. By default, `conf` is set to be 0.95.
- **type**: a character string indicating the type of residuals which should be used. The available options are: internally Studentized ("internal") and externally Studentized ("external") residuals. See Cook and Weisberg (1982, pages 18-20).
- **plot.it**: an (optional) logical switch indicating if the normal QQ-plot with simulated envelope of residuals is required or just the data matrix in which it is based. By default, `plot.it` is set to be TRUE.
- **identify**: an (optional) positive integer value indicating the number of individuals to identify on the QQ-plot with simulated envelope of residuals. This is only appropriate if `plot.it=TRUE`.
- **...**: further arguments passed to or from other methods. If `plot.it=TRUE` then `...` may be used to include graphical parameters to customize the plot. For example, `col`, `pch`, `cex`, `main`, `sub`, `xlab`, `ylab`.

Details

The simulated envelope is built by simulating `rep` independent realizations of the response variable for each individual, which is accomplished taking into account the following: (1) the model assumption about the distribution of the response variable; (2) the estimates of the parameters in the linear predictor; and (3) the estimate of the dispersion parameter. The interest model is re-fitted `rep` times, as each time the vector of observed responses is replaced by one of the simulated samples. The type-type residuals are computed and then sorted for each replicate, so that for each `i = 1, 2, ..., n`, where `n` is the number of individuals in the sample, there is a random sample of size `rep` of the `i`-th order statistic of the type-type residuals. Therefore, the simulated envelope is composed of the quantiles `(1 - conf)/2` and `(1 + conf)/2` of the random sample of size `rep` of the `i`-th order statistic of the type-type residuals for `i = 1, 2, ..., n`. 
Value

A matrix with the following four columns:

- **Lower limit** the quantile \((1 - \text{conf})/2\) of the random sample of size \(\text{rep}\) of the \(i\)-th order statistic of the type-type residuals for \(i = 1, 2, \ldots, n\),
- **Median** the quantile 0.5 of the random sample of size \(\text{rep}\) of the \(i\)-th order statistic of the type-type residuals for \(i = 1, 2, \ldots, n\),
- **Upper limit** the quantile \((1 + \text{conf})/2\) of the random sample of size \(\text{rep}\) of the \(i\)-th order statistic of the type-type residuals for \(i = 1, 2, \ldots, n\),
- **Residuals** the observed type-type residuals,

References


See Also

envelope.glm, envelope.overglm

Examples

```
Example 1: Fuel consumption of automobiles
fit1 <- lm(mpg ~ log(hp) + log(wt), data=mtcars)
envelope(fit1, rep=100, conf=0.95, type="external", col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)
```

```
Example 2: Species richness in plots
data(richness)
fit2 <- lm(Species ~ Biomass + pH + Biomass*pH, data=richness)
envelope(fit2, rep=100, conf=0.95, type="internal", col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)
```

```
Example 3: Gas consumption in a home before and after insulation
whiteside <- MASS::whiteside
fit3 <- lm(Gas ~ Temp + Insul + Temp*Insul, data=whiteside)
envelope(fit3, rep=100, conf=0.95, type="internal", col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)
```
Normal QQ-plot with Simulated Envelope of Residuals for alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion

Description

Produces a normal QQ-plot with simulated envelope of residuals for regression models based on the negative binomial, beta-binomial, and random-clumped binomial distributions, which are alternatives to the Poisson and binomial regression models under the presence of overdispersion.

Usage

```r
## S3 method for class 'overglm'
envolve(
  object,
  rep = 25,
  conf = 0.95,
  type = c("quantile", "response", "standardized"),
  plot.it = TRUE,
  identify,
  ...
)
```

Arguments

- `object`: an object of class `overglm`.
- `rep`: an (optional) positive integer which allows to specify the number of replicates which should be used to build the simulated envelope. By default, `rep` is set to be 25.
- `conf`: an (optional) value in the interval (0, 1) indicating the confidence level which should be used to build the pointwise confidence intervals, which conform the simulated envelope. By default, `conf` is set to be 0.95.
- `type`: an (optional) character string which allows to specify the required type of residuals. The available options are: (1) the difference between the observed response and the fitted mean ("response"); (2) the standardized difference between the observed response and the fitted mean ("standardized"); and (3) the randomized quantile residual ("quantile"). By default, `type` is set to be "quantile".
- `plot.it`: an (optional) logical switch indicating if the normal QQ-plot with simulated envelope of residuals is required or just the data matrix in which it is based. By default, `plot.it` is set to be TRUE.
- `identify`: an (optional) positive integer value indicating the number of individuals to identify on the QQ-plot with simulated envelope of residuals. This is only appropriate if `plot.it=TRUE`.
- `...`: further arguments passed to or from other methods. If `plot.it=TRUE` then `...` may be used to include graphical parameters to customize the plot. For example, `col`, `pch`, `cex`, `main`, `sub`, `xlab`, `ylab`.
Details

The simulated envelope is built by simulating rep independent realizations of the response variable for each individual, which is accomplished taking into account the following: (1) the model assumption about the distribution of the response variable; (2) the estimates of the parameters in the linear predictor; and (3) the estimate of the dispersion parameter. The interest model is re-fitted rep times, as each time the vector of observed responses is replaced by one of the simulated samples. The type-type residuals are computed and then sorted for each replicate, so that for each $i = 1, 2, ..., n$, where $n$ is the number of individuals in the sample, there is a random sample of size rep of the $i$-th order statistic of the type-type residuals. Therefore, the simulated envelope is composed of the quantiles $(1 - \text{conf})/2$ and $(1 + \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$.

Value

A matrix with the following four columns:

- **Lower limit** the quantile $(1 - \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$,
- **Median** the quantile 0.5 of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$,
- **Upper limit** the quantile $(1 + \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$,
- **Residuals** the observed type-type residuals,

References


See Also

envelope.lm, envelope.glm, envelope.zeroinflation

Examples

```
#### Example 1: Self diagnosed ear infections in swimmers
data(swimmers)
fit1 <- overglm(infections ~ frequency + location, family="nb1(log)", data=swimmers)
envelope(fit1, rep=30, conf=0.95, type="quantile", col="red", pch=20, col.lab="blue",
        col.axis="blue", col.main="black", family="mono", cex=0.8, plot.it=TRUE)
```

```
#### Example 2: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit2 <- overglm(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)
```
envelope(fit2, rep=30, conf=0.95, type="quantile", col="red", pch=20, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8, plot.it=TRUE)

Example 3: Agents to stimulate cellular differentiation
data(cellular)
fit3 <- overglm(cbind(cells,200-cells) ~ tnf + ifn, family="bb(logit)", data=cellular)
envelope(fit3, rep=30, conf=0.95, type="quantile", col="red", pch=20, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8, plot.it=TRUE)

envelope.zeroinflation

Normal QQ-plot with Simulated Envelope of Residuals for Regression Models to deal with Zero-Excess in Count Data

Description

Produces a normal QQ-plot with simulated envelope of residuals for regression models used to deal with zero-excess in count data.

Usage

## S3 method for class 'zeroinflation'
envelope(
  object,
  rep = 20,
  conf = 0.95,
  type = c("quantile", "response", "standardized"),
  plot.it = TRUE,
  identify,
  ...
)

Arguments

object an object of the class zeroinflation.
rep an (optional) positive integer which allows to specify the number of replicates which should be used to build the simulated envelope. By default, rep is set to be 25.
conf an (optional) value in the interval (0,1) indicating the confidence level which should be used to build the pointwise confidence intervals, which conform the simulated envelope. By default, conf is set to be 0.95.
type an (optional) character string which allows to specify the required type of residuals. The available options are: (1) the difference between the observed response and the fitted mean ("response"); (2) the standardized difference between the observed response and the fitted mean ("standardized"); (3) the randomized quantile residual ("quantile"). By default, type is set to be "quantile".
plot.it

an (optional) logical switch indicating if the normal QQ-plot with simulated envelope of residuals is required or just the data matrix in which it is based. By default, plot.it is set to be TRUE.

identify

an (optional) positive integer value indicating the number of individuals to identify on the QQ-plot with simulated envelope of residuals. This is only appropriate if plot.it=TRUE.

... further arguments passed to or from other methods. If plot.it=TRUE then ... may be used to include graphical parameters to customize the plot. For example, col, pch, cex, main, sub, xlab, ylab.

Details

The simulated envelope is built by simulating rep independent realizations of the response variable for each individual, which is accomplished taking into account the following: (1) the model assumption about the distribution of the response variable; (2) the estimates of the parameters in the linear predictor; and (3) the estimate of the dispersion parameter. The interest model is re-fitted rep times, as each time the vector of observed responses is replaced by one of the simulated samples. The type-type residuals are computed and then sorted for each replicate, so that for each $i = 1, 2, ..., n$, where $n$ is the number of individuals in the sample, there is a random sample of size rep of the $i$-th order statistic of the type-type residuals. Therefore, the simulated envelope is composed of the quantiles $(1 - \text{conf})/2$ and $(1 + \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$.

Value

A matrix with the following four columns:

| Lower limit | the quantile $(1 - \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$, |
| Median      | the quantile 0.5 of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$, |
| Upper limit | the quantile $(1 + \text{conf})/2$ of the random sample of size rep of the $i$-th order statistic of the type-type residuals for $i = 1, 2, ..., n$, |
| Residuals   | the observed type-type residuals. |

References


See Also

envelope.lm, envelope.glm, envelope.overglm
Examples

#### Example 1: Self diagnosed ear infections in swimmers

data(swimmers)
fit <- zeroinf(infections ~ frequency | location, family="nb1(log)", data=swimmers)
envelope(fit, rep=30, conf=0.95, type="quantile", col="red", pch=20, col.lab="blue",
  col.axis="blue", col.main="black", family="mono", cex=0.8)

---

estequa  

Function to extract estimating equations

Description

Extracts estimating equations evaluated at the parameter estimates and the observed data for a fitted model object.

Usage

esteem(object, ...)

Arguments

object  a fitted model object.
...

Value

A vector with the value of the estimating equations evaluated at the parameter estimates and the observed data.

---

estequa.glm  

Estimating Equations in Generalized Linear Models

Description

Extracts estimating equations evaluated at the parameter estimates and the observed data for a generalized linear model fitted to the data.

Usage

## S3 method for class 'glm'
estequa(object, ...)

---
Arguments

object an object of the class glm which is obtained from the fit of a generalized linear model.

... further arguments passed to or from other methods.

Value

A vector with the value of the estimating equations evaluated at the parameter estimates and the observed data.

Examples

## Example 1
Auto <- ISLR::Auto
mod <- mpg ~ cylinders + displacement + acceleration + origin + horsepower*weight
fit1 <- glm(mod, family=inverse.gaussian("log"), data=Auto)
estequa(fit1))

## Example 2
burn1000 <- aplore3::burn1000
burn1000 <- within(burn1000, death <- factor(death, levels=c("Dead","Alive")))
mod2 <- death ~ age + gender + race + tbsa + inh_inj + flame + age*inh_inj + tbsa*inh_inj
fit2 <- glm(mod2, family=binomial("logit"), data=burn1000)
estequa(fit2)

## Example 3
data(skincancer)
fit3 <- glm(cases ~ offset(log(population)) + city + age, family=poisson("log"), data=skincancer)
estequa(fit3)
Value

A vector with the value of the estimating equations evaluated at the parameter estimates and the observed data.

Examples

##### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), corstr="AR-M-dependent", data=spruces)
estequa(fit1)

##### Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ visit + group
fit2 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent", data=depression)
estequa(fit2)

##### Example 3: Treatment for severe postnatal depression (2)
mod3 <- dep - visit*group
fit3 <- glmgee(mod3, id=subj, family=gaussian(identity), corstr="AR-M-dependent", data=depression)
estequa(fit3)

##### Example 4: Dental Clinical Trial
data(rinse)
mod4 <- score/3.6 ~ rinse*time
fit4 <- glmgee(mod4, family=binomial(log), id=subject, corstr="Exchangeable", data=rinse)
estequa(fit4)

##### Example 5: Shoulder Pain after Laparoscopic Cholecystectomy
data(cholecystectomy)
mod5 <- pain2 ~ treatment + age + time
corstr <- "Stationary-M-dependent(2)"
fit5 <- glmgee(mod5, family=binomial(logit), id=id, corstr=corstr, data=cholecystectomy)
estequa(fit5)

##### Example 6: Guidelines for Urinary Incontinence Discussion and Evaluation
data(GUIDE)
mod6 <- bothered ~ gender + age + dayacc + severe + toilet
fit6 <- glmgee(mod6, family=binomial(logit), id=practice, corstr="Exchangeable", data=GUIDE)
estequa(fit6)

##### Example 7: Tests of Auditory Perception in Children with OME
OME <- MASS::OME
mod7 <- cbind(Correct, Trials-Correct) ~ Loud + Age + OME
fit7 <- glmgee(mod7, family = binomial(cloglog), id = ID, corstr = "Exchangeable", data = OME)
estequa(fit7)
Description

Computes the estimating equations evaluated at the parameter estimates and the observed data for regression models based on the negative binomial, beta-binomial, and random-clumped binomial distributions, which are alternatives to the Poisson and binomial regression models under the presence of overdispersion.

Usage

## S3 method for class 'overglm'
estequa(object, ...)

Arguments

object an object of the class overglm.

... further arguments passed to or from other methods.

Value

A vector with the values of the estimating equations evaluated at the parameter estimates and the observed data.

Examples

### Example 1: Ability of retinyl acetate to prevent mammary cancer in rats
data(mammary)
fit1 <- overglm(tumors ~ group, family="nb1(identity)", data=mammary)
estequa(fit1)

### Example 2: Self diagnozed ear infections in swimmers
data(swimmers)
fit2 <- overglm(infections ~ frequency + location, family="nb1(log)", data=swimmers)
estequa(fit2)

### Example 3: Urinary tract infections in HIV-infected men
data(uti)
fit3 <- overglm(episodes ~ cd4 + offset(log(time)), family="nb1(log)", data = uti)
estequa(fit3)

### Example 4: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit4 <- overglm(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)
estequa(fit4)

### Example 5: Agents to stimulate cellular differentiation
Estequa.zeroinflation

Estimating Equations in Regression Models to deal with Zero-Excess in Count Data

Description

Computes the estimating equations evaluated at the parameter estimates and the observed data for regression models to deal with zero-excess in count data.

Usage

```r
## S3 method for class 'zeroinflation'
estequa(object, submodel = c("counts", "zeros"), ...)
```

Arguments

- **object**: an object of the class `zeroinflation`.
- **submodel**: an (optional) character string which allows to specify the model: "counts" or "zeros". By default, `submodel` is set to be "counts".
- ... further arguments passed to or from other methods.

Value

A vector with the values of the estimating equations evaluated at the parameter estimates and the observed data.
### Fabric faults

**Fabric faults**

#### Description
The main objective of the analysis of this dataset is to assess if there is an association between the number of faults in fabric rolls and their length.

#### Usage
```r
data(fabric)
```

#### Format
A data frame with 32 rows and 2 variables:
- **roll** a numeric vector indicating the length of the rolls.
- **faults** a numeric vector indicating the number of faults.

#### References
FisherScoring

Fisher Scoring algorithm in Generalized Linear Models

Description
This function displays the entire path performed by the Fisher Scoring algorithm for parameter estimation in Generalized Linear Models. It starts with the starting value until convergence is achieved or the maximum number of iterations is exceeded.

Usage
FisherScoring(object, verbose = TRUE, digits = 10)

Arguments
- object: one object of the class glm.
- verbose: an (optional) logical indicating if should the report of results be printed. By default, verbose is set to be TRUE.
- digits: an (optional) integer value indicating the number of decimal places to be used. By default, digits is set to be 10.

Value
a matrix whose first three columns are the following:
- Iteration: the iteration number,
- Deviance: value of the (unscaled) deviance computed using the current value of the parameter vector,
- Tolerance: value of \(\frac{|\text{deviance} - \text{deviance}_{\text{old}}|}{(\text{deviance}_{\text{old}} + 0.1)}\),

Examples

#### Example 1: Fuel efficiency of cars
Auto <- ISLR::Auto
fit1 <- glm(mpg ~ horsepower + weight + horsepower*weight, family=Gamma(inverse), data=Auto, control=list(trace=TRUE))
FisherScoring(fit1)

#### Example 2: Hill races in Scotland
data(races)
GHYC

Gosho-Hamada-Yoshimura’s Criterion for Generalized Estimating Equations

Description

Computes the Gosho-Hamada-Yoshimura’s criterion (GHYC) for one or more objects of the class glmgee.

Usage

GHYC(..., verbose = TRUE, digits = 3)

Arguments

... one or several objects of the class glmgee.
verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.
digits an (optional) integer indicating the number of digits to print.
Value

A data.frame with the values of the GHYC for each glmgee object in the input.

References


See Also

QIC, CIC, RJC, AGPC, SGPC

Examples

```
### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
GHYC(fit1, fit2, fit3, fit4)

### Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ visit + group
fit1 <- glmgee(mod2, id=subj, family=binomial(logit), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
GHYC(fit1, fit2, fit3, fit4)

### Example 3: Treatment for severe postnatal depression (2)
mod3 <- dep ~ visit*group
fit1 <- glmgee(mod3, id=subj, family=gaussian(identity), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Exchangeable")
GHYC(fit1, fit2, fit3)
```

 glmgee

*Fit Generalized Estimating Equations*

Description

Produces an object of the class glmgee in which the main results of a Generalized Estimating Equation (GEE) fitted to the data are stored.
Usage

```r
glmgee(
  formula,
  family = gaussian(),
  weights,
  id,
  waves,
  data,
  subset,
  corstr,
  corr,
  start = NULL,
  scale.fix = FALSE,
  scale.value = 1,
  toler = 1e-05,
  maxit = 50,
  trace = FALSE,
  ...
)
```

Arguments

- **formula**: a formula expression of the form `response ~ x1 + x2 + ...`, which is a symbolic description of the linear predictor of the model to be fitted to the data.
- **family**: an (optional) family object, that is, a list of functions and expressions for defining link and variance functions. Families (and links) supported are the same supported by `glm` using its `family` argument, that is, `gaussian`, `binomial`, `poisson`, `Gamma`, `inverse.gaussian`, and `quasi`. The family `negative.binomial` in the library `MASS` are also available. By default, the argument `family` is set to be `gaussian(identity)`.
- **weights**: an (optional) vector of positive "prior weights" to be used in the fitting process. The length of weights should be the same as the total number of observations.
- **id**: a vector which identifies the subjects or clusters. The length of `id` should be the same as the number of observations.
- **waves**: an (optional) positive integer-valued variable that is used to identify the order and spacing of observations within clusters. This argument is crucial when there are missing values and gaps in the data. By default, `waves` is equal to the integers from 1 to the size of each cluster.
- **data**: an (optional) data frame in which to look for variables involved in the formula expression, as well as for variables specified in the arguments `id` and `weights`. The data are assumed to be sorted by `id` and time.
- **subset**: an (optional) vector specifying a subset of observations to be used in the fitting process.
- **corstr**: an (optional) character string which allows to specify the working-correlation structure. The available options are: "Independence", "Unstructured","Stationary-M-dependent(m)", "Non-Stationary-M-dependent(m)", "AR-M-dependent(m)", etc.
"Exchangeable" and "User-defined", where \( m \) represents the lag of the dependence. By default, \( \text{corstr} \) is set to be "Independence".

\textbf{corr} an (optional) square matrix of the same dimension of the maximum cluster size containing the user specified correlation. This is only appropriate if \( \text{corstr} \) is specified to be "User-defined".

\textbf{start} an (optional) vector of starting values for the parameters in the linear predictor.

\textbf{scale.fix} an (optional) logical variable. If TRUE, the scale parameter is fixed at the value of \( \text{scale.value} \). By default, \( \text{scale.fix} \) is set to be FALSE.

\textbf{scale.value} an (optional) numeric value at which the scale parameter should be fixed. This is only appropriate if \( \text{scale.fix} = \text{TRUE} \). By default, \( \text{scale.value} \) is set to be 1.

\textbf{toler} an (optional) positive value which represents the convergence tolerance. The convergence is reached when the maximum of the absolute relative differences between the values of the parameters in the linear predictor in consecutive iterations of the fitting algorithm is lower than \( \text{toler} \). By default, \( \text{toler} \) is set to be 0.00001.

\textbf{maxit} an (optional) integer value which represents the maximum number of iterations allowed for the fitting algorithm. By default, \( \text{maxit} \) is set to be 50.

\textbf{trace} an (optional) logical variable. If TRUE, output is produced for each iteration of the estimating algorithm.

... further arguments passed to or from other methods.

\section*{Details}

The values of the multivariate response variable measured on \( n \) subjects or clusters, denoted by \( y_i = (y_{i1}, \ldots, y_{in_i})^T \) for \( i = 1, \ldots, n \), are assumed to be realizations of independent random vectors denoted by \( Y_i = (Y_{i1}, \ldots, Y_{in_i})^T \) for \( i = 1, \ldots, n \). The random variables associated to the \( i \)-th subject or cluster, \( Y_{ij} \) for \( j = 1, \ldots, n_i \), are assumed to satisfy \( \mu_{ij} = E(Y_{ij}), \text{Var}(Y_{ij}) = \phi \omega_{ij} V(\mu_{ij}) \) and \( \text{Corr}(Y_{ij}, Y_{ik}) = r_{jk}(\rho) \), where \( \phi > 0 \) is the dispersion parameter, \( V(\mu_{ij}) \) is the variance function, \( \omega_{ij} > 0 \) is a known weight, and \( \rho = (\rho_1, \ldots, \rho_q)^T \) is a parameter vector. In addition, \( \mu_{ij} \) is assumed to be dependent on the regressors vector \( x_{ij} \) by \( g(\mu_{ij}) = z_{ij} + x_{ij}^T \beta \), where \( g(\cdot) \) is the link function, \( z_{ij} \) is a known offset and \( \beta = (\beta_1, \ldots, \beta_p)^T \) is a vector of regression parameters. The parameter estimates are obtained by iteratively solving the estimating equations described by Liang and Zeger (1986).

If the maximum cluster size is 6 and for a cluster of size 4 the value of \textit{waves} is set to be 2, 4, 5, 6, then it means that the data at times 1 and 3 are missing, which should be taken into account by \texttt{glmgee} when the structure of the correlation matrix is assumed to be "Unstructured", "Stationary-M-dependent", "Non-Stationary-M-dependent" or "AR-M-dependent". If in this scenario \textit{waves} is not specified then \texttt{glmgee} assumes that the available data for this cluster were taken at times 1, 2, 3 and 4.

A set of standard extractor functions for fitted model objects is available for objects of class \texttt{glmgee}, including methods to generic functions such as \texttt{print}, \texttt{summary}, \texttt{model.matrix}, \texttt{estequa}, \texttt{coef}, \texttt{vcov}, \texttt{logLik}, \texttt{fitted}, \texttt{confint} and \texttt{predict}. In addition, the model may be assessed using functions such as \texttt{anova.glmgee}, \texttt{residuals.glmgee}, \texttt{dfbeta.glmgee}, \texttt{cooks.distance.glmgee} and \texttt{localInfluence.glmgee}. The variable selection may be accomplished using the routine \texttt{stepCriterion.glmgee}. 

...
**Value**

An object of class `glmgee` in which the main results of the GEE model fitted to the data are stored, i.e., a list with components including:

- `coefficients`: a vector with the estimates of $\beta_1, \ldots, \beta_p$,
- `fitted.values`: a vector with the estimates of $\mu_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- `start`: a vector with the starting values used,
- `iter`: a numeric constant with the number of iterations,
- `prior.weights`: a vector with the values of $\omega_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- `offset`: a vector with the values of $z_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- `terms`: an object containing the terms objects,
- `loglik`: the value of the quasi-log-likelihood function evaluated at the parameter estimates and the observed data,
- `estfun`: a vector with the estimating equations evaluated at the parameter estimates and the observed data,
- `formula`: the formula,
- `levels`: the levels of the categorical regressors,
- `contrasts`: an object containing the contrasts corresponding to levels,
- `converged`: a logical indicating successful convergence,
- `model`: the full model frame,
- `y`: a vector with the values of $y_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- `family`: an object containing the `family` object used,
- `linear.predictors`: a vector with the estimates of $g(\mu_{ij})$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- `R`: a matrix with the (robust) estimate of the variance-covariance,
- `corr`: a matrix with the estimate of the working-correlation,
- `corstr`: a character string specifying the working-correlation structure,
- `id`: a vector which identifies the subjects or clusters,
sizes a vector with the values of $n_i$ for $i = 1, ..., n$.

call the original function call.

References


See Also

`gnmgee, wglmgee`

Examples

```
# Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), corstr="AR-M-dependent(1)", data=spruces)
summary(fit1, corr.digits=2)

# Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ visit + group
fit2 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent(1)", data=depression)
summary(fit2, corr.digits=2)

# Example 3: Treatment for severe postnatal depression (2)
mod3 <- dep ~ visit*group
fit3 <- glmgee(mod3, id=subj, family=gaussian, corstr="AR-M-dependent(1)", data=depression)
summary(fit3, corr.digits=2)

# Example 4: Dental Clinical Trial
data(rinse)
mod4 <- score/3.6 ~ rinse*time
fit4 <- glmgee(mod4, family=binomial(log), id=subject, corstr="Exchangeable", data=rinse)
summary(fit4, corr.digits=2)

# Example 5: Shoulder Pain after Laparoscopic Cholecystectomy
data(cholecystectomy)
mod5 <- pain2 ~ treatment + age + time
corstr <- "Stationary-M-dependent(2)"
fit5 <- glmgee(mod5, family=binomial(logit), id=id, corstr=corstr, data=cholecystectomy)
summary(fit5, varest="bias-corrected")

# Example 6: Guidelines for Urinary Incontinence Discussion and Evaluation
```
data(GUIDE)
mod6 <- bothered ~ gender + age + dayacc + severe + toilet
fit6 <- glmgee(mod6, family=binomial(logit), id=practice, corstr="Exchangeable", data=GUIDE)
summary(fit6)

##### Example 7: Tests of Auditory Perception in Children with OME
OME <- MASS::OME
mod7 <- cbind(Correct, Trials-Correct) ~ Loud + Age + OME
fit7 <- glmgee(mod7, family = binomial(cloglog), id = ID, corstr = "Exchangeable", data = OME)
summary(fit7, corr=FALSE)

##### Example 8: Epileptic seizures
data(Seizures)
Seizures2 <- within(Seizures, time4 <- ifelse(time==4,1,0))
mod8 <- seizures ~ log(age) + time4 + log(base/4)*treatment
fit8 <- glmgee(mod8, family=poisson(log), id=id, corstr="Exchangeable", data=Seizures2)
summary(fit8)

---

**gnmgee**  
*Fit Nonlinear Generalized Estimating Equations*

**Description**

Produces an object of the class `glmgee` in which the main results of a Nonlinear Generalized Estimating Equation (GEE) fitted to the data are stored.

**Usage**

```r
gnmgee(
  formula,
  family = gaussian(),
  weights = NULL,
  id,
  waves,
  data,
  subset = NULL,
  corstr,
  corr,
  start = NULL,
  scale.fix = FALSE,
  scale.value = 1,
  toler = 1e-05,
  maxit = 50,
  trace = FALSE,
  ...
)
```
Arguments

formula
a nonlinear model formula including variables and parameters, which is a symbolic description of the nonlinear predictor of the model to be fitted to the data.

family
an (optional) family object, that is, a list of functions and expressions for defining link and variance functions. Families (and links) supported are the same supported by glm using its family argument, that is, gaussian, binomial, poisson, Gamma, inverse.gaussian, and quasi. The family negative.binomial in the library MASS are also available. By default, the argument family is set to be gaussian(identity).

weights
an (optional) vector of positive "prior weights" to be used in the fitting process. The length of weights should be the same as the total number of observations.

id
a vector which identifies the subjects or clusters. The length of id should be the same as the number of observations.

waves
an (optional) positive integer-valued variable that is used to identify the order and spacing of observations within clusters. This argument is crucial when there are missing values and gaps in the data. By default, waves is equal to the integers from 1 to the size of each cluster.

data
an (optional) data frame in which to look for variables involved in the formula expression, as well as for variables specified in the arguments id and weights. The data are assumed to be sorted by id and time.

subset
an (optional) vector specifying a subset of observations to be used in the fitting process.

corstr
an (optional) character string which allows to specify the working-correlation structure. The available options are: "Independence", "Unstructured", "Stationary-M-dependent(m)", "Non-Stationary-M-dependent(m)", "AR-M-dependent(m)", "Exchangeable" and "User-defined", where m represents the lag of the dependence. By default, corstr is set to be "Independence".

corr
an (optional) square matrix of the same dimension of the maximum cluster size containing the user specified correlation. This is only appropriate if corstr is specified to be "User-defined".

start
an (optional) vector of starting values for the parameters in the nonlinear predictor. When start is missing (and formula is not a self-starting model, see nls and selfStart), a very cheap guess for start is tried.

scale.fix
an (optional) logical variable. If TRUE, the scale parameter is fixed at the value of scale.value. By default, scale.fix is set to be FALSE.

scale.value
an (optional) numeric value at which the scale parameter should be fixed. This is only appropriate if scale.fix=TRUE. By default, scale.value is set to be 1.

toler
an (optional) positive value which represents the convergence tolerance. The convergence is reached when the maximum of the absolute relative differences between the values of the parameters in the nonlinear predictor in consecutive iterations of the fitting algorithm is lower than toler. By default, toler is set to be 0.00001.

maxit
an (optional) integer value which represents the maximum number of iterations allowed for the fitting algorithm. By default, maxit is set to be 50.
trace an (optional) logical variable. If TRUE, output is produced for each iteration of the estimating algorithm.

... further arguments passed to or from other methods.

Details

The values of the multivariate response variable measured on \( n \) subjects or clusters, denoted by \( y_i = (y_{i1}, \ldots, y_{in_i})^T \) for \( i = 1, \ldots, n \), are assumed to be realizations of independent random vectors denoted by \( Y_i = (Y_{i1}, \ldots, Y_{in_i})^T \) for \( i = 1, \ldots, n \). The random variables associated to the \( i \)-th subject or cluster, \( Y_{ij} \) for \( j = 1, \ldots, n_i \), are assumed to satisfy \( \mu_{ij} = \mathbb{E}(Y_{ij}), \text{Var}(Y_{ij}) = \frac{\phi}{\omega_{ij}} V(\mu_{ij}) \) and \( \text{Corr}(Y_{ij}, Y_{ik}) = r_{jk}(\rho) \), where \( \phi > 0 \) is the dispersion parameter, \( V(\mu_{ij}) \) is the variance function, \( \omega_{ij} > 0 \) is a known weight, and \( \rho = (\rho_1, \ldots, \rho_q)^T \) is a parameter vector. In addition, \( \mu_{ij} \) is assumed to be dependent on the regressors vector \( x_{ij} \) by \( g(\mu_{ij}) = z_{ij} + m(x_{ij}, \beta) \), where \( g(\cdot) \) is the link function, \( z_{ij} \) is a known offset, \( \beta = (\beta_1, \ldots, \beta_p)^T \) is a vector of regression parameters and \( m(x_{ij}, \beta) \) is a known nonlinear function of \( \beta \). The parameter estimates are obtained by iteratively solving the estimating equations described by Liang and Zeger (1986).

If the maximum cluster size is 6 and for a cluster of size 4 the value of waves is set to be 2, 4, 5, 6, then it means that the data at times 1 and 3 are missing, which should be taken into account by gnmgee when the structure of the correlation matrix is assumed to be "Unstructured", "Stationary-M-dependent", "Non-Stationary-M-dependent" or "AR-M-dependent". If in this scenario waves is not specified then gnmgee assumes that the available data for this cluster were taken at times 1, 2, 3 and 4.

A set of standard extractor functions for fitted model objects is available for objects of class \texttt{glmgee}, including methods to generic functions such as \texttt{print}, \texttt{summary}, \texttt{model.matrix}, \texttt{estequa}, \texttt{coef}, \texttt{vcov}, \texttt{logLik}, \texttt{fitted}, \texttt{confint} and \texttt{predict}. In addition, the model may be assessed using functions such as \texttt{anova.glmgee}, \texttt{residuals.glmgee}, \texttt{dfbeta.glmgee} and \texttt{cooks.distance.glmgee}.

Value

an object of class \texttt{glmgee} in which the main results of the GEE model fitted to the data are stored, i.e., a list with components including

- \texttt{coefficients} a vector with the estimates of \( \beta_1, \ldots, \beta_p \),
- \texttt{fitted.values} a vector with the estimates of \( \mu_{ij} \) for \( i = 1, \ldots, n \) and \( j = 1, \ldots, n_i \),
- \texttt{start} a vector with the starting values used,
- \texttt{iter} a numeric constant with the number of iterations,
- \texttt{prior.weights} a vector with the values of \( \omega_{ij} \) for \( i = 1, \ldots, n \) and \( j = 1, \ldots, n_i \),
- \texttt{offset} a vector with the values of \( z_{ij} \) for \( i = 1, \ldots, n \) and \( j = 1, \ldots, n_i \),
- \texttt{terms} an object containing the terms objects,
- \texttt{loglik} the value of the quasi-log-likelihood function evaluated at the parameter estimates and the observed data,
estfun a vector with the estimating equations evaluated at the parameter estimates and the observed data,

formula the formula,

levels the levels of the categorical regressors,

contrasts an object containing the contrasts corresponding to levels,

converged a logical indicating successful convergence,

model the full model frame,

y a vector with the values of $y_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,

family an object containing the family object used,

linear.predictors a vector with the estimates of $g(\mu_{ij})$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,

R a matrix with the (robust) estimate of the variance-covariance,

corr a matrix with the estimate of the working-correlation,

corstr a character string specifying the working-correlation structure,

id a vector which identifies the subjects or clusters,

sizes a vector with the values of $n_i$ for $i = 1, \ldots, n$,

call the original function call,

References


See Also

`glmgee, wglmgee`

Examples

```r
### Example : Orange trees grown at Riverside, California
data(Oranges)
```
mod <- Trunk ~ b1/(1 + exp((b2-Days)/b3))
start <- c(b1=200, b2=760, b3=375)
fit1 <- gnmgee(mod, start=start, id=Tree, family=Gamma(identity), corstr="Exchangeable",
data=Oranges)
summary(fit1, corr.digits=2)

mod <- Trunk ~ SSlogis(Days,b1,b2,b3)
fit2 <- gnmgee(mod, id=Tree, family=Gamma(identity), corstr="Exchangeable", data=Oranges)
summary(fit2, corr.digits=2)

GUIDE

Guidelines for Urinary Incontinence Discussion and Evaluation

Description

These data arose from a randomized controlled trial that assessed if provider adherence to a set of guidelines for treatment of patients with urinary incontinence (UI) affected patient outcomes. Data were collected on 137 elderly patients from 38 medical practices. The number of patients per practice ranged from 1 to 8 and the median was 4 patients. The statistical analysis aims to determine what predicts whether or not a patient considers their UI a problem that interferes with him/her daily life.

Usage

data(GUIDE)

Format

A data frame with 137 rows and 7 variables:

- **bothered** a numeric vector giving the answer to the following: Do you consider this accidental loss of urine a problem that interferes with your day to day activities or bothers you in other ways? 1 for "Yes" and 0 for "No".
- **gender** a factor giving the patient’s gender: "Male" or "Female".
- **age** a numeric vector giving the standardized age: (age in years - 76)/10.
- **dayacc** a numeric vector giving the patient’s report of the number of leaking accidents they experience in an average day (derived from number of accidents reported per week).
- **severe** a factor giving the severity of the loss of urine: "1" if there is only some moisture; "2" if the patient wet the underwear; "3" if the urine trickled down the thigh; and "4" if the patient wet the floor.
- **toilet** a numeric vector giving the patient’s report on the number of times during the day he (or she) usually go to the toilet to urinate.
- **practice** a character string giving the identifier of the medical practice.

Source

http://www.bios.unc.edu/~preisser/personal/uidata/prefaq99.dat
References

Examples
```r
data(GUIDE)
mod <- bothered ~ gender + age + dayacc + severe + toilet
fit <- glmgee(mod, family=binomial(logit), id=practice, corstr="Exchangeable", data=GUIDE)
summary(fit)
```

---

**gvif**  
*Generalized Variance Inflation Factor*

**Description**
Computes the generalized variance inflation factor (GVIF) for a fitted model object.

**Usage**
gvif(model, ...)

**Arguments**
- model: a fitted model object.
- ...: further arguments passed to or from other methods.

**Value**
An object with the values of the GVIF for all effects in the model.

---

**gvif.glm**  
*Generalized Variance Inflation Factor*

**Description**
Computes the generalized variance inflation factor (GVIF) for a generalized linear model.

**Usage**
```r
## S3 method for class 'glm'
gvif(model, verbose = TRUE, ...)
```
Arguments

model  an object of the class \textit{glm}.
verbose  an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.

Details

If the number of degrees of freedom is 1 then the GVIF reduces to the Variance Inflation Factor (VIF).

Value

A matrix with so many rows as effects in the model and the following columns:

- \text{GVIF}  the values of GVIF,
- \text{df}  the number of degrees of freedom,
- \text{GVIF}^{1/(2*df)}  the values of GVIF^{1/2df},

References


See Also

gvif.lm

Examples

##### Example 1: Fuel consumption of automobiles
Auto <- ISLR::Auto
Auto2 <- within(Auto, origin <- factor(origin))
mod <- mpg ~ cylinders + displacement + acceleration + origin + horsepower*weight
fit1 <- glm(mod, family=inverse.gaussian("log"), data=Auto2)
gvif(fit1)

##### Example 2: Patients with burn injuries
burn1000 <- aplore3::burn1000
burn1000 <- within(burn1000, death <- factor(death, levels=c("Dead","Alive")))
mod2 <- death ~ gender + race + flame + age*inh_inj + tbsa*inh_inj
fit2 <- glm(mod2, family=binomial("logit"), data=burn1000)
gvif(fit2)

##### Example 3: Hill races in Scotland
data(races)
fit3 <- glm(rtime ~ log(distance) + cclimb, family=Gamma(log), data=races)
gvif(fit3)
Description

Computes the generalized variance inflation factor (GVIF) for a weighted or unweighted normal linear model.

Usage

## S3 method for class 'lm`
`gvif(model, verbose = TRUE, ...)

Arguments

- `model`: an object of the class `lm`.
- `verbose`: an (optional) logical switch indicating if should the report of results be printed. By default, `verbose` is set to be `TRUE`.
- `...`: further arguments passed to or from other methods.

Details

If the number of degrees of freedom is 1 then the GVIF reduces to the Variance Inflation Factor (VIF).

Value

A matrix with so many rows as effects in the model and the following columns:

- `GVIF`: the values of GVIF,
- `df`: the number of degrees of freedom,
- `GVIF^{(1/(2*df))}`: the values of GVIF^{1/(2*df)},

References


See Also

gvif.glm
Examples

##### Example 1: New York air quality measurements
fit1 <- lm(log(Ozone) ~ Solar.R + Temp + Wind, data = airquality)
gvif(fit1)

##### Example 2: Fuel consumption of automobiles
fit2 <- lm(mpg ~ log(hp) + log(wt) + qsec, data = mtcars)
gvif(fit2)

##### Example 3: Credit card balance
Credit <- ISLR::Credit
fit3 <- lm(Balance ~ Cards + Age + Rating + Income + Student + Limit, data = Credit)
gvif(fit3)

---

gvif.overglm

Generalized Variance Inflation Factor for alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion

Description

Computes the generalized variance inflation factor (GVIF) for regression models based on the negative binomial, beta-binomial, and random-clumped binomial distributions, which are alternatives to the Poisson and binomial regression models under the presence of overdispersion. The GVIF is aimed to identify collinearity problems.

Usage

## S3 method for class 'overglm'
gvif(model, verbose = TRUE, ...)

Arguments

model an object of class overglm.
verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.
... further arguments passed to or from other methods.

Details

If the number of degrees of freedom is 1 then the GVIF reduces to the Variance Inflation Factor (VIF).
Value

A matrix with so many rows as effects in the model and the following columns:

- GVIF: the values of GVIF,
- df: the number of degrees of freedom,
- GVIF^{1/(2*df)}: the values of GVIF^{1/2df}.

References


See Also

gvif.lm, gvif.glm

Examples

```r
# Example 1: Self diagnosed ear infections in swimmers
data(swimmers)
fit1 <- overglm(infections ~ frequency + location, family="nb1(log)", data=swimmers)
gvif(fit1)

# Example 2: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit2 <- overglm(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)
gvif(fit2)

# Example 3: Agents to stimulate cellular differentiation
data(cellular)
fit3 <- overglm(cbind(cells,200-cells) ~ tnf + ifn, family="bb(logit)", data=cellular)
gvif(fit3)
```

---

**hltest** *The Hosmer-Lemeshow Goodness-of-Fit Test*

Description

Computes the Hosmer-Lemeshow goodness-of-fit test for a generalized linear model fitted to binary responses.

Usage

```r
hltest(model, verbose = TRUE, ...)
```
Arguments

model an object of the class glm, which is obtained from the fit of a generalized linear model where the distribution for the response variable is assumed to be binomial.

verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.

... further arguments passed to or from other methods.

Value

A matrix with the following four columns:

hm a matrix with the values of Group, Size, Observed and Expected, which are required to compute the statistic of the test,

statistic the value of the statistic of the test,

df the number of degrees of freedom, given by the number of groups minus 2,

p.value the p-value of the test computed using the Chi-square distribution,

References


Examples

####### Example 1: Patients with burn injuries
burn1000 <- aplore3::burn1000
burn1000 <- within(burn1000, death <- factor(death, levels=c("Dead","Alive")))
fit1 <- glm(death ~ age*inh_inj + tbsa*inh_inj, family=binomial("logit"), data=burn1000)
hltest(fit1)

####### Example 2: Bladder cancer in mice
data(bladder)
fit2 <- glm(cancer/exposed ~ dose, weights=exposed, family=binomial("cloglog"), data=bladder)
hltest(fit2)

####### Example 3: Liver cancer in mice
data(liver)
fit3 <- glm(cancer/exposed ~ dose, weights=exposed, family=binomial("probit"), data=liver)
hltest(fit3)
Description

The data consists of the proportion of lactic dehydrogenase enzyme leakage obtained as a response of hepatocyte cell toxicity to the effects of different combinations of carbon tetrachloride (CCl4) and chloroform (CHCl3). Thus, the main objective of the data analysis is to evaluate the effects of CCl4, CHCl3 and their interactions on the response.

Usage

data(ldh)

Format

A data frame with 448 rows and 5 variables:

- **LDH** a numeric vector indicating the proportion of lactic dehydrogenase enzyme leakage, a surrogate for cell toxicity.
- **CCl4** a numeric vector indicating the carbon tetrachloride at 0, 1, 2.5 and 5 mM.
- **CHCl3** a numeric vector indicating the chloroform at 0, 5, 10 and 25 mM.
- **Flask** a numeric vector indicating the flask of isolated hepatocyte suspensions.
- **Time** a numeric vector indicating the time at 0, 0.01, 0.25, 0.50, 1, 2 and 3 hours.

Source


References


Examples

data(ldh)
opt <- unique(ldh$CCl4)
dev.new()
par(mfrow=c(1,length(opt)))
for(i in 1:length(opt))
  boxplot(LDH ~ Time, data=subset(ldh,CCl4==opt[i]), ylim=c(0,0.8), main=paste("CCl4=",opt[i]))
dev.new()
opt <- unique(ldh$CHCl3)
par(mfrow=c(1,length(opt)))
for(i in 1:length(opt))
leverage

boxplot(LDH ~ Time, data=subset(ldh,CHCl3==opt[i]), ylim=c(0,0.8), main=paste("CHCl3="opt[i]))

---

leverage  Leverage

**Description**

Computes leverage measures for a fitted model object.

**Usage**

`leverage(object, ...)`

**Arguments**

- `object` a fitted model object.
- `...` further arguments passed to or from other methods.

**Value**

An object with the values of the leverage measures.

---

leverage.glmeeg  Leverage for Generalized Estimating Equations

**Description**

Computes and, optionally, displays a graph of the leverage measures at the cluster- and observation-level.

**Usage**

```r
## S3 method for class 'glmeeg'
leverage(
  object,
  level = c("clusters", "observations"),
  plot.it = FALSE,
  identify,
  ...)
```
Arguments

object an object of class glmgee.
level an (optional) character string indicating the level for which the leverage measures are required. The options are: cluster-level ("clusters") and observation-level ("observations"). By default, level is set to be "clusters".
plot.it an (optional) logical indicating if the plot of the measures of leverage are required or just the data matrix in which that plot is based. By default, plot.it is set to be FALSE.
identify an (optional) integer indicating the number of (level="clusters") or observations (level="observations") to identify on the plot of the leverage measures. This is only appropriate if plot.it is specified to be TRUE.

Value

A vector with the values of the leverage measures with so many rows as clusters (level="clusters") or observations (level="observations") in the sample.

References


Examples

```
Example 1: Tests of Auditory Perception in Children with OME
OME <- MASS::OME
mod <- cbind(Correct, Trials-Correct) ~ Loud + Age + OME
fit1 <- glmgee(mod, family = binomial(cloglog), id = ID, corstr = "Exchangeable", data = OME)
leverage(fit1, level="clusters", plot.it=TRUE)

Example 2: Guidelines for Urinary Incontinence Discussion and Evaluation
data(GUIDE)
mod <- bothered ~ gender + age + dayacc + severe + toilet
fit2 <- glmgee(mod, family=binomial(logit), id=practice, corstr="Exchangeable", data=GUIDE)
leverage(fit2, level="clusters", plot.it=TRUE)
leverage(fit2, level="observations", plot.it=TRUE)
```
Liver cancer in mice

Description

Female mice were continuously fed dietary concentrations of 2-Acetylaminofluorene (2-AAF), a carcinogenic and mutagenic derivative of fluorene. Serially sacrificed, dead or moribund mice were examined for tumors and deaths dates recorded. These data consist of the incidences of liver neoplasms in mice observed during 18 months.

Usage

data(liver)

Format

A data frame with 8 rows and 3 variables:

dose a numeric vector giving the dose, in parts per $10^4$, of 2-AAF.
exposed a numeric vector giving the number of mice exposed to each dose of 2-AAF.
cancer a numeric vector giving the number of mice with liver cancer for each dose of 2-AAF.

References


See Also

bladder

Examples

data(liver)
dev.new()
barplot(100*cancer/exposed ~ dose, beside=TRUE, data=liver, col="red",
       xlab="Dose of 2-AAF", ylab="% of mice with liver cancer")
Description
Computes measures of local influence for a fitted model object.

Usage
localInfluence(object, ...)

Arguments
- object: a fitted model object.
- ...: further arguments passed to or from other methods.

Value
An object with the measures of local influence.

Description
Computes some measures and, optionally, display graphs of them to perform influence analysis based on the approaches described in Cook (1986).

Usage
```r
## S3 method for class 'glm'
localInfluence(
  object,
  type = c("total", "local"),
  perturbation = c("case-weight", "response", "covariate"),
  covariate,
  coefs,
  plot.it = FALSE,
  identify,
  ...
)
```
Arguments

- **object**: an object of class `glm`.
- **type**: an (optional) character string indicating the type of approach to study the local influence. The options are: the absolute value of the elements of the eigenvector which corresponds to the maximum absolute eigenvalue ("local"); and the absolute value of the elements of the main diagonal ("total"). By default, type is set to be "total".
- **perturbation**: an (optional) character string indicating the perturbation scheme to apply. The options are: case weight perturbation of observations ("case-weight"); perturbation of covariates ("covariate"); and perturbation of response ("response"). By default, perturbation is set to be "case-weight".
- **covariate**: an character string which (partially) match with the names of one of the parameters in the linear predictor. This is only appropriate if perturbation="covariate".
- **coefs**: an (optional) character string which (partially) match with the names of some of the parameters in the linear predictor.
- **plot.it**: an (optional) logical indicating if the plot of the measures of local influence is required or just the data matrix in which that plot is based. By default, plot.it is set to be FALSE.
- **identify**: an (optional) integer indicating the number of observations to identify on the plot of the measures of local influence. This is only appropriate if plot.it=TRUE.
- **...**: further arguments passed to or from other methods. If plot.it=TRUE then ... may be used to include graphical parameters to customize the plot. For example, col, pch, cex, main, sub, xlab, ylab.

Value

A matrix as many rows as observations in the sample and one column with the values of the measures of local influence.

References


---

**localInfluence.glmgee** *Local Influence for Generalized Estimating Equations*

Description

Computes some measures and, optionally, display graphs of them to perform influence analysis based on the approaches described in Cook (1986) and Jung (2008).
Usage

## S3 method for class 'glmgee'
localInfluence(
  object,
  type = c("total", "local"),
  perturbation = c("cw-clusters", "cw-observations", "response"),
  coefs,
  plot.it = FALSE,
  identify,
  ...
)

Arguments

object an object of class glmgee.

type an (optional) character string indicating the type of approach to study the local influence. The options are: the absolute value of the elements of the eigenvector which corresponds to the maximum absolute eigenvalue ("local"); and the elements of the main diagonal ("total"). By default, type is set to be "total".

perturbation an (optional) character string indicating the perturbation scheme to apply. The options are: case weight perturbation of clusters ("cw-clusters"); Case weight perturbation of observations ("cw-observations"); and perturbation of response ("response"). By default, perturbation is set to be "cw-clusters".

coefs an (optional) character string which (partially) match with the names of some of the parameters in the linear predictor.

plot.it an (optional) logical indicating if the plot of the measures of local influence is required or just the data matrix in which that plot is based. By default, plot.it is set to be FALSE.

identify an (optional) integer indicating the number of clusters/observations to identify on the plot of the measures of local influence. This is only appropriate if plot.it=TRUE.

... further arguments passed to or from other methods. If plot.it=TRUE then ... may be used to include graphical parameters to customize the plot. For example, col, pch, cex, main, sub, xlab, ylab.

Value

A matrix as many rows as clusters/observations in the sample and one column with the values of the measures of local influence.

References


**localInfluence.overglm**

*Local Influence for alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion*

**Examples**

```
Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), corstr="AR-M-dependent", data=spruces)
localInfluence(fit1,type="total",perturbation="cw-clusters",coefs="treat",plot.it=TRUE)
```

```
Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ visit + group
fit2 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent", data=depression)
localInfluence(fit2,type="total",perturbation="cw-clusters",coefs="group",plot.it=TRUE)
```

```
Example 3: Treatment for severe postnatal depression (2)
mod3 <- dep ~ visit*group
fit3 <- glmgee(mod3, id=subj, family=gaussian(identity), corstr="AR-M-dependent", data=depression)
localInfluence(fit3,type="total",perturbation="cw-clusters",coefs="visit:group",plot.it=TRUE)
```

**localInfluence.overglm**

*Local Influence for alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion*

**Description**

Computes local influence measures under the case-weight perturbation scheme for alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion. Those local influence measures may be chosen to correspond to all parameters in the linear predictor or (via coefs) for just some subset of them.

**Usage**

```r
## S3 method for class 'overglm'
localInfluence(
  object, 
  type = c("total", "local"), 
  coefs, 
  plot.it = FALSE, 
  identify, 
  ...
)
```

**Arguments**

- **object**: an object of class `overglm`.  

---
localInfluence.overglm

**type**
an (optional) character string which allows to specify the local influence approach: the absolute value of the elements of the main diagonal of the normal curvature matrix ("total") or the eigenvector which corresponds to the maximum absolute eigenvalue of the normal curvature matrix ("local"). By default, **type** is set to be "total".

**coefs**
an (optional) character string which (partially) match with the names of some model parameters.

**plot.it**
an (optional) logical indicating if the plot is required or just the data matrix in which that plot is based. By default, **plot.it** is set to be FALSE.

**identify**
an (optional) integer indicating the number of individuals to identify on the plot. This is only appropriate if **plot.it**=TRUE.

... further arguments passed to or from other methods. If **plot.it**=TRUE then ... may be used to include graphical parameters to customize the plot. For example, **col**, **pch**, **cex**, **main**, **sub**, **xlab**, **ylab**.

**Value**
A matrix as many rows as individuals in the sample and one column with the values of the local influence measure.

**References**

**Examples**

```
#### Example 1: Self diagnosed ear infections in swimmers
data(swimmers)
fit1 <- overglm(infections ~ frequency + location, family="nb1(log)", data=swimmers)

### Local influence for all parameters in the linear predictor
localInfluence(fit1, type="local", plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)

### Local influence for the parameter associated with 'frequency'
localInfluence(fit1, type="local", plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", coef="frequency", col.axis="blue", col.main="black", family="mono", cex=0.8)

#### Example 2: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit2 <- overglm(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)

### Local influence for all parameters in the linear predictor
localInfluence(fit2, type="local", plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)

### Local influence for the parameter associated with 'fem'
localInfluence(fit2, type="local", plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", coef="fem", col.axis="blue", col.main="black", family="mono", cex=0.8)
```
Example 3: Agents to stimulate cellular differentiation

data(cellular)
fit3 <- overglm(cbind(cells, 200-cells) ~ tnf + ifn, family="bb(logit)", data=cellular)

### Local influence for all parameters in the linear predictor
localInfluence(fit3, type="local", plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)

### Local influence for the parameter associated with 'tnf'
localInfluence(fit3, type="local", plot.it=TRUE, col="red", lty=1, lwd=1, col.lab="blue", coef="tnf", col.axis="blue", col.main="black", family="mono", cex=0.8)

---

**Ability of retinyl acetate to prevent mammary cancer in rats**

**Description**

A total of 76 female rats were injected with a carcinogen for mammary cancer. Then, all animals were given retinyl acetate (retinoid) to prevent mammary cancer for 60 days. After this phase, the 48 animals that remained tumor-free were randomly assigned to continue retinoid prophylaxis or control. Rats were then palpated for tumors twice weekly, and observations ended 182 days after initial carcinogen injections began. The main objective of the analysis was to assess the difference in tumor development between the treated and control groups.

**Usage**

data(mammary)

**Format**

A data frame with 48 rows and 2 variables:

- **group** a factor giving the group to which the rat was assigned: "retinoid" or "control".
- **tumors** a numeric vector giving the number of tumors identified on the rat.

**References**


**Examples**

data(mammary)
dev.new()
boxplot(tumors ~ group, data=mammary, outline=FALSE, xlab="Group", ylab="Number of tumors", col=c("yellow","blue"))
Oranges

---

**Description**

The data arose from five orange trees grown in Riverside, California, during 1969-1973. The response is the trunk circumference, in millimeters, and the predictor variable is time, in days. The predictor variable has an arbitrary origin and was taken on December 31, 1968.

**Usage**

```r
data(Oranges)
```

**Format**

A data frame with 35 rows and 3 variables:

- **Trunk** a numeric vector indicating the trunk circumference, in millimeters.
- **Days** a numeric vector indicating the time, in days, since December 31, 1968.
- **Tree** a numeric vector with the identifier of each orange tree.

**References**


**Examples**

```r
dev.new()
data(Oranges)
with(Oranges, plot(Days, Trunk, pch=16, col="blue"))
```

---

orobanche

---

**Description**

These data arose from a study of the germination of two species of Orobanche seeds (O. aegyptiaca 75 and O. aegyptiaca 73) grown on 1/125 dilutions of two different root extract media (cucumber and bean) in a 2x2 factorial layout with replicates. The data consist of the number of seeds and germinating seeds for each replicate. Interest is focused on the possible differences in germination rates for the two types of seed and root extract and whether there is any interaction.

**Usage**

```r
data(orobanche)
```
ossification

Format

A data frame with 21 rows and 4 variables:

- **specie** a factor indicating the specie of Orobanche seed: O. aegyptiaca 75 ("Aegyptiaca 75") and O. aegyptiaca 73 ("Aegyptiaca 73").
- **extract** a factor indicating the root extract: cucumber ("Cucumber") and bean ("Bean").
- **seeds** a numeric vector indicating the total number of seeds.
- **germinated** a numeric vector indicating the number of germinated seeds.

References


Examples

data(orobanche)
out <- aggregate(cbind(germinated,seeds) ~ extract + specie, data=orobanche, sum)
dev.new()
barplot(100*germinated/seeds ~ extract + specie, beside=TRUE, data=out, width=0.3,
col=c("yellow","blue"), xlab="Specie", ylab="% of germinated seeds")
legend("topleft",c("Bean","Cucumber"),fill=c("yellow","blue"),title="Extract",bty="n")

ossification

Teratogenic effects of phenytoin and trichloropropene oxide

Description

The data come from a 2x2 factorial design with 81 pregnant mice. In the experiment each pregnant mouse was randomly allocated to a control group and three treated groups. These groups received daily, by gastric gavage, 60 mg/kg of phenytoin, 100 mg/kg of trichloropropene oxide, or 60 mg/kg of phenytoin and 100 mg/kg of trichloropropene oxide. On day 18 of gestation, the fetuses were recovered, stained, and cleared. Then, by visual inspection, the presence or absence of ossification was determined for the different joints of the right and left forepaws. The experiment investigated the synergy of phenytoin and trichloropropene oxide to produce ossification at the phalanges, teratogenic effects.

Usage

data(ossification)
Format

A data frame with 81 rows and 4 variables:

- **fetuses** a numeric vector giving the number of fetuses showing ossification on the left middle third phalanx.
- **litter** a numeric vector giving the litter size.
- **pht** a factor giving the dose (mg/kg) of phenytoin: "0 mg/kg" or "60 mg/kg".
- **tcpo** a factor giving the dose (mg/kg) of trichloropropene oxide: "0 mg/kg" or "100 mg/kg".

References


Examples

```r
data(ossification)
dev.new()
boxplot(100*fetuses/litter ~ pht, data=subset(ossification,tcpo=="0 mg/kg"),
   at=c(1:2) - 0.2, col="yellow", boxwex=0.25, xaxt="n",
   xlab="Dose of PHT", ylab="% of fetuses showing ossification")
boxplot(100*fetuses/litter ~ pht, data=subset(ossification,tcpo=="100 mg/kg"),
   add=TRUE, at=c(1:2) + 0.2, col="blue", boxwex=0.25, xaxt="n")
axis(1, at=c(1:2), labels=levels(ossification$pht))
legend("bottomleft", legend=c("0 mg/kg","100 mg/kg"), fill=c("yellow","blue"),
   title="Dose of TCPO", bty="n", cex=0.9)
```

overglm

Alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion.

Description

Allows to fit regression models based on the negative binomial, beta-binomial, and random-clumped binomial distributions, which are alternatives to the Poisson and binomial regression models under the presence of overdispersion.

Usage

```r
overglm(
   formula,
   family = "nb1(log)",
   weights,
   data,
   subset,
```
overglm

na.action = na.omit(),
reltol = 1e-13,
start = NULL,
...
)

Arguments

formula a formula expression of the form response ~ x1 + x2 + ..., which is a symbolic description of the linear predictor of the model to be fitted to the data.

family A character string that allows you to specify the distribution describing the response variable. In addition, it allows you to specify the link function to be used in the model for $\mu$. The following distributions are supported: negative binomial I ("nb1"), negative binomial II ("nb2"), negative binomial ("nbf"), zero-truncated negative binomial I ("ztnb1"), zero-truncated negative binomial II ("ztnb2"), zero-truncated negative binomial ("ztnbf"), zero-truncated poisson ("ztpoi"), beta-binomial ("bb") and random-clumped binomial ("rcb"). Link functions available for these models are the same as those available for Poisson and binomial models via glm. See family documentation.

weights an (optional) vector of positive "prior weights" to be used in the fitting process. The length of weights should be the same as the number of observations.

data an (optional) data frame in which to look for variables involved in the formula expression, as well as for variables specified in the arguments weights and subset.

subset an (optional) vector specifying a subset of individuals to be used in the fitting process.

na.action a function which indicates what should happen when the data contain NAs. By default na.action is set to be na.omit().

reltol an (optional) positive value which represents the relative convergence tolerance for the BFGS method in optim. By default, reltol is set to be 1e-13.

start an (optional) vector of starting values for the parameters in the linear predictor.

Details

The negative binomial distribution can be obtained as mixture of the Poisson and Gamma distributions. If $Y | \lambda \sim \text{Poisson}(\lambda)$, where $E(Y | \lambda) = \text{Var}(Y | \lambda) = \lambda$, and $\lambda \sim \text{Gamma}(\theta, \nu)$, in which $E(\lambda) = \theta$ and $\text{Var}(\lambda) = \nu \theta^2$, then $Y$ is distributed according to the negative binomial distribution. As follows, some special cases are described:

1. If $\theta = \mu$ and $\nu = \phi$ then $Y \sim \text{Negative Binomial I}$, $E(Y) = \mu$ and $\text{Var}(Y) = \mu(1 + \phi \mu)$.
2. If $\theta = \mu$ and $\nu = \phi / \mu$ then $Y \sim \text{Negative Binomial II}$, $E(Y) = \mu$ and $\text{Var}(Y) = \mu(1 + \phi)$.
3. If $\theta = \mu$ and $\nu = \phi \mu^\tau$ then $Y \sim \text{Negative Binomial}$, $E(Y) = \mu$ and $\text{Var}(Y) = \mu(1 + \phi \mu^{\tau+1})$.

Therefore, the regression models based on the negative binomial and zero-truncated negative binomial distributions are alternatives, under overdispersion, to those based on the Poisson and zero-truncated Poisson distributions, respectively.
The beta-binomial distribution can be obtained as a mixture of the binomial and beta distributions. If $mY | π \sim \text{Binomial}(m, π)$, where $E(Y | π) = π$ and $\text{Var}(Y | π) = m^{-1}π(1 − π)$, and $π \sim \text{Beta}(μ, φ)$, in which $E(π) = μ$ and $\text{Var}(π) = (φ+1)^{-1}μ(1−μ)$, with $φ > 0$, then $mY \sim \text{Beta-Binomial}(m, μ, φ)$, so that $E(π) = μ$ and $\text{Var}(π) = (φ+1)^{-1}μ(1−μ)$. Therefore, the regression model based on the beta-binomial distribution is an alternative, under overdispersion, to the binomial regression model.

The random-clumped binomial distribution can be obtained as a mixture of the binomial and Bernoulli distributions. If $mY | π \sim \text{Binomial}(m, π)$, where $E(Y | π) = π$ and $\text{Var}(Y | π) = m^{-1}π(1 − π)$, whereas $π = (1−φ)μ + φ$ with probability $μ$, and $π = (1−φ)μ$ with probability $1−μ$, in which $E(π) = μ$ and $\text{Var}(π) = φ^2μ(1−μ)$, with $φ \in (0, 1)$, then $mY \sim \text{Random-clumped Binomial}(m, μ, φ)$, so that $E(Y) = μ$ and $\text{Var}(Y) = m^{-1}μ(1−μ)[1 + φ^2(m−1)]$. Therefore, the regression model based on the random-clumped binomial distribution is an alternative, under overdispersion, to the binomial regression model.

In all cases, even where the response variable is described by a zero-truncated distribution, the fitted model describes the way in which $μ$ is dependent on some covariates. Parameter estimation is performed using the maximum likelihood method. The model parameters are estimated by maximizing the log-likelihood function through the BFGS method available in the routine optim. The accuracy and speed of the BFGS method are increased because the call to the routine optim is performed using analytical instead of the numerical derivatives. The variance-covariance matrix estimate is obtained as being minus the inverse of the (analytical) hessian matrix evaluated at the parameter estimates and the observed data.

A set of standard extractor functions for fitted model objects is available for objects of class zeroInflation, including methods to the generic functions such as print, summary, model.matrix, estequa, coef, vcov, logLik, fitted, confint, AIC, BIC and predict. In addition, the model fitted to the data may be assessed using functions such as anova.overglm, residuals.overglm, dfbeta.overglm, cooks.distance.overglm, localInfluence.overglm, gvif.overglm and envelope.overglm. The variable selection may be accomplished using the routine stepCriterion.overglm.

Value
an object of class overglm in which the main results of the model fitted to the data are stored, i.e., a list with components including

- coefficients: a vector containing the parameter estimates,
- fitted.values: a vector containing the estimates of $μ_1, \ldots, μ_n$,
- start: a vector containing the starting values used,
- prior.weights: a vector containing the case weights used,
- offset: a vector containing the offset used,
- terms: an object containing the terms objects,
- loglik: the value of the log-likelihood function evaluated at the parameter estimates,
- estfun: a vector containing the estimating functions evaluated at the parameter estimates.
and the observed data,

- **formula**: the formula,
- **levels**: the levels of the categorical regressors,
- **contrasts**: an object containing the contrasts corresponding to levels,
- **converged**: a logical indicating successful convergence,
- **model**: the full model frame,
- **y**: the response count vector,
- **family**: an object containing the `family` object used,
- **linear.predictors**: a vector containing the estimates of $g(\mu_1), \ldots, g(\mu_n)$,
- **R**: a matrix with the Cholesky decomposition of the inverse of the variance-covariance matrix of all parameters in the model,
- **call**: the original function call.

**References**


**See Also**

- `zeroalt`, `zeroinf`

**Examples**

```r
### Example 1: Ability of retinyl acetate to prevent mammary cancer in rats
data(mammary)
fit1 <- overglm(tumors ~ group, family="nb1(identity)", data=mammary)
summary(fit1)

### Example 2: Self diagnosed ear infections in swimmers
data(swimmers)
fit2 <- overglm(infections ~ frequency + location, family="nb1(log)", data=swimmers)
```
### Example 3: Urinary tract infections in HIV-infected men

data(uti)

fit3 <- overglm(episodes ~ cd4 + offset(log(time)), family="nb1(log)", data = uti)

summary(fit3)

### Example 4: Article production by graduate students in biochemistry PhD programs

bioChemists <- pscl::bioChemists

fit4 <- overglm(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)

summary(fit4)

### Example 5: Agents to stimulate cellular differentiation

data(cellular)

fit5 <- overglm(cbind(cells,200-cells) ~ tnf + ifn, family="bb(logit)", data=cellular)

summary(fit5)

### Example 6: Teratogenic effects of phenytoin and trichloropropene oxide

data(ossification)

model6 <- cbind(fetuses,litter-fetuses) ~ pht + tcpo

fit6 <- overglm(model6, family="rcb(cloglog)", data=ossification)

summary(fit6)

### Example 7: Germination of orobanche seeds

data(orobanche)

model7 <- cbind(germinated,seeds-germinated) ~ specie + extract

fit7 <- overglm(model7, family="rcb(cloglog)", data=orobanche)

summary(fit7)

---

**PAC**

***Pardo-Alonso’s Criterion for Generalized Estimating Equations***

**Description**

Computes the Pardo-Alonso’s criterion (PAC) for one or more objects of the class glmgee.

**Usage**

PAC(..., verbose = TRUE, digits = 4)

**Arguments**

... one or several objects of the class glmgee.

verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.

digits an (optional) integer indicating the number of digits to print.
pipeline

Value

A data.frame with the values of the PAC for each glmgee object in the input.

References


See Also

QIC, CIC, RJC, AGPC, SGPC, GHYC

Examples

##### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces

data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
PAC(fit1, fit2, fit3, fit4)

##### Example 2: Treatment for severe postnatal depression

data(depression)
mod2 <- depressd ~ visit + group
fit1 <- glmgee(mod2, id=subj, family=binomial(logit), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
PAC(fit1, fit2, fit3, fit4)

##### Example 3: Treatment for severe postnatal depression (2)

mod3 <- dep ~ visit*group
fit1 <- glmgee(mod3, id=subj, family=gaussian(identity), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Exchangeable")
PAC(fit1, fit2, fit3)

pipeline

Alaska pipeline

Description

The Alaska pipeline data consists of in-field ultrasonic measurements of defects depths in the Alaska pipeline. The depth of the defects was measured again in the laboratory. These measurements were performed in six batches. The data were analyzed to calibrate the bias of field measurements relative to laboratory measurements. In this analysis, the field measurement is the response variable and the laboratory measurement is the predictor variable.
Usage

data(pipeline)

Format

A data frame with 107 rows and 2 variables:

- **Field**: a numeric vector indicating the number of defects measured in the field.
- **Lab**: a numeric vector indicating the number of defects measured in the laboratory.

Source


References


Examples

data(pipeline)
dev.new()
xlab <- "In-laboratory measurements"
ylab <- "In-field measurements"
with(pipeline,plot(Lab,Field,pch=20,xlab=xlab,ylab=ylab))

---

**predict.glmgee**

Predictions for Generalized Estimating Equations

Description

Produces predictions and optionally estimates standard errors of those predictions from a fitted generalized estimating equation.

Usage

```r
## S3 method for class 'glmgee'
predict(  
  object,  
  ...,
  newdata,  
  se.fit = FALSE,
  type = c("link", "response"),
  varest = c("robust", "df-adjusted", "model", "bias-corrected")
)
```
Arguments

object  
an object of the class `glmgee`.

...  
further arguments passed to or from other methods.

newdata  
an (optional) data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.

se.fit  
an (optional) logical switch indicating if standard errors are required. By default, `se.fit` is set to be `FALSE`.

type  
an (optional) character string giving the type of prediction required. The default, "link", is on the scale of the linear predictors, and the alternative, "response", is on the scale of the response variable.

varest  
an (optional) character string indicating the type of estimator which should be used to the variance-covariance matrix of the interest parameters. The available options are: robust sandwich-type estimator ("robust"), degrees-of-freedom-adjusted estimator ("df-adjusted"), bias-corrected estimator ("bias-corrected"), and the model-based or naive estimator ("model"). By default, `varest` is set to be "robust".

Value

A matrix with so many rows as `newdata` and one column with the predictions. If `se.fit=TRUE` then a second column with estimates standard errors is included.

Examples

##### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces

data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces, corstr="AR-M-dependent")
newdata1 <- data.frame(days=c(556,556),treat=as.factor(c("normal","ozone-enriched"))
predict(fit1,newdata=newdata1,type="response",se.fit=TRUE)

##### Example 2: Treatment for severe postnatal depression

data(depression)
mod2 <- depressd ~ visit + group
fit2 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent", data=depression)
newdata2 <- data.frame(visit=c(6,6),group=as.factor(c("placebo","oestrogen"))
predict(fit2,newdata=newdata2,type="response",se.fit=TRUE)

##### Example 3: Treatment for severe postnatal depression (2)

mod3 <- dep ~ visit*group
fit3 <- glmgee(mod3, id=subj, family=gaussian(identity), corstr="AR-M-dependent", data=depression)
newdata3 <- data.frame(visit=c(6,6),group=as.factor(c("placebo","oestrogen"))
predict(fit3,newdata=newdata3,type="response",se.fit=TRUE)
Description

Computes the quasi-likelihood under the independence model criterion (QIC) for one or more objects of the class glmgee.

Usage

QIC(..., k = 2, u = FALSE, verbose = TRUE, digits = 2)

Arguments

... one or several objects of the class glmgee.

k an (optional) non-negative value giving the magnitude of the penalty. By default, k is set to be 2.

u an (optional) logical switch indicating if QIC should be replaced by QICu. By default, u is set to be FALSE.

verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.

digits an (optional) integer indicating the number of digits to print.

Value

A data.frame with the values of -2*quasi-likelihood, the number of parameters in the linear predictor, and the value of QIC (or QICu if u=TRUE) for each glmgee object in the input.

References


See Also

CIC, GHYC, RJC, AGPC, SGPC

Examples

###### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
QIC(fit1, fit2, fit3, fit4)

##### Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ visit + group
fit1 <- glmgee(mod2, id=subj, family=binomial(logit), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
QIC(fit1, fit2, fit3, fit4)

##### Example 3: Treatment for severe postnatal depression (2)
mod3 <- dep ~ visit*group
fit1 <- glmgee(mod3, id=subj, family=gaussian(identity), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Exchangeable")
QIC(fit1, fit2, fit3)

---

**rabbits**

*Age and Eye Lens Weight of Rabbits in Australia*

**Description**

The dry weight of the eye lens was measured for 71 free-living wild rabbits of known age. Eye lens weight tends to vary much less with environmental conditions than does total body weight, and therefore may be a much better indicator of age.

**Usage**

data(rabbits)

**Format**

A data frame with 71 rows and 2 variables:

- **age** a numeric vector indicating the rabbit age, in days.
- **wlens** a numeric vector indicating the dry weight of eye lens, in milligrams.

**References**

Examples

data(rabbits)
dev.new()
with(rabbits,plot(age,wlen,xlab="Age (in days)",pch=16,col="blue",
ylab="Dry weight of eye lens (in milligrams")
)

races
Hill races in Scotland

Description

Each year the Scottish Hill Runners Association publishes a list of hill races in Scotland for the year. These data consist of the record time, distance, and cumulative climb of 35 of those races. The statistical analysis of these data aims to explain the differences between the record time of the races. This is done using their differences in distance and cumulative climb.

Usage

data(races)

Format

A data frame with 35 rows and 4 variables:

- **race** a character vector giving the names of the races.
- **distance** a numeric vector giving the distance, in miles, of the races.
- **cclimb** a numeric vector giving the cumulative climb, in thousands of feet, of the races.
- **rtime** a numeric vector giving the record time, in minutes, of the races.

References


Examples

data(races)
breaks <- with(races,quantile(cclimb,probs=c(0:2)/2))
labels <- c("low","high")
races2 <- within(races,cli <- cut(cclimb,include.lowest=TRUE,breaks,labels))
dev.new()
with(races2,plot(log(distance),log(rtime),pch=16,col=as.numeric(cli)))
legend("topleft", legend=c("low","high"), title="Cumulative climb",
col=c(1:2), pch=16, bty="n")
Description

Calculates residuals for a fitted generalized estimating equation.

Usage

```r
## S3 method for class 'glmgee'
residuals(
  object,
  ..., 
  type = c("mahalanobis", "pearson", "deviance"),
  plot.it = FALSE,
  identify
)
```

Arguments

- `object`: a object of the class `glmgee`.
- `...`: further arguments passed to or from other methods.
- `type`: an (optional) character string giving the type of residuals which should be returned. The available options are: (1) "pearson"; (2) "deviance"; (3) the distance between the observed response vector and the fitted mean vector using a metric based on the product between the cluster size and fitted variance-covariance matrix ("mahalanobis"). By default, `type` is set to be "mahalanobis".
- `plot.it`: an (optional) logical switch indicating if a plot of the residuals is required. By default, `plot.it` is set to be `FALSE`.
- `identify`: an (optional) integer value indicating the number of individuals/clusters to identify on the plot of residuals. This is only appropriate when `plot.it=TRUE`.

Value

A vector with the observed residuals type `type`.

Examples

```r
#### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces, corstr="AR-M-dependent")
### Plot to assess the adequacy of the chosen variance function
residuals(fit1, type="deviance", plot.it=TRUE, col="red", pch=20, col.lab="blue",
         col.axis="blue", col.main="black", family="mono", cex=0.8)
### Plot to identify trees suspicious to be outliers
```
residuals(fit1, type="mahalanobis", plot.it=TRUE, col="red", pch=20, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)

Example 2: Treatment for severe postnatal depression

```r
data(depression)
mod2 <- depressd ~ visit + group
fit2 <- glmgee(mod2, id=subj, family=binomial(logit), corstr="AR-M-dependent", data=depression)
residuals(fit2, type="mahalanobis", plot.it=TRUE, col="red", pch=20, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)
```

Example 3: Treatment for severe postnatal depression (2)

```r
mod3 <- dep ~ visit*group
fit3 <- glmgee(mod3, id=subj, family=gaussian(identity), corstr="AR-M-dependent", data=depression)
residuals(fit3, type="pearson", plot.it=TRUE, col="red", pch=20, col.lab="blue", col.axis="blue", col.main="black", family="mono", cex=0.8)
```
residuals.overglm

- **type**: an (optional) character string which allows to specify the required type of residuals. The available options are: (1) the difference between the observed response and the fitted mean ("response"); (2) the standardized difference between the observed response and the fitted mean ("standardized"); and (3) the randomized quantile residual ("quantile"). By default, type is set to be "quantile".

- **plot.it**: an (optional) logical switch indicating if the plot of residuals versus the fitted values is required. By default, plot.it is set to be FALSE.

- **identify**: an (optional) positive integer value indicating the number of individuals to identify on the plot of residuals versus the fitted values. This is only appropriate if plot.it=TRUE.

- **...**: further arguments passed to or from other methods. If plot.it=TRUE then ... may be used to include graphical parameters to customize the plot. For example, col, pch, cex, main, sub, xlab, ylab.

**Value**

A vector with the observed type-type residuals.

**References**


**Examples**

```
#### Example 1: Self diagnosed ear infections in swimmers
data(swimmers)
fit1 <- overglm(infections ~ frequency + location, family="nb1(log)", data=swimmers)
residuals(fit1, type="quantile", plot.it=TRUE, col="red", pch=20, col.lab="blue",
          col.axis="blue", col.main="black", family="mono", cex=0.8)
```

```
#### Example 2: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit2 <- overglm(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)
residuals(fit2, type="quantile", plot.it=TRUE, col="red", pch=20, col.lab="blue",
          col.axis="blue", col.main="black", family="mono", cex=0.8)
```

```
#### Example 3: Agents to stimulate cellular differentiation
data(cellular)
fit3 <- overglm(cbind(cells,200-cells) ~ tnf + ifn, family="bb(logit)", data=cellular)
residuals(fit3, type="quantile", plot.it=TRUE, col="red", pch=20, col.lab="blue",
          col.axis="blue", col.main="black", family="mono", cex=0.8)
```
residuals.zeroinflation

Residuals in Regression Models to deal with Zero-Excess in Count Data

Description

Computes various types of residuals to assess the individual quality of model fit in regression models to deal with zero-excess in count data.

Usage

```r
## S3 method for class 'zeroinflation'
residuals(
  object,
  type = c("quantile", "standardized", "response"),
  plot.it = FALSE,
  identify,
  ...
)
```

Arguments

- `object`: an object of class `zeroinflation`.
- `type`: an (optional) character string which allows to specify the required type of residuals. The available options are: (1) the difference between the observed response and the fitted mean ("response"); (2) the standardized difference between the observed response and the fitted mean ("standardized"); (3) the randomized quantile residual ("quantile"). By default, `type` is set to be "quantile".
- `plot.it`: an (optional) logical switch indicating if the plot of residuals versus the fitted values is required. By default, `plot.it` is set to be `FALSE`.
- `identify`: an (optional) positive integer value indicating the number of individuals to identify on the plot of residuals versus the fitted values. This is only appropriate if `plot.it=TRUE`.
- `...`: further arguments passed to or from other methods. If `plot.it=TRUE` then `...` may be used to include graphical parameters to customize the plot. For example, `col`, `pch`, `cex`, `main`, `sub`, `xlab`, `ylab`.

Value

A vector with the observed residuals type `type`.

References

Examples

####### Example 1: Self diagnosed ear infections in swimmers
data(swimmers)
fit1 <- zeroalt(infections ~ frequency | location, family="nb1(log)", data=swimmers)
residuals(fit1, type="quantile", col="red", pch=20, col.lab="blue", plot.it=TRUE,
col.axis="blue", col.main="black", family="mono", cex=0.8)

####### Example 2: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit2 <- zeroinf(art ~ fem + kid5 + ment | ment, family="nb1(log)", data = bioChemists)
residuals(fit2, type="quantile", col="red", pch=20, col.lab="blue", plot.it=TRUE,
col.axis="blue", col.main="black", family="mono", cex=0.8)

---

residuals2

Residuals for Linear and Generalized Linear Models

Description

Computes residuals for a fitted linear or generalized linear model.

Usage

residuals2(object, type, standardized = FALSE, plot.it = TRUE, identify, ...)

Arguments

- **object**: a object of the class `lm` or `glm`.
- **type**: an (optional) character string giving the type of residuals which should be returned. The available options for LMs are: (1) externally studentized ("external"); (2) internally studentized ("internal") (default). The available options for GLMs are: (1) "pearson"; (2) "deviance" (default); (3) "quantile".
- **standardized**: an (optional) logical switch indicating if the residuals should be standardized by dividing by the square root of \((1 - h)\), where \(h\) is a measure of leverage. By default, standardized is set to be FALSE.
- **plot.it**: an (optional) logical switch indicating if a plot of the residuals versus the fitted values is required. By default, plot.it is set to be FALSE.
- **identify**: an (optional) integer value indicating the number of individuals to identify on the plot of residuals. This is only appropriate when plot.it=TRUE.
- **...**: further arguments passed to or from other methods

Value

A vector with the observed residuals type type.
Examples

````
# Example 1: Species richness in plots
data(richness)
fit1 <- lm(Species ~ Biomass + pH, data=richness)
residuals2(fit1, type="external", col="red", pch=20, col.lab="blue", plot.it=TRUE,
        col.axis="blue", col.main="black", family="mono", cex=0.8)
```
```
# Example 2: Lesions of Aucuba mosaic virus
data(aucuba)
fit2 <- glm(lesions ~ time, family=poisson, data=aucuba)
residuals2(fit2, type="quantile", col="red", pch=20, col.lab="blue", plot.it=TRUE,
        col.axis="blue",col.main="black",family="mono",cex=0.8)
```
````

```
richness

Species richness

Description

In these data the response is species richness represented by a count of the number of plant species on plots with different biomass and three different soil pH levels: low, mid, and high.

Usage

data(richness)

Format

A data frame with 90 rows and 3 variables:

- **Biomass** a numeric vector giving the value of the biomass in the plots.
- **pH** a factor giving the soil pH level in the plots: "low", "mid", and "high".
- **Species** a numeric vector giving the number of plant species in the plots.

References


Examples

```
data(richness)
dev.new()
with(richness,plot(Biomass,Species,col=as.numeric(pH),pch=16))
legend("topright", legend=c("low","mid","high"), col=c(1:3), pch=16,
        title="pH level", bty="n")
```
Description

These data arose from a study in dentistry. In this trial, subjects were generally healthy adult male and female volunteers, ages 18–55, with pre-existing plaque but without advanced periodontal disease. Prior to entry, subjects were screened for a minimum of 20 sound, natural teeth and a minimum mean plaque index of 2.0. Subjects with gross oral pathology or on antibiotic, antibacterial, or anti-inflammatory therapy were excluded from the study. One hundred nine volunteers were randomized in a double-blinded way to one of two novel mouth rinses (A and B) or to a control mouth rinse. Plaque was scored at baseline, at 3 months, and at 6 months by the Turesky modification of the Quigley-Hein index, a continuous measure. Four subjects had missing plaque scores. The main objective of the analysis is to measure the effectiveness of three mouth rinses at inhibiting dental plaque.

Usage

data(rinse)

Format

A data frame with 315 rows and 7 variables:

- **subject** a character string giving the identifier of the volunteer.
- **gender** a factor indicating the gender of the volunteer: "Female" and "Male".
- **age** a numeric vector indicating the age of the volunteer.
- **rinse** a factor indicating the type of rinse used by the volunteer: "Placebo", "A" and "B".
- **smoke** a factor indicating if the volunteer smoke: "Yes" and "No".
- **time** a numeric vector indicating the time (in months) since the treatment began.
- **score** a numeric vector giving the subject’s score of plaque.

References


Examples

data(rinse)
dev.new()
boxplot(score ~ time, data=subset(rinse,rinse=="Placebo"), ylim=c(0,3.5),
at=c(1:3)-0.2, col="yellow", xaxt="n", boxwex=0.15)
boxplot(score ~ time, data=subset(rinse,rinse=="A"), add=TRUE,
at=c(1:3), col="gray", xaxt="n", boxwex=0.15)
boxplot(score ~ time, data=subset(rinse,rinse=="B"), add=TRUE,
at=c(1:3) + 0.2, col="blue", xaxt="n", boxwex=0.15)
Description

Computes the Rotnitzky–Jewell’s criterion (RJC) for one or more objects of the class glmgee.

Usage

RJC(..., verbose = TRUE, digits = 3)

Arguments

... one or several objects of the class glmgee.
verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.
digits an (optional) integer indicating the number of digits to print.

Value

A data.frame with the values of the RJC for each glmgee object in the input.

References


See Also

QIC, CIC, GHYC, AGPC, SGPC

Examples

##### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days, 4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)"
fit4 <- update(fit1, corstr="Exchangeable")
RJC(fit1, fit2, fit3, fit4)

##### Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ visit + group
```r
fit1 <- glmgee(mod2, id=subj, family=binomial(logit), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
RJC(fit1, fit2, fit3, fit4)

# Example 3: Treatment for severe postnatal depression (2)
mod3 <- dep ~ visit*group
fit1 <- glmgee(mod3, id=subj, family=gaussian(identity), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Exchangeable")
RJC(fit1, fit2, fit3)
```

---

**The Receiver Operating Characteristic (ROC) Curve**

### Description

Computes the exact area under the ROC curve (AUROC), the Gini coefficient, and the Kolmogorov-Smirnov (KS) statistic for a binary classifier. Optionally, this function can plot the ROC curve, that is, the plot of the estimates of Sensitivity versus the estimates of 1-Specificity.

### Usage

```r
ROCc(object, plot.it = TRUE, verbose = TRUE, ...)
```

### Arguments

- **object**: a matrix with two columns: the first one is a numeric vector of 1's and 0's indicating whether each row is a "success" or a "failure"; the second one is a numeric vector of values indicating the probability (or propensity score) of each row to be a "success". Optionally, object can be an object of the class glm which is obtained from the fit of a generalized linear model where the distribution of the response variable is assumed to be binomial.

- **plot.it**: an (optional) logical switch indicating if the plot of the ROC curve is required or just the data matrix in which it is based. By default, plot.it is set to be TRUE.

- **verbose**: an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.

- **...**: further arguments passed to or from other methods. For example, if plot.it=TRUE then ... may to include graphical parameters as col, pch, cex, main, sub, xlab, ylab.
Value

A list which contains the following objects:

- **roc**: A matrix with the Cutoffs and the associated estimates of Sensitivity and Specificity.
- **auroc**: The exact area under the ROC curve.
- **gini**: The value of the Gini coefficient computed as $2(auroc-0.5)$.
- **ks**: The value of the Kolmogorov-Smirnov statistic computed as the maximum value of $|1-$Sensitivity-$Specificity|$.

References


Examples

#### Example: Patients with burn injuries

```r
burn1000 <- aplore3::burn1000
burn1000 <- within(burn1000, death2 <- ifelse(death=="Dead",1,0))

### splitting the sample: 70% for the training sample and 30% for the validation sample
train <- sample(1:nrow(burn1000),size=nrow(burn1000)*0.7)
traindata <- burn1000[train,]
testdata <- burn1000[-train,]

fit <- glm(death ~ age*inh_inj + tbsa*inh_inj, family=binomial("logit"), data=traindata)
probs <- predict(fit, newdata=testdata, type="response")

### ROC curve for the validation sample
ROC(cbind(testdata[,"death2"],probs), col="red", col.lab="blue", col.axis="black", col.main="black", family="mono")
```

Description

The dataset reports the number of epileptic seizures in each of four two-week intervals, and in a baseline eight-week interval, for Progabide treatment and placebo groups with a total of 59 individuals.

Usage

```r
data(Seizures)
```
**Format**

A data frame with 236 rows and 6 variables:

- `seizures` a numeric vector indicating the number of epileptic seizures.
- `treatment` a factor indicating the applied treatment: "Progabide" and "Placebo".
- `base` a numeric vector indicating the number of epileptic seizures in the baseline eight-week interval.
- `age` a numeric vector indicating the age of the individuals.
- `time` a numeric vector indicating which the two-week interval corresponds to the reported number of epileptic seizures.
- `id` a numeric vector indicating the identifier of each individual.

**Source**


**References**


**Examples**

```R
dev.new()
data(Seizures)
boxplot(seizures ~ treatment:time, data=Seizures, ylim=c(0,25), col=c("blue","yellow"))
```

---

**Description**

Computes the Schwarz-type penalized Gaussian pseudo-likelihood criterion (SGPC) for one or more objects of the class `glmgee`.

**Usage**

```R
SGPC(..., verbose = TRUE, digits = 2)
```
Arguments

... one or several objects of the class `glmgee`.

verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.

digits an (optional) integer indicating the number of digits to print.

Value

A data.frame with the values of the gaussian pseudo-likelihood, the number of parameters in the linear predictor plus the number of parameters in the correlation matrix, and the value of SGPC for each `glmgee` object in the input.

References


See Also

`QIC, CIC, RJC, GHYC, AGPC`

Examples

```
##### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod1 <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod1, id=tree, family=Gamma(log), data=spruces)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
SGPC(fit1, fit2, fit3, fit4)

##### Example 2: Treatment for severe postnatal depression
data(depression)
mod2 <- depressd ~ visit + group
fit1 <- glmgee(mod2, id=subj, family=binomial(logit), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
fit3 <- update(fit1, corstr="Stationary-M-dependent(2)")
fit4 <- update(fit1, corstr="Exchangeable")
SGPC(fit1, fit2, fit3, fit4)

##### Example 3: Treatment for severe postnatal depression (2)
mod3 <- dep ~ visit*group
fit1 <- glmgee(mod3, id=subj, family=gaussian(identity), data=depression)
fit2 <- update(fit1, corstr="AR-M-dependent")
```
shelflife

shelflife <- update(fit1, corstr="Exchangeable")
SGPC(fit1, fit2, fit3)

<table>
<thead>
<tr>
<th>shelflife</th>
<th>Shelf life of a photographic developer</th>
</tr>
</thead>
</table>

Description

These data arise from an experiment using accelerated life testing to determine the estimated shelf life of a photographic developer. Maximum density and temperature seem to be reliable indicators of overall developer/film performance.

Usage

data(shelflife)

Format

A data frame with 21 rows and 3 variables:

- **Time**  a numeric vector giving the shelf life, in hours.
- **Temp**  a factor giving the temperature, in degrees celsius.
- **Dmax**  a numeric vector giving the maximum density.

References


Examples

data(shelflife)
dev.new()
with(shelflife, plot(Dmax, Time, pch=16, col=as.numeric(Temp)))
legend("topright", legend=c("72C","82C","92C"), col=c(1:3), pch=16,
    title="Temperature", bty="n")
Skin cancer in women

Description

The data describe the incidence of nonmelanoma skin cancer among women stratified by age in Minneapolis (St. Paul) and Dallas (Fort Worth).

Usage

data(skincancer)

Format

A data frame with 16 rows and 4 variables:

- **cases** a numeric vector giving the nonmelanoma skin cancer counts.
- **city** a factor giving the city to which correspond the skin cancer counts: "St.Paul" and "Ft.Worth".
- **ageC** a factor giving the age range to which correspond the skin cancer counts: "15-24", "25-34", "35-44", "45-54", "55-64", "65-74", "75-84" and "85+".
- **population** a numeric vector giving the population of women.
- **age** a numeric vector giving the midpoint of age range.

References


Examples

data(skincancer)
dev.new()
barplot(1000*cases/population ~ city + ageC, beside=TRUE, col=c("yellow","blue"), data=skincancer)
legend("topleft", legend=c("St.Paul","Ft.Worth"), title="City", fill=c("yellow","blue"), bty="n")
Effect of ozone-enriched atmosphere on growth of sitka spruces

Description

These data are analyzed primarily to determine how ozone pollution affects tree growth. As ozone pollution is common in urban areas, the impact of increased ozone concentrations on tree growth is of considerable interest. The response variable is tree size, where size is conventionally measured by the product of tree height and stem diameter squared. In the first group, 54 trees were grown under an ozone-enriched atmosphere, ozone exposure at 70 parts per billion. In the second group, 25 trees were grown under normal conditions. The size of each tree was observed 13 times across time, that is, 152, 174, 201, 227, 258, 469, 496, 528, 556, 579, 613, 639 and 674 days since the beginning of the experiment. Hence, the objective is to compare the trees' growth patterns under the two conditions.

Usage

data(spruces)

Format

A data frame with 1027 rows and 4 variables:

- **tree**: a factor giving an unique identifier for each tree.
- **days**: a numeric vector giving the number of days since the beginning of the experiment.
- **size**: a numeric vector giving an estimate of the volume of the tree trunk.
- **treat**: a factor giving the treatment received for each tree: "normal" and "ozone-enriched".

References


Examples

data(spruces)
dev.new()
boxplot(size ~ days, data=subset(spruces,treat=="normal"), at=c(1:13) - 0.2, col="yellow", boxwex=0.3, xaxt="n", xlim=c(0.9,13.1))
boxplot(size ~ days, data=subset(spruces,treat=="ozone-enriched"), add=TRUE, at=c(1:13) + 0.2, col="blue", boxwex=0.3, xaxt="n")
axis(1, at=c(1:13), labels=unique(spruces$days))
axis(2, at=seq(0,2000,250), labels=seq(0,2000,250))
legend("topleft", legend=c("normal","ozone-enriched"), fill=c("yellow","blue"), title="Atmosphere", bty="n")
Steel

Hardened Steel

Description
This dataset consists of failure times for hardened steel specimens in a rolling contact fatigue test. Ten independent observations were taken at each of the four contact stress values. Response is the time that each specimen of hardened steel failed.

Usage
data(Steel)

Format
A data frame with 40 rows and 2 variables:

- **stress**: a numeric vector indicating the values of contact stress, in pounds per square inch x 10^{-6}.
- **life**: a numeric vector indicating the length of the time until the specimen of the hardened steel failed.

References

Examples
dev.new()
data(Steel)
with(Steel, plot(log(stress), log(life), pch=16, xlab="Log(Stress)", ylab="log(Life)"))

stepCriterion

Variable selection in regression models from a chosen criterion

Description
Generic function for selecting variables from a fitted regression model using a chosen criterion.

Usage
stepCriterion(model, ...)

Arguments
- **model**: a fitted model object.
- **...**: further arguments passed to or from other methods.
Value

A list which includes the descriptions of the linear predictors of the initial and final models as well as the criterion used to compare the candidate models.

Description

Performs variable selection in generalized linear models using hybrid versions of forward stepwise and backward stepwise.

Usage

```
## S3 method for class 'glm'
stepCriterion(
  model,
  criterion = c("adjr2", "bic", "aic", "p-value", "qicu"),
  test = c("wald", "lr", "score", "gradient"),
  direction = c("forward", "backward"),
  levels = c(0.05, 0.05),
  trace = TRUE,
  scope,
  ...
)
```

Arguments

- **model**: an object of the class `glm`.
- **criterion**: an (optional) character string indicating the criterion which should be used to compare the candidate models. The available options are: AIC ("aic"), BIC ("bic"), adjusted deviance-based R-squared ("adjr2"), and p-value of the test ("p-value"). By default, `criterion` is set to be "adjr2".
- **test**: an (optional) character string indicating the statistical test which should be used to compare nested models. The available options are: Wald ("wald"), Rao's score ("score"), likelihood-ratio ("lr") and gradient ("gradient") tests. By default, `test` is set to be "wald".
- **direction**: an (optional) character string indicating the type of procedure which should be used. The available options are: hybrid backward stepwise ("backward") and hybrid forward stepwise ("forward"). By default, `direction` is set to be "forward".
- **levels**: an (optional) two-dimensional vector of values in the interval (0, 1) indicating the levels at which the variables should in and out from the model. This is only appropriate if `criterion="p-value"`. By default, `levels` is set to be `c(0.05,0.05)`. 
trace
   an (optional) logical switch indicating if should the stepwise reports be printed. By default, trace is set to be TRUE.

scope
   an (optional) list, containing components lower and upper, both formula-type objects, indicating the range of models which should be examined in the stepwise search. By default, lower is a model with no predictors and upper is the linear predictor of the model in model.

... further arguments passed to or from other methods. For example, k, that is, the magnitude of the penalty in the AIC/QICu, which by default is set to be 2.

Details

The "hybrid forward stepwise" algorithm starts with the simplest model (which may be chosen at the argument scope, and by default, is a model whose parameters in the linear predictor, except the intercept, if any, are set to be 0), and then the candidate models are built by hierarchically including effects in the linear predictor, whose "relevance" and/or "importance" in the model fit is assessed by comparing nested models (that is, by comparing the models with and without the added effect) using a criterion previously specified. If an effect is added to the equation, this strategy may also remove any effect which, according to the previously specified criterion, no longer provides improvement in the model fit. That process continues until no more effects are included or excluded. The "hybrid backward stepwise" algorithm works similarly.

Value

a list list with components including

initial  a character string indicating the linear predictor of the "initial model",
direction a character string indicating the type of procedure which was used,
criterion a character string indicating the criterion used to compare the candidate models,
final    a character string indicating the linear predictor of the "final model",

References


See Also

stepCriterion.lm, stepCriterion.overglm, stepCriterion.glmgee

Examples

####### Example 1: Fuel consumption of automobiles
Auto <- ISLR::Auto
Auto2 <- within(Auto, origin <- factor(origin))
mod <- mpg ~ cylinders + displacement + acceleration + origin + horsepower*weight

---

stepCriterion.glm
fit1 <- glm(mod, family=inverse.gaussian("log"), data=Auto2)
stepCriterion(fit1, direction="forward", criterion="p-value", test="lr")
stepCriterion(fit1, direction="backward", criterion="bic")

#### Example 2: Patients with burn injuries
burn1000 <- aplore3::burn1000
burn1000 <- within(burn1000, death <- factor(death, levels=c("Dead","Alive")))
upper <- ~ age + gender + race + tbsa + inh_inj + flame + age*inh_inj + tbsa*inh_inj
lower <- ~ 1
fit2 <- glm(death ~ age + gender + race + tbsa + inh_inj, family=binomial("logit"), data=burn1000)
stepCriterion(fit2, direction="backward", criterion="bic", scope=list(lower=lower, upper=upper))
stepCriterion(fit2, direction="forward", criterion="p-value", test="score")

#### Example 3: Skin cancer in women
data(skincancer)
upper <- cases ~ city + age + city*age
fit3 <- glm(upper, family=poisson("log"), offset=log(population), data=skincancer)
stepCriterion(fit3, direction="backward", criterion="aic", scope=list(lower=~ 1, upper=upper))
stepCriterion(fit3, direction="forward", criterion="p-value", test="lr")

---

**stepCriterion.glmgee**  
Variable selection in Generalized Estimating Equations

**Description**

Performs variable selection in generalized estimating equations using hybrid versions of forward stepwise and backward stepwise.

**Usage**

```r
## S3 method for class 'glmgee'
stepCriterion(
  model,
  criterion = c("p-value", "qic", "qicu", "agpc", "sgpc"),
  test = c("wald", "score"),
  direction = c("forward", "backward"),
  levels = c(0.05, 0.05),
  trace = TRUE,
  scope,
  digits = 5,
  varest = c("robust", "df-adjusted", "model", "bias-corrected"),
  ...)
```

**Arguments**

- `model`: an object of the class glmgee which is obtained from the fit of a generalized estimating equation.
criterion: an (optional) character string indicating the criterion which should be used to compare the candidate models. The available options are: QIC ("qic"), QICu ("qicu"), Akaike-type penalized gaussian pseudo-likelihood criterion ("agpc"), Schwarz-type penalized gaussian pseudo-likelihood criterion ("sgpc") and p-value of the test test ("p-value"). By default, criterion is set to be "p-value".

test: an (optional) character string indicating the statistical test which should be used to compare nested models. The available options are: Wald ("wald") and generalized score ("score") tests. By default, test is set to be "wald".

direction: an (optional) character string indicating the type of procedure which should be used. The available options are: hybrid backward stepwise ("backward") and hybrid forward stepwise ("forward"). By default, direction is set to be "forward".

levels: an (optional) two-dimensional vector of values in the interval (0, 1) indicating the levels at which the variables should in and out from the model. This is only appropriate if criterion="p-value". By default, levels is set to be c(0.05, 0.05).

trace: an (optional) logical switch indicating if should the stepwise reports be printed. By default, trace is set to be TRUE.

scope: an (optional) list, containing components lower and upper, both formula-type objects, indicating the range of models which should be examined in the stepwise search. By default, lower is a model with no predictors and upper is the linear predictor of the model in model.

digits: an (optional) integer indicating the number of digits which should be used to print the most of the criteria to compare the candidate models. By default, digits is set to be 5.

varest: an (optional) character string indicating the type of estimator which should be used to the variance-covariance matrix of the interest parameters in the Wald-type test. The available options are: robust sandwich-type estimator ("robust"), degrees-of-freedom-adjusted estimator ("df-adjusted"), bias-corrected estimator ("bias-corrected"), and the model-based or naive estimator ("model"). By default, varest is set to be "robust".

... further arguments passed to or from other methods. For example, k, that is, the magnitude of the penalty in the AGPC, which by default is set to be 2.

Value

A list which contains the following objects:

• initial: a character string indicating the linear predictor of the "initial model".
• direction: a character string indicating the type of procedure which was used.
• criterion: a character string indicating the criterion used to compare the candidate models.
• final: a character string indicating the linear predictor of the "final model".
References


See Also

stepCriterion.lm, stepCriterion.glm, stepCriterion.overglm

Examples

```
#### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
data(spruces)
mod <- size ~ poly(days,4)*treat
fit1 <- glmgee(mod, id=tree, family=Gamma(log), data=spruces, corstr="AR-M-dependent")
stepCriterion(fit1, criterion="p-value", direction="forward", scope=list(lower=~1,upper=mod))

#### Example 2: Treatment for severe postnatal depression
data(depression)
mod <- depressd ~ visit*group
fit2 <- glmgee(mod, id=subj, family=binomial(probit), corstr="AR-M-dependent", data=depression)
stepCriterion(fit2, criterion="agpc", direction="forward", scope=list(lower=~1,upper=mod))

#### Example 3: Treatment for severe postnatal depression (2)
mod <- dep ~ visit*group
fit2 <- glmgee(mod, id=subj, family=gaussian(identity), corstr="AR-M-dependent", data=depression)
stepCriterion(fit2, criterion="sgpc", direction="forward", scope=list(lower=~1,upper=mod))
```

Description

Performs variable selection in normal linear models using a hybrid versions of forward stepwise and backward stepwise.

Usage

```
## S3 method for class 'lm'
stepCriterion(
  model,
  criterion = c("bic", "aic", "adjr2", "prdr2", "cp", "p-value"),
  direction = c("forward", "backward"),
  levels = c(0.05, 0.05),
  trace = TRUE,
```

stepCriterion.lm  Variable Selection in Normal Linear Models
Arguments

model an object of the class lm.
criterion an (optional) character string indicating the criterion which should be used to compare the candidate models. The available options are: AIC ("aic"), BIC ("bic"), adjusted R-squared ("adjr2"), predicted R-squared ("prdr2"), Mallows' CP ("cp") and p-value of the F test ("p-value"). By default, criterion is set to be "bic".
direction an (optional) character string indicating the type of procedure which should be used. The available options are: hybrid backward stepwise ("backward") and hybrid forward stepwise ("forward"). By default, direction is set to be "forward".
levels an (optional) two-dimensional vector of values in the interval (0, 1) indicating the levels at which the variables should in and out from the model. This is only appropriate if criterion="p-value". By default, levels is set to be c(0.05, 0.05).
trace an (optional) logical switch indicating if should the stepwise reports be printed. By default, trace is set to be TRUE.
scope an (optional) list containing components lower and upper, both formula-type objects, indicating the range of models which should be examined in the stepwise search. By default, lower is a model with no predictors and upper is the linear predictor of the model in model.
... further arguments passed to or from other methods. For example, k, that is, the magnitude of the penalty in the AIC/QICu, which by default is set to be 2.

Details

The "hybrid forward stepwise" algorithm starts with the simplest model (which may be chosen at the argument scope, and by default, is a model whose parameters in the linear predictor, except the intercept, if any, are set to be 0), and then the candidate models are built by hierarchically including effects in the linear predictor, whose "relevance" and/or "importance" in the model fit is assessed by comparing nested models (that is, by comparing the models with and without the added effect) using a criterion previously specified. If an effect is added to the equation, this strategy may also remove any effect which, according to the previously specified criteria, no longer provides an improvement in the model fit. That process continues until no more effects are included or excluded. The "hybrid backward stepwise" algorithm works similarly.

Value

a list list with components including

initial a character string indicating the linear predictor of the "initial model",
direction a character string indicating the type of procedure which was used,
criterion a character string indicating the criterion used to compare the candidate models,
final a character string indicating the linear predictor of the "final model".

References

See Also
stepCriterion.glm, stepCriterion.overglm, stepCriterion.glmgee
stepCriterion.glm, stepCriterion.overglm, stepCriterion.glmgee

Examples

```
###### Example 1: New York air quality measurements
fit1 <- lm(log(Ozone) ~ Solar.R + Temp + Wind, data=airquality)
scope=list(lower=~1, upper=~Solar.R*Temp*Wind)
stepCriterion(fit1, direction="forward", criterion="adjr2", scope=scope)
stepCriterion(fit1, direction="forward", criterion="bic", scope=scope)
stepCriterion(fit1, direction="forward", criterion="p-value", scope=scope)

###### Example 2: Fuel consumption of automobiles
fit2 <- lm(mpg ~ log(hp) + log(wt) + qsec, data=mtcars)
scope=list(lower=~1, upper=~log(hp)*log(wt)*qsec)
stepCriterion(fit2, direction="backward", criterion="bic", scope=scope)
stepCriterion(fit2, direction="forward", criterion="cp", scope=scope)
stepCriterion(fit2, direction="backward", criterion="prdr2", scope=scope)

###### Example 3: Credit card balance
Credit <- ISLR::Credit
fit3 <- lm(Balance ~ Cards + Age + Rating + Income + Student + Limit, data=Credit)
stepCriterion(fit3, direction="forward", criterion="prdr2")
stepCriterion(fit3, direction="forward", criterion="cp")
stepCriterion(fit3, direction="forward", criterion="p-value")
```

stepCriterion.overglm Variable selection for alternatives to the Poisson and Binomial Regression Models under the presence of Overdispersion

Description
Performs variable selection using hybrid versions of forward stepwise and backward stepwise by comparing hierarchically built candidate models using a criterion previously specified such as AIC, BIC or p-value of the significance tests.
Usage

```r
## S3 method for class 'overglm'
stepCriterion(
  model,
  criterion = c("bic", "aic", "p-value"),
  test = c("wald", "score", "lr", "gradient"),
  direction = c("forward", "backward"),
  levels = c(0.05, 0.05),
  trace = TRUE,
  scope,
  ...
)
```

Arguments

- `model`: an object of the class `overglm`.
- `criterion`: an (optional) character string which allows to specify the criterion which should be used to compare the candidate models. The available options are: AIC ("aic"), BIC ("bic"), and p-value of the test-type test ("p-value"). By default, criterion is set to be "bic".
- `test`: an (optional) character string which allows to specify the statistical test which should be used to compare nested models. The available options are: Wald ("wald"), Rao's score ("score"), likelihood-ratio ("lr") and gradient ("gradient") tests. By default, test is set to be "wald".
- `direction`: an (optional) character string which allows to specify the type of procedure which should be used. The available options are: hybrid backward stepwise ("backward") and hybrid forward stepwise ("forward"). By default, direction is set to be "forward".
- `levels`: an (optional) two-dimensional vector of values in the interval (0,1) indicating the levels at which the variables should in and out from the model. This is only appropriate if criterion="p-value". By default, levels is set to be c(0.05,0.05).
- `trace`: an (optional) logical switch indicating if should the stepwise reports be printed. By default, trace is set to be TRUE.
- `scope`: an (optional) list, containing components `lower` and `upper`, both formula-type objects, indicating the range of models which should be examined in the stepwise search. By default, `lower` is a model with no predictors and `upper` is the linear predictor of the model in `model`.
- `...`: further arguments passed to or from other methods. For example, k, that is, the magnitude of the penalty in the AIC, which by default is set to be 2.

Value

A list which contains the following objects:

- `initial`: a character string indicating the linear predictor of the "initial model",
direction  a character string indicating the type of procedure which was used,
criterion  a character string indicating the criterion used to compare the candidate models,
final     a character string indicating the linear predictor of the "final model".

References


See Also

stepCriterion.lm, stepCriterion.glm, stepCriterion.glmgee

Examples

###### Example 1: Self diagnosed ear infections in swimmers
data(swimmers)
fit1 <- overglm(infections ~ age + gender + frequency + location, family="nb1(log)", data=swimmers)
stepCriterion(fit1, criterion="p-value", direction="forward", test="lr")
stepCriterion(fit1, criterion="bic", direction="backward", test="score")

###### Example 2: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit2 <- overglm(art ~ fem + mar + kid5 + phd + ment, family="nb1(log)", data = bioChemists)
stepCriterion(fit2, criterion="p-value", direction="forward", test="lr")
stepCriterion(fit2, criterion="bic", direction="backward", test="score")

###### Example 3: Agents to stimulate cellular differentiation
data(cellular)
fit3 <- overglm(cbind(cells,200-cells) ~ tnf + ifn + tnf*ifn, family="bb(logit)", data=cellular)
stepCriterion(fit3, criterion="p-value", direction="backward", test="lr")
stepCriterion(fit3, criterion="bic", direction="forward", test="score")
Description
A pilot surf/health study was conducted by NSW Water Board in 1990 on 287 recruits. The objective of the study was to determine whether beach swimmers run an increased risk of contracting ear infections than non-beach swimmers.

Usage
data(swimmers)

Format
A data frame with 287 rows and 5 variables:

frequency a factor giving the recruit’s perception of whether he or she is a frequent swimmer: "frequent" and "occasional".
location a factor giving the recruit’s usually chosen swimming location: "beach" and "non-beach".
age a factor giving the recruit’s age range: "15-19", "20-24" and "25-29".
gender a factor giving the recruit’s gender: "male" and "female".
infections a numeric vector giving the number of self diagnosed ear infections that were reported by the recruit.

References

Examples
data(swimmers)
dev.new()
boxplot(infections ~ frequency, data=subset(swimmers, location=="non-beach"),
at=c(1:2) - 0.2, col="yellow", boxwex=0.25, xaxt="n")
boxplot(infections ~ frequency, data=subset(swimmers, location=="beach"), add=TRUE,
at=c(1:2) + 0.2, col="blue", boxwex=0.25, xaxt="n")
axis(1, at=c(1:2), labels=levels(swimmers$frequency))
legend("topleft", title="Location", legend=c("non-beach","beach"),
fill=c("yellow","blue"), bty="n")
Description

The data arose from a horticultural experiment to study the number of roots produced by 270 micro-
propagated shoots of the columnar apple cultivar Trajan. During the rooting period, all shoots were
maintained under identical conditions. However, the shoots themselves were cultured on media
containing different concentrations of the cytokinin 6-benzylaminopurine (BAP), in growth cabi-
nets with an 8 or 16 hour photoperiod. The objective is to assess the effect of both the photoperiod
and BAP concentration levels on the number of roots produced.

Usage

data(Trajan)

Format

A data frame with 270 rows and 4 variables:

- **roots** a numeric vector indicating the number of roots produced.
- **shoot** a numeric vector indicating the number of micropropogated shoots.
- **photoperiod** a factor indicating the photoperiod, in hours: 8 or 16.
- **bap** a numeric vector indicating the concentrations of the cytokinin 6-benzylaminopurine: 2.2, 4.4,
  8.8 or 17.6.

Source

https://support.sas.com/rnd/app/stat/examples/GENMODZIP/sas.html

References

of the XIXth international biometric conference, 179–192.

Ridout M., Hinde J., Demétrio C.G. (2001). A score test for testing a zero-inflated Poisson regres-

diagnostics for zero-inflated negative binomial regression models. Computational Statistics & Data
Analysis 55:1304-1318.

Examples

data(Trajan)
dev.new()
boxplot(roots ~ bap, data=subset(Trajan, photoperiod=="8"), at=c(1:4) - 0.15,
col="blue", boxwex=0.2, xaxt="n", ylim=c(-0.5,17))
boxplot(roots ~ bap, data=subset(Trajan, photoperiod=="16"), add=TRUE,
Urinary Tract Infections in HIV-infected Men

Description

These data arose from a study conducted in the Department of Internal Medicine at Utrecht University Hospital, in the Netherlands. In this study, 98 HIV-infected men were followed for up to two years. Urinary cultures were obtained during the first visit and every six months thereafter. Also, cultures were obtained between regular scheduled visits when signs and symptoms of urinary tract infections (UTI) occurred, or when patients had a fever of unknown origin. CD4+ cell counts were also measured. A CD4+ count is a blood test to determine how well the immune system works in people diagnosed with HIV. In general, a decreasing CD4+ count indicates HIV progression.

Usage

data(uti)

Format

A data frame with 98 rows and 3 variables:

- **episodes** a numeric vector indicating the number of episodes, that is, the number of times each patient had urinary tract infections (UTI).
- **time** a numeric vector indicating the time to follow up, in months.
- **cd4** a numeric vector indicating the immune status of the patient as measured by the CD4+ cell counts.

References


Examples

```r
data(uti)
dev.new()
uti2 <- within(uti, cd4C <- cut(log(cd4), 4, labels=c("low","mid-low","mid-high","high")))
out <- aggregate(cbind(episodes,time) ~ cd4C, sum, data=uti2)
barplot(12*episodes/time ~ cd4C, beside=TRUE, data=out, col="red",
xlab="CD4+ cell count", ylab="Number of UTIs per year")
```

vcov.glmgee

**Estimate of the variance-covariance matrix in GEEs**

Description

Computes the type-type estimate of the variance-covariance matrix from an object of the class `glmgee`.

Usage

```r
## S3 method for class 'glmgee'
vcov(
  object,
  ...,
  type = c("robust", "df-adjusted", "model", "bias-corrected", "jackknife")
)
```

Arguments

- `object`: An object of the class `glmgee`.
- `...`: further arguments passed to or from other methods.
- `type`: an (optional) character string indicating the type of estimator which should be used. The available options are: robust sandwich-type estimator ("robust"), degrees-of-freedom-adjusted estimator ("df-adjusted"), bias-corrected estimator ("bias-corrected"), and the model-based or naive estimator ("model"). By default, `type` is set to be "robust".

Value

A matrix with the type-type estimate of the variance-covariance matrix.

References

Examples

##### Example 1: Effect of ozone-enriched atmosphere on growth of sitka spruces
```r
data(spruces)
mod <- size ~ poly(days,4) + treat
fit1 <- glmgee(mod, id=tree, family=Gamma(log), data=spruces, corstr="Exchangeable")
vcov(fit1)
vcov(fit1,type="bias-corrected")
```

##### Example 2: Treatment for severe postnatal depression
```r
data(depression)
mod <- depressd ~ visit + group
fit3 <- glmgee(mod, id=subj, family=binomial(logit), corstr="AR-M-dependent", data=depression)
vcov(fit3)
vcov(fit3,type="bias-corrected")
```

##### Example 3: Treatment for severe postnatal depression (2)
```r
mod <- dep ~ visit*group
fit2 <- glmgee(mod, id=subj, family=gaussian(identity), corstr="AR-M-dependent", data=depression)
vcov(fit2)
vcov(fit2,type="bias-corrected")
```

vdtest

Test for Varying Dispersion Parameter

Description

Generic function for testing for varying dispersion parameter from a fitted model.

Usage

```r
vdtest(model, ...)
```

Arguments

- `model` a fitted model object.
- `...` further arguments passed to or from other methods.

Value

A list which includes the main attributes of the test as, for example, value of the statistic and p-value.
Description
Performs Rao’s score test for varying dispersion parameter in weighted and unweighted generalized linear models in which the response distribution is assumed to be Gaussian, Gamma, or inverse Gaussian.

Usage

```r
## S3 method for class 'glm'
vdtest(model, varformula, verbose = TRUE, ...)
```

Arguments

- `model`: an object of the class `glm` where the distribution of the response variable is assumed to be `gaussian`, `Gamma` or `inverse.gaussian`.
- `varformula`: an (optional) formula expression of the form `~ z1 + z2 + ... + zq` describing only the potential explanatory variables for the dispersion. By default, the same explanatory variables are taken as in the model for the mean.
- `verbose`: an (optional) logical switch indicating if should the report of results be printed. By default, `verbose` is set to be `TRUE`.
- `...`: further arguments passed to or from other methods.

Details

From the generalized linear model with varying dispersion in which \( \log(\phi) = \gamma_0 + \gamma_1 z_1 + \gamma_2 z_2 + ... + \gamma_q z_q \), where \( \phi \) is the dispersion parameter of the distribution used to describe the response variable, the Rao’s score test (denoted here as \( S \)) to assess the hypothesis \( H_0 : \gamma = 0 \) versus \( H_1 : \gamma \neq 0 \) is computed, where \( \gamma = (\gamma_1, \ldots, \gamma_q) \). The corresponding \( p \)-value is computed from the chi-squared distribution with \( q \) degrees of freedom, that is, \( p \text{-value} = \text{Prob}[\chi^2_q > S] \). If the object `model` corresponds to an unweighted generalized linear model then this test assesses assumptions of constant variance and constant coefficient of variation on models in which the response distribution is assumed to be Gaussian and Gamma, respectively.

Value

- a list list with components including
  - `statistic`: value of the Rao’s score test (\( S \)),
  - `df`: number of degrees of freedom (\( q \)),
  - `p.value`: \( p \)-value of the test,
References


See Also

vdtest.lm

Examples

```
### Example 1: Fuel consumption of automobiles
Auto <- ISLR::Auto
fit1 <- glm(mpg ~ weight*horsepower, family=inverse.gaussian("log"), data=Auto)
vdtest(fit1)

### Example 2: Hill races in Scotland
data(races)
fit2 <- glm(rtime ~ log(distance) + cclimb, family=Gamma("log"), data=races)
vdtest(fit2)

### Example 3: Mammal brain and body weights
data(brains)
fit3 <- glm(BrainWt ~ log(BodyWt), family=Gamma("log"), data=brains)
vdtest(fit3)
```

vdtest.lm

*Test for Varying Dispersion Parameter in Normal Linear Models*

Description

Performs Rao's score test for varying dispersion parameter in weighted and unweighted normal linear models.

Usage

```
# S3 method for class 'lm'
vdtest(model, varformula, verbose = TRUE, ...)
```

Arguments

- **model**: an object of the class `lm`.
- **varformula**: an (optional) formula expression of the form \( z_1 + z_2 + \ldots + z_q \) indicating the potential explanatory variables for the dispersion parameter. By default, the same explanatory variables are taken as in the model for the mean.
- **verbose**: an (optional) logical switch indicating if should the report of results be printed. By default, `verbose` is set to be `TRUE`.
- ... further arguments passed to or from other methods.
Details
From the heteroskedastic normal lineal model in which \( \log(\sigma^2) = \gamma_0 + \gamma_1 z_1 + \gamma_2 z_2 + \ldots + \gamma_q z_q \), where \( \sigma^2 \) is the dispersion parameter of the distribution of the random errors, the Rao’s score test (denoted here as \( S \)) to assess the hypothesis \( H_0 : \gamma = 0 \) versus \( H_1 : \gamma \neq 0 \) is computed, where \( \gamma = (\gamma_1, \ldots, \gamma_q) \). The corresponding \( p \)-value is computed from the chi-squared distribution with \( q \) degrees of freedom, that is, \( p \)-value = \( \text{Prob}[\chi^2_q > S] \). If the object model corresponds to an unweighted normal linear model, then the test assess the assumption of constant variance, which coincides with the non-studentized Breusch-Pagan test against heteroskedasticity.

Value
a list list with components including

- **statistic** value of the Rao’s score test (\( S \)),
- **df** number of degrees of freedom (\( q \)),
- **p.value** \( p \)-value of the test,

References


See Also
vdtest.glm

Examples

```r
# Example 1: Fuel consumption of automobiles
fit1 <- lm(mpg ~ log(hp) + log(wt), data=mtcars)
vdtest(fit1)

# Example 2: Species richness in plots
data(richness)
fit2 <- lm(Species ~ Biomass + pH, data=richness)
vdtest(fit2)

# The test conclusions change when the outlying observations are excluded
fit2a <- lm(Species ~ Biomass + pH, data=richness, subset=-c(1,3,18,20))
vdtest(fit2a)

# Example 3: Gas consumption in a home before and after insulation
whiteside <- MASS::whiteside
fit3 <- lm(Gas ~ Temp + Insul + Temp*Insul, data=whiteside)
vdtest(fit3)
```
### The test conclusions change when the outlying observations are excluded

```r
fit3a <- lm(Gas ~ Temp + Insul + Temp*Insul, data=whiteside, subset=-c(8,9,36,46,55))
vdtest(fit3a)
```

---

**wglmgee**

*Fit Weighted Generalized Estimating Equations*

---

**Description**

Produces an object of the class `wglmgee` in which the main results of a Weighted Generalized Estimating Equation (WGEE) fitted to the data are stored.

**Usage**

```r
wglmgee(
  formula,
  level = c("observations", "clusters"),
  family = gaussian(),
  weights, id, data, subset, corstr, corr,
  start = NULL, scale.fix = FALSE, scale.value = 1,
  toler = 1e-05, maxit = 50,
  trace = FALSE,
  ...
)
```

**Arguments**

- **formula**: an `Formula` expression of the form `response ~ x1 + x2 + ... | z1 + z2 + ...`, whose first part is a symbolic description of the linear predictor of the GEE model to be fitted to the data, whereas the second part is a symbolic description of the linear predictor of the logistic model to be used to calculate the missingness probabilities under the MAR assumption. Then, those probabilities are used to computed the weights to be included in the parameter estimation algorithm.

- **level**: an (optional) character string which allows to specify the weighted GEE method. The available options are: "observations" and "clusters" for Observation- and Cluster-specified Weighted GEE, respectively. By default, `level` is set to be "observations".
family an (optional) family object, that is, a list of functions and expressions for defining link and variance functions. Families (and links) supported are the same supported by `glm` using its `family` argument, that is, `gaussian`, `binomial`, `poisson`, `Gamma`, `inverse.gaussian`, and `quasi`. The family `negative.binomial` in the library `MASS` are also available. By default, the argument `family` is set to be `gaussian(identity)`.

weights an (optional) vector of positive "prior weights" to be used in the fitting process. The length of weights should be the same as the total number of observations.

id a vector which identifies the subjects or clusters. The length of id should be the same as the number of observations.

data an (optional) data frame in which to look for variables involved in the formula expression, as well as for variables specified in the arguments `id` and `weights`. The data are assumed to be sorted by `id` and time.

subset an (optional) vector specifying a subset of observations to be used in the fitting process.

corstr an (optional) character string which allows to specify the working-correlation structure. The available options are: "Independence", "Unstructured", "Stationary-M-dependent(m)", "Non-Stationary-M-dependent(m)", "AR-M-dependent(m)", "Exchangeable" and "User-defined", where `m` represents the lag of the dependence. By default, `corstr` is set to be "Independence".

corr an (optional) square matrix of the same dimension of the maximum cluster size containing the user specified correlation. This is only appropriate if `corstr` is specified to be "User-defined".

start an (optional) vector of starting values for the parameters in the linear predictor.

scale.fix an (optional) logical variable. If TRUE, the scale parameter is fixed at the value of `scale.value`. By default, `scale.fix` is set to be FALSE.

scale.value an (optional) numeric value at which the scale parameter should be fixed. This is only appropriate if `scale.fix=TRUE`. By default, `scale.value` is set to be 1.

toler an (optional) positive value which represents the convergence tolerance. The convergence is reached when the maximum of the absolute relative differences between the values of the parameters in the linear predictor in consecutive iterations of the fitting algorithm is lower than `toler`. By default, `toler` is set to be 0.00001.

maxit an (optional) integer value which represents the maximum number of iterations allowed for the fitting algorithm. By default, `maxit` is set to be 50.

trace an (optional) logical variable. If TRUE, output is produced for each iteration of the estimating algorithm.

... further arguments passed to or from other methods.

Details

The values of the multivariate response variable measured on `n` subjects or clusters, denoted by \( y_i = (y_{i1}, \ldots, y_{in_i})^\top \) for \( i = 1, \ldots, n \), are assumed to be realizations of independent random vectors denoted by \( Y_i = (Y_{i1}, \ldots, Y_{in_i})^\top \) for \( i = 1, \ldots, n \). The random variables associated to the \( i \)-th subject or cluster, \( Y_{ij} \) for \( j = 1, \ldots, n_i \), are assumed to satisfy \( \mu_{ij} = E(Y_{ij}), \text{Var}(Y_{ij}) = \frac{\sigma_{ij}}{\omega_{ij}} V(\mu_{ij}) \).
and $\text{Corr}(Y_{ij}, Y_{ik}) = r_{jk}(\rho)$, where $\phi > 0$ is the dispersion parameter, $V(\mu_{ij})$ is the variance function, $\omega_{ij} > 0$ is a known weight, and $\rho = (\rho_1, \ldots, \rho_q)^T$ is a parameter vector. In addition, $\mu_{ij}$ is assumed to be dependent on the regressors vector $x_{ij}$ by $g(\mu_{ij}) = z_{ij} + x_{ij}^T \beta$, where $g(\cdot)$ is the link function, $z_{ij}$ is a known offset and $\beta = (\beta_1, \ldots, \beta_p)^T$ is a vector of regression parameters. The probabilities $\text{Pr}[T_{ij} = 1 | T_{i,j-1} = 1, x_{i1}, \ldots, x_{ij}, Y_{i1}, \ldots, Y_{i,j-1}]$ are estimated by using a logistic model whose covariates are given by $z_1, \ldots, z_r$. Then, those probabilities are used to computed the weights to be included in the parameter estimation algorithm.

A set of standard extractor functions for fitted model objects is available for objects of class glmgee, including methods to the generic functions such as print, summary, model.matrix, estequa, coef, vcov, fitted, confint and predict. The input data are assumed to be ordered in time within each cluster.

**Value**

an object of class wglmgee in which the main results of the weighted GEE model fitted to the data are stored, i.e., a list with components including

- **coefficients** a vector with the estimates of $\beta_1, \ldots, \beta_p$,
- **fitted.values** a vector with the estimates of $\mu_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- **start** a vector with the starting values used,
- **iter** a numeric constant with the number of iterations,
- **prior.weights** a vector with the values of $\omega_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- **offset** a vector with the values of $z_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- **terms** an object containing the terms objects,
- **estfun** a vector with the estimating equations evaluated at the parameter estimates and the observed data,
- **formula** the formula,
- **levels** the levels of the categorical regressors,
- **contrasts** an object containing the contrasts corresponding to levels,
- **converged** a logical indicating successful convergence,
- **model** the full model frame,
- **y** a vector with the values of $y_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$,
- **family** an object containing the family object used,
- **linear.predictors** a vector with the estimates of $g(\mu_{ij})$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$. 
R a matrix with the (robust) estimate of the variance-covariance,
corr a matrix with the estimate of the working-correlation,
corstr a character string specifying the working-correlation structure,
level a character string specifying the weighted GEE method,
id a vector which identifies the subjects or clusters,
sizes a vector with the values of \( n_i \) for \( i = 1, \ldots, n \),
call the original function call,

References

See Also
glmgee, gnmggee

Examples

```r
### Example: Amenorrhea rates over time
data(amenorrhea)
amenorrhea2 <- within(amenorrhea,{
  Ctime <- factor(Time)
  Ctime <- relevel(Ctime,ref="1")
  ylag1 <- c(0,amenorrhea[-length(ID)])
  ylag1 <- ifelse(Time==0,0,ylag1))

mod <- amenorrhea ~ poly(Time,2) + Dose | Ctime + Dose + ylag1

### Observation-specified Weighted GEE
fit1 <- wglmgee(mod, family=binomial, data=amenorrhea2, id=ID,
                 corstr="AR-M-dependent(1)", level="observations")
summary(fit1)

### Cluster-specified Weighted GEE
fit2 <- wglmgee(mod, family=binomial, data=amenorrhea2, id=ID,
                 corstr="AR-M-dependent(1)", level="observations")
summary(fit2)
```
zero.excess

Test for zero-excess in Count Regression Models

Description

Allows to assess if the observed number of zeros is significantly higher than expected according to the fitted count regression model (poisson or negative binomial).

Usage

zero.excess(object, verbose = TRUE)

Arguments

object an object of the class glm, for poisson regression models, or an object of the class overglm, for negative binomial regression models.

verbose an (optional) logical switch indicating if should the report of results be printed. By default, verbose is set to be TRUE.

Details

According to the formulated count regression model, we have that \( Y_k \sim P(y; \mu_k, \phi) \) for \( k = 1, \ldots, n \) are independent random variables. Then, the expected number of zeros is the sum of \( P(0; \hat{\mu}_k, \hat{\phi}) \) for \( k = 1, \ldots, n \), where \( \hat{\mu}_k \) and \( \hat{\phi} \) represent the estimates of \( \mu_k \) and \( \phi \), respectively, obtained from the fitted model. Thus, the test statistic reduces to the standardized difference between the observed and expected number of zeros. The distribution of that statistic, under the null hypothesis, tends to be the standard normal when the sample size, \( n \), tends to infinity.

Value

A matrix with 1 row and the following columns:

| Observed | the observed number of zeros, |
| Expected | the expected number of zeros, |
| z-value  | the value of the statistical test, |
| Pr(>z)   | the p-value of the statistical test. |

See Also

overglm, zeroinf
Examples

####### Example 1: Self diagnosed ear infections in swimmers
    data(swimmers)
    fit1 <- glm(infections ~ frequency + location, family=poisson, data=swimmers)
    zero.excess(fit1)
    fit2 <- overglm(infections ~ frequency + location, family="nb1", data=swimmers)
    zero.excess(fit2)

####### Example 2: Article production by graduate students in biochemistry PhD programs
    bioChemists <- pscl::bioChemists
    fit1 <- glm(art ~ fem + kid5 + ment, family=poisson, data = bioChemists)
    zero.excess(fit1)
    fit2 <- overglm(art ~ fem + kid5 + ment, family="nb1", data = bioChemists)
    zero.excess(fit2)

####### Example 3: Roots Produced by the Columnar Apple Cultivar Trajan
    data(Trajan)
    fit1 <- glm(roots ~ photoperiod, family=poisson, data=Trajan)
    zero.excess(fit1)
    fit2 <- overglm(roots ~ photoperiod, family="nbf", data=Trajan)
    zero.excess(fit2)

zeroalt

Zero-Altered Regression Models to deal with Zero-Excess in Count Data

Description

Allows to fit a zero-altered (Poisson or negative binomial) regression model to deal with zero-excess in count data.

Usage

zeroalt(
    formula,  
data,     
    subset, 
    na.action = na.omit(), 
    weights, 
    family = "poi(log)", 
    zero.link = c("logit", "probit", "cloglog", "cauchit", "log"), 
    reltol = 1e-13, 
    start = list(counts = NULL, zeros = NULL), 
    ...)


Arguments

formula  a Formula expression of the form \( \text{response} \sim x_1 + x_2 + \ldots | z_1 + z_2 + \ldots \), which is a symbolic description of the linear predictors of the models to be fitted to \( \mu \) and \( \pi \), respectively. See Formula documentation. If a formula of the form \( \text{response} \sim x_1 + x_2 + \ldots \) is supplied, the same regressors are employed in both components. This is equivalent to \( \text{response} \sim x_1 + x_2 + \ldots | x_1 + x_2 + \ldots \).

data  an (optional) data frame in which to look for variables involved in the formula expression, as well as for variables specified in the arguments weights and subset.

subset  an (optional) vector specifying a subset of observations to be used in the fitting process.

na.action  a function which indicates what should happen when the data contain NAs. By default na.action is set to be na.omit().

weights  an (optional) vector of positive "prior weights" to be used in the fitting process. The length of weights should be the same as the number of observations. By default, weights is set to be a vector of 1s.

family  an (optional) character string that allows you to specify the distribution to describe the response variable, as well as the link function to be used in the model for \( \mu \). The following distributions are supported: (zero-altered) negative binomial I ("nb1"), (zero-altered) negative binomial II ("nb2"), (zero-altered) negative binomial III ("nbf"), and (zero-altered) poisson ("poi"). Link functions are the same as those available in Poisson models via glm. See family documentation. By default, family is set to be Poisson with log link.

zero.link  an (optional) character string which allows to specify the link function to be used in the model for \( \pi \). Link functions available are the same than those available in binomial models via glm. See family documentation. By default, zero.link is set to be "logit".

reltol  an (optional) positive value which represents the relative convergence tolerance for the BFGS method in optim. By default, reltol is set to be 1e-13.

start  an (optional) list with two components named "counts" and "zeros", which allows to specify the starting values to be used in the iterative process to obtain the estimates of the parameters in the linear predictors of the models for \( \mu \) and \( \pi \), respectively.

Details

The zero-altered count distributions, also called hurdle models, may be obtained as the mixture between a zero-truncated count distribution and the Bernoulli distribution. Indeed, if \( Y \mid \nu = 1 \) is 0 with probability 1 and \( Y \mid \nu = 0 \sim \text{ZTP}(\mu) \), where \( \nu \sim \text{Bernoulli}(\pi) \), then \( Y \) is distributed according to the Zero-Altered Poisson distribution, denoted here as \( \text{ZAP}(\mu, \pi) \).

Similarly, if \( Y \) is a count random variable such that \( Y \mid \nu = 1 \) is 0 with probability 1 and \( Y \mid \nu = 0 \sim \text{ZTNB}(\mu, \phi, \tau) \), where \( \nu \sim \text{Bernoulli}(\pi) \), then \( Y \) is distributed according to the Zero-Altered
Negative Binomial distribution, denoted here as ZANB$(\mu, \phi, \tau, \pi)$. The Zero-Altered Negative Binomial I $(\mu, \phi, \pi)$ and Zero-Altered Negative Binomial II $(\mu, \phi, \pi)$ distributions are special cases of ZANB when $\tau = 0$ and $\tau = -1$, respectively.

The "counts" model may be expressed as $g(\mu_i) = x_i^T \beta$ for $i = 1, \ldots, n$, where $g(\cdot)$ is the link function specified at the argument family. Similarly, the "zeros" model may be expressed as $h(\pi_i) = z_i^T \gamma$ for $i = 1, \ldots, n$, where $h(\cdot)$ is the link function specified at the argument zero.link. Parameter estimation is performed using the maximum likelihood method. The parameter vector $\gamma$ is estimated by applying the routine glm.fit, where a binary-response model (1 or "success" if response=0 and 0 or "fail" if response>0) is fitted. Then, the rest of the model parameters are estimated by maximizing the log-likelihood function based on the zero-truncated count distribution through the BFGS method available in the routine optim. The accuracy and speed of the BFGS method are increased because the call to the routine optim is performed using the analytical instead of the numerical derivatives. The variance-covariance matrix estimate is obtained as being minus the inverse of the (analytical) hessian matrix evaluated at the parameter estimates and the observed data. A set of standard extractor functions for fitted model objects is available for objects of class zeroinflation, including methods to the generic functions such as print, summary, model.matrix, estequa, coef, vcov, logLik, fitted, confint, AIC, BIC and predict. In addition, the model fitted to the data may be assessed using functions such as anova.zeroinflation, residuals.zeroinflation, df-beta.zeroinflation, cooks.distance.zeroinflation and envelope.zeroinflation.

Value

An object of class zeroinflation in which the main results of the model fitted to the data are stored, i.e., a list with components including

- coefficients: a list with elements "counts" and "zeros" containing the parameter estimates from the respective models,
- fitted.values: a list with elements "counts" and "zeros" containing the estimates of $\mu_1, \ldots, \mu_n$ and $\pi_1, \ldots, \pi_n$, respectively,
- start: a vector containing the starting values for all parameters in the model,
- prior.weights: a vector containing the case weights used,
- offset: a list with elements "counts" and "zeros" containing the offset vectors, if any, from the respective models,
- terms: a list with elements "counts", "zeros" and "full" containing the terms objects for the respective models,
- loglik: the value of the log-likelihood function evaluated at the parameter estimates and the observed data,
- estfun: a list with elements "counts" and "zeros" containing the estimating functions evaluated at the parameter estimates and the observed data for the respective models,
- formula: the formula,
levels the levels of the categorical regressors,

contrasts a list with elements "counts" and "zeros" containing the contrasts corresponding to levels from the respective models,

converged a logical indicating successful convergence,

model the full model frame,

y the response count vector,

family a list with elements "counts" and "zeros" containing the family objects used in the respective models,

linear.predictors a list with elements "counts" and "zeros" containing the estimates of $g(\mu_1), \ldots, g(\mu_n)$ and $h(\pi_1), \ldots, h(\pi_n)$, respectively,

R a matrix with the Cholesky decomposition of the inverse of the variance-covariance matrix of all parameters in the model,

call the original function call.

References


See Also

overglm, zeroinf

Examples

####### Example 1: Roots Produced by the Columnar Apple Cultivar Trajan
data(Trajan)
fit1 <- zeroalt(roots ~ photoperiod, family="nbf(log)", zero.link="logit", data=Trajan)
summary(fit1)

####### Example 2: Self diagnosed ear infections in swimmers
data(swimmers)
fit2 <- zeroalt(infections ~ frequency | location, family="nb1(log)", data=swimmers)
summary(fit2)

####### Example 3: Article production by graduate students in biochemistry PhD programs
bioChemists <- pscl::bioChemists
fit3 <- zeroalt(art ~ fem + kid5 + ment, family="nb1(log)", data = bioChemists)
summary(fit3)
Zero-Inflated Regression Models to deal with Zero-Excess in Count Data

Description

Allows to fit a zero-inflated (Poisson or negative binomial) regression model to deal with zero-excess in count data.

Usage

zeroinf(
    formula, data, subset, na.action = na.omit(), weights,
    family = "poi(log)", zero.link = c("logit", "probit", "cloglog", "cauchit", "log"),
    reltol = 1e-13,
    start = list(counts = NULL, zeros = NULL), ...
)

Arguments

formula a Formula expression of the form response ~ x1 + x2 + ... | z1 + z2 + ..., which is a symbolic description of the linear predictors of the models to be fitted to \( \mu \) and \( \pi \), respectively. See Formula documentation. If a formula of the form response ~ x1 + x2 + ... is supplied, then the same regressors are employed in both components. This is equivalent to response ~ x1 + x2 + ... | x1 + x2 + ... .

data an (optional) data frame in which to look for variables involved in the formula expression, as well as for variables specified in the arguments weights and subset.

subset an (optional) vector specifying a subset of observations to be used in the fitting process.

na.action a function which indicates what should happen when the data contain NAs. By default na.action is set to be na.omit().

weights an (optional) vector of positive "prior weights" to be used in the fitting process. The length of weights should be the same as the number of observations. By default, weights is set to be a vector of 1s.
family

an (optional) character string that allows you to specify the distribution to describe the response variable, as well as the link function to be used in the model for $\mu$. The following distributions are supported: (zero-inflated) negative binomial I ("nb1"), (zero-inflated) negative binomial II ("nb2"), (zero-inflated) negative binomial ("nbf"), and (zero-inflated) poisson ("poi"). Link functions are the same as those available in Poisson models via glm. See family documentation. By default, family is set to be Poisson with log link.

zero.link

an (optional) character string which allows to specify the link function to be used in the model for $\pi$. Link functions available are the same than those available in binomial models via glm. See family documentation. By default, zero.link is set to be "logit".

reltol

an (optional) positive value which represents the relative convergence tolerance for the BFGS method in optim. By default, reltol is set to be 1e-13.

start

an (optional) list with two components named "counts" and "zeros", which allows to specify the starting values to be used in the iterative process to obtain the estimates of the parameters in the linear predictors to the models for $\mu$ and $\pi$, respectively.

... further arguments passed to or from other methods.

Details

The zero-inflated count distributions may be obtained as the mixture between a count distribution and the Bernoulli distribution. Indeed, if $Y$ is a count random variable such that $Y | \nu = 1$ is 0 with probability 1 and $Y | \nu = 0 \sim \text{Poisson}(\mu)$, where $\nu \sim \text{Bernoulli}(\pi)$, then $Y$ is distributed according to the Zero-Inflated Poisson distribution, denoted here as ZIP($\mu, \pi$).

Similarly, if $Y$ is a count random variable such that $Y | \nu = 1$ is 0 with probability 1 and $Y | \nu = 0 \sim \text{NB}(\mu, \phi, \tau)$, where $\nu \sim \text{Bernoulli}(\pi)$, then $Y$ is distributed according to the Zero-Inflated Negative Binomial distribution, denoted here as ZINB($\mu, \phi, \tau, \pi$). The Zero-Inflated Negative Binomial I ($\mu, \phi, \pi$) and Zero-Inflated Negative Binomial II ($\mu, \phi, \pi$) distributions are special cases of ZINB when $\tau = 0$ and $\tau = -1$, respectively.

The "counts" model may be expressed as $g(\mu_i) = x_i^T \beta$ for $i = 1, \ldots, n$, where $g(\cdot)$ is the link function specified at the argument family. Similarly, the "zeros" model may be expressed as $h(\pi_i) = z_i^T \gamma$ for $i = 1, \ldots, n$, where $h(\cdot)$ is the link function specified at the argument zero.link. Parameter estimation is performed using the maximum likelihood method. The model parameters are estimated by maximizing the log-likelihood function through the BFGS method available in the routine optim. Analytical derivatives are used instead of numerical derivatives to increase BFGS method accuracy and speed. The variance-covariance matrix estimate is obtained as being minus the inverse of the (analytical) hessian matrix evaluated at the parameter estimates and the observed data.

A set of standard extractor functions for fitted model objects is available for objects of class zeroinflation, including methods for generic functions such as print, summary, model.matrix, esteqa, coef, vcov, logLik, fitted, confint, AIC, BIC and predict. In addition, the model fitted to the data may be assessed using functions such as anova.zeroinflation, residuals.zeroinflation, dbeta.zeroinflation, cooks.distance.zeroinflation and envelope.zeroinflation.
Value

An object of class `zeroinflation` in which the main results of the model fitted to the data are stored, i.e., a list with components including:

- **coefficients**: a list with elements "counts" and "zeros" containing the parameter estimates from the respective models;
- **fitted.values**: a list with elements "counts" and "zeros" containing the estimates of $\mu_1, \ldots, \mu_n$ and $\pi_1, \ldots, \pi_n$, respectively;
- **start**: a vector containing the starting values for all parameters in the model;
- **prior.weights**: a vector containing the case weights used;
- **offset**: a list with elements "counts" and "zeros" containing the offset vectors, if any, from the respective models;
- **terms**: a list with elements "counts", "zeros" and "full" containing the terms objects for the respective models;
- **loglik**: the value of the log-likelihood function evaluated at the parameter estimates and the observed data;
- **estfun**: a list with elements "counts" and "zeros" containing the estimating functions evaluated at the parameter estimates and the observed data for the respective models;
- **formula**: the formula;
- **levels**: the levels of the categorical regressors;
- **contrasts**: a list with elements "counts" and "zeros" containing the contrasts corresponding to levels from the respective models;
- **converged**: a logical indicating successful convergence;
- **model**: the full model frame;
- **y**: the response count vector;
- **family**: a list with elements "counts" and "zeros" containing the family objects used in the respective models;
- **linear.predictors**: a list with elements "counts" and "zeros" containing the estimates of $g(\mu_1), \ldots, g(\mu_n)$ and $h(\pi_1), \ldots, h(\pi_n)$, respectively;
- **R**: a matrix with the Cholesky decomposition of the inverse of the variance-covariance matrix of all parameters in the model;
call the original function call.

References


See Also

`overglm`, `zeroalt`

Examples

###### Example 1: Roots Produced by the Columnar Apple Cultivar Trajan

data(Trajan)

fit1 <- zeroinf(roots ~ photoperiod, family="nbf(log)", zero.link="logit", data=Trajan)

summary(fit1)

###### Example 2: Self diagnosed ear infections in swimmers

data(swimmers)

fit2 <- zeroinf(infections ~ frequency | location, family="nb1(log)", data=swimmers)

summary(fit2)

###### Example 3: Article production by graduate students in biochemistry PhD programs

bioChemists <- pscl::bioChemists

fit3 <- zeroinf(art ~ fem + kid5 + ment | ment, family="nb1(log)", data = bioChemists)

summary(fit3)
Index

* datasets
  advertising, 6
  amenorrhea, 9
  aucuba, 16
  bladder, 16
  brains, 22
  cellular, 23
  cholecystectomy, 24
  coupons, 33
  depression, 34
  dilution, 40
  fabric, 55
  GUIDE, 67
  ldh, 74
  liver, 77
  mammary, 83
  Oranges, 84
  orobanche, 84
  ossification, 85
  pipeline, 91
  rabbits, 95
  races, 96
  richness, 102
  rinse, 103
  Seizures, 106
  shelflife, 109
  skincancer, 110
  spruces, 111
  Steel, 112
  swimmers, 121
  Trajan, 123
  uti, 124
  adjR2, 4
  adjR2.glm, 4
  adjR2.lm, 5
  advertising, 6
  AGPC, 7, 25, 58, 91, 94, 104, 108
  AIC, 137, 140
  amenorrhea, 9
  anova.glmgee, 10, 60, 65
  anova.overglm, 11, 88
  anova.zeroinflation, 13, 137, 140
  anova2, 14
  aucuba, 16
  BIC, 137, 140
  bladder, 16, 77
  BoxTidwell, 17
  BoxTidwell.glm, 18, 21
  BoxTidwell.lm, 19, 20
  brains, 22
  cellular, 23
  cholecystectomy, 24
  CIC, 8, 25, 58, 91, 94, 104, 108
  coef, 137, 140
  confint, 137, 140
  confint2, 26
  cooks.distance.glmgee, 28, 60, 65
  cooks.distance.overglm, 30, 88
  cooks.distance.zeroinflation, 31, 137, 140
  coupons, 33
  depression, 34
  dfbeta.glmgee, 29, 35, 60, 65
  dfbeta.overglm, 30, 37, 88
  dfbeta.zeroinflation, 32, 38, 137, 140
  dilution, 40
  envelope, 40
  envelope.glm, 41, 45, 47, 49
  envelope.lm, 43, 43, 47, 49
  envelope.overglm, 43, 45, 46, 49, 88
  envelope.zeroinflation, 47, 48, 137, 140
  estequa, 50, 137, 140
  estequa.glm, 50
  estequa.glmgee, 51
  estequa.overglm, 53
esterqua.zeroinflation, 54
fabric, 55
family, 59, 61, 64, 66, 87, 89, 131, 132, 136, 138, 140, 141
FisherScoring, 56
fitted, 137, 140
Formula, 136, 139
GHYC, 8, 25, 57, 91, 94, 104, 108
glm, 59, 64, 87, 131, 136, 140
glm.fit, 137
glmgee, 58, 66, 133
gnmgee, 62, 63, 133
GUIDE, 67
gvif, 68
gvif.glm, 68, 70, 72
gvif.lm, 69, 70, 72
gvif.overglm, 71, 88
hlttest, 72
ldh, 74
leverage, 75
leverage.glmgee, 75
liver, 17, 77
localInfluence, 78
localInfluence.glm, 78
localInfluence.glmgee, 60, 79
localInfluence.overglm, 81, 88
logLik, 137, 140
mammary, 83
model.matrix, 137, 140
nls, 64
optim, 87, 88, 136, 137, 140
Oranges, 84
orobanche, 84
ossification, 85
overglm, 86, 134, 138, 142
PAC, 90
pipeline, 91
predict, 137, 140
predict.glmgee, 92
print, 137, 140
QIC, 8, 25, 58, 91, 94, 104, 108
rabbits, 95
races, 96
residuals.glmgee, 60, 65, 97
residuals.overglm, 88, 98
residuals.zeroinflation, 100, 137, 140
residuals2, 101
richness, 102
rinse, 103
RJC, 8, 25, 58, 91, 94, 104, 108
ROCC, 105
Seizures, 106
selfStart, 64
SGPC, 8, 25, 58, 91, 94, 104, 107
shelflife, 109
skincancer, 110
spruces, 111
Steel, 112
stepCriterion, 112
stepCriterion.glm, 113, 117, 119, 121
stepCriterion.glmgee, 60, 114, 115, 119, 121
stepCriterion.lm, 114, 117, 117, 121
stepCriterion.overglm, 88, 114, 117, 119, 119
summary, 137, 140
swimmers, 121
Trajan, 123
uti, 124
vcov, 137, 140
vcov.glmgee, 10, 125
dvttest, 126
dvttest.lm, 127, 129
dvttest.lm, 128, 128
wglmgee, 62, 66, 130
zero.excess, 134
zeroot, 89, 135, 142
zeroinf, 89, 134, 138, 139