Package 'blm'

February 19, 2015

Type Package

Title Binomial linear and linear-expit regression

Version 2013.2.4.4

Date 8-14-2013

Depends R (>= 2.10.1), methods

Imports stats, stats4

Author S. Kovalchik

Maintainer S.Kovalchik <s.a.kovalchik@gmail.com>

Description Implements regression models for binary data on the absolute risk scale. These models are applicable to cohort and population-based case-control data.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-08-16 18:36:19

R topics documented:

`blm-package` *Binomial linear and linear-expit regression model*

Description

The functions `blm` and `lexpit` implement a binomial linear and linear-expit regression model. Estimates are the maximum likelihood estimates with constrained optimization through adaptive barrier method to ensure that estimable probabilities are in the (0,1) interval.

Details

Package:	blm
Type:	Package
Version:	2013.2.4.4
Date:	2013-8-14
Depends:	R (>= 2.10.1), methods
Imports:	stats, stats4
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Maintainer: Stephanie Kovalchik <s.a.kovalchik@gmail.com>

References

Kovalchik S, Varadhan R (2013). Fitting Additive Binomial Regression Models with the R Package blm. *Journal of Statistical Software*, 54(1), 1-18. URL: http://www.jstatsoft.org/v54/i01/.

See Also

constrOptim, blm, lexpit

---

aarp *Nested case-control data set of bladder cancer in the NIH-AARP Diet and Health Study*

Description

The aarp data set is a nested case-control study of bladder cancer outcomes in the NIH-AARP Diet and Health Study. The data set is intended for demonstration purposes only.

Usage

```
aarp
```

Format

bladder70:	indicator of bladder cancer by age 70 years
female:	indicator of female gender
smoke_status:	factor of smoking status (four categories)
w:	inverse of sampling fraction
redmeat:	total daily redmeat consumption (grams/day)
fiber.centered:	total daily fiber consumption (grams), centered on sample median
educ:	factor of education status (six categories)

Source

National Cancer Institute. National Institutes of Health AARP Diet and Health Study. http://dietandhealth.cancer.gov/. Accessed: 12/10/2012

Examples

```
data(aarp)

# ABSOLUTE RISK OF BLADDER CANCER BY 70 YEARS
# FOR DIFFERENT GENDER AND RISK GROUP

fit <- blm(bladder70~female * smoke_status,
      data = aarp,
      weight=aarp$w)

# INTERCEPT IS BASELINE RISK
# ALL OTHER COEFFICIENTS ARE RISK DIFFERENCES FROM BASELINE
```

```
summary(fit)
```

`blm` *Fit a binomial linear regression model*

Description

A direct probability model for regression with a binary outcome from observational data.

Usage

```
blm(formula, data, na.action = na.omit, weights = NULL,
    strata = NULL, par.init = NULL, warn=FALSE,...)
```

Arguments

`formula`	formula for linear model for binary outcome, `event~x1+x2+...`
`data`	data.frame containing the variables of `formula`
`na.action`	function specifying how missing data should be handled, na.action
`weights`	Vector of weights equal to the number of observations. For population-based case-control study, weights are the inverse sampling fractions for controls.
`strata`	vector indicating the stratification for weighted regression with stratified observational data
`par.init`	vector (optional) of initial parameters
`warn`	logical indicator whether to include warnings during algorithm fitting. Default of `FALSE` suppresses warnings when testing for feasible parameters.
`...`	Additional arguments passed to `constrOptim`

Details

The `blm` model coefficients are the solutions to the maximum of a pseudo log-likelihood using a constrained optimization algorithm with an adaptive barrier method, `constrOptim` (Lange, 2010). Variance estimates are based on Taylor linearization (Shah, 2002). When `weights` are not NULL, it is assumed that the study is a case-control design.

Value

Returns an object of class `blm`.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

References

Kovalchik S, Varadhan R (2013). Fitting Additive Binomial Regression Models with the R Package blm. *Journal of Statistical Software*, 54(1), 1-18. URL: http://www.jstatsoft.org/v54/i01/.

Lange, K. (2010) *Numerical Analysis for Statisticians*, Springer.

Shah, BV. (2002) Calculus of Taylor deviations. Joint Statistical Meetings.

See Also

constrOptim

Examples

```
data(ccdata)

fit <- blm(y~female+packyear, weights = ccdata$w,strata=ccdata$strata,
data=ccdata)

summary(fit)


data(aarp)

# ABSOLUTE RISK OF BLADDER CANCER BY 70 YEARS
# FOR DIFFERENT GENDER AND RISK GROUP

fit <- blm(bladder70~female * smoke_status,
      data = aarp,
      weight=aarp$w)

logLik(fit)

# INTERCEPT IS BASELINE RISK
# ALL OTHER COEFFICIENTS ARE RISK DIFFERENCES FROM BASELINE

summary(fit)

# RISK DIFFERENCE CONFIDENCE INTERVALS (PER 1,000 PERSONS)
confint(fit)*1000
```

---

blm-class *Class "blm"*

Description

Class for binomial linear regression (BLM).

Objects from the Class

Objects can be created by calls of the form `new("blm", ...)`.

Slots

`coef`: vector of fitted coefficients

`vcov`: matrix of variance-covariate estimates for `coef`

`formula`: model formula

`df.residual`: residual degrees of freedom

`data`: data frame used in fitting, after applying `na.action`

`which.kept`: vector of index of values in original data source that were used in the model fitting

`y`: response vector for fitted model

`weights`: vector of weights used in model fitting

`strata`: stratification factor for weighted regression.

`converged`: logical message about convergence status at the end of algorithm

`par.init`: initial parameter values for optimization algorithm

`loglik` value of log-likelihood (normalized for weighted likelihood) under full model

`loglik.null` value of log-likelihood (normalized for weighted likelihood) under null model

`barrier.value` value of the barrier function at the optimum

Methods

show `signature(object = "blm")`: Display point estimates of `blm` object.

print `signature(x = "blm",...)`: Display point estimates of `blm` object.

summary `signature(object = "blm",...)`: List of estimates and convergence information.

coef `signature(object = "blm")`: Extractor for fitted coefficients.

logLik `signature(object = "blm")`: Extractor for log-likelihood of `blm` model.

model.formula `signature(object = "blm")`: Extractor for formula of `blm` object.

resid `signature(object = "blm")`: Extractor for residuals.

vcov `signature(object = "blm")`: Extractor for variance-covariance based on Taylor series large-sample Hessian approximation with the pseudo-likelihood of the constrained optimization.

predict `signature(object = "blm")`: Returns vector of linear predictors for each subject of the fitted model.

confint `signature(object = "blm", parm, level = 0.95,...)`: Returns confidence interval (at a given `level`) for the specified regression parameters.

See Also

`blm`, `constrOptim`

ccdata *Simulated case-control dataset*

Description

Simulated population-based case-control dataset

Usage

```
ccdata
```

Format

female:	indicator for female gender
packyear:	discrete variable representing pack-years smoked
strata:	stratification variable
y:	indicator of case status (1 for case, 0 for control)
w:	inverse of sampling fraction

coef *Get coefs from* blm *and* lexpit *objects.*

Description

Extract vector of coefs of the fit of a `blm` or `lexpit` model.

Methods

coef `signature(object = "blm")`: Extractor for MLEs returned as a matrix with one column.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

confint *Confidence intervals for parameters of* blm *and* lexpit *objects.*

Description

Return the confidence intervals for specified parameters and confidence level.

Methods

confint signature(object = "blm", parm, level = 0.95,...): Returns confidence interval (at a given level) for the specified regression parameters.

confint signature(object = "lexpit", parm, level = 0.95,...): Returns confidence interval (at a given level) for the specified regression parameters.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
data(ccdata)

fit <- lexpit(y~female, y~packyear, data = ccdata,
        weight = ccdata$w, strata = ccdata$strata)

confint(fit)
```

crude.risk *Risk-exposure scatter plot*

Description

Calculates the weighted average crude risk against the average exposure level for a continuous exposure. Each point corresponds to overlapping subgroups of 20 percent of the sample ordered from lowest to highest exposure and a sliding window of 1

Usage

```
crude.risk(formula, data, weights = NULL, na.action = na.omit)
```

Arguments

formula	formula specifying the binary outcome and the continuous covariate of interest, e.g. y~x
data	dataframe containing the variables specified in formula
weights	vector of sample weights
na.action	function used for handling missing variables in the variables of formula and weights

Details

The crude.risk function is intended to explore the possible functional relationship between risk and exposure in a non-parametric way.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

See Also

risk.exposure.plot

Examples

```
data(aarp)

risk <- crude.risk(bladder70~redmeat,
  weights = aarp$w,
  data = aarp)

risk.exposure.plot(risk,
   xlab = "Avg. Red Meat Consumption")
```

EO *Compute the ratio of expected event to observed events for* blm *and* lexpit *objects.*

Description

Returns a list of expected to observed counts and the specified confidence interval. The argument group can be used to estimate this ratio by the categories of the categorical variable group. If population-based case-control data is used to fit the model, the expected counts are for the population and make use of the sampling weights.

Usage

```
EO(object, index = NULL, level = 0.95)
```

Arguments

object	object of class blm or lexpit
index	factor for computing E/O comparison by subgroups
level	numeric, confidence level (between 0 and 1) for the E/O ratios

Value

Data frame with:

E expected count

O observed counts

EtoO ratio of expected to observed

lowerCI lower endpoint of confidence interval for E over O ratio

upperCI upper endpoint of confidence interval for E over O ratio

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
data(ccdata)

fit <- blm(y~female+packyear,data = ccdata,
weight = ccdata$w, strata = ccdata$strata)

EO(fit)

EO(fit, ccdata$strata) # BY FACTOR
```

---

expit *Inverse-logit function*

Description

Returns the inverse logit. Where,

$$expit(x) = \frac{\exp(x)}{(1 + \exp(x))}$$

Usage

```
expit(x)
```

Arguments

x	numeric vector

Value

Numeric that is the inverse logit of x.

Examples

```
expit(1:10)
```

---

gof *Hosmer-lemeshow goodness-of-fit statistics for* blm *and* lexpit *objects.*

Description

Computes the deviance and Pearson chi-squared statistics for the fit from a blm or lexpit model. These tests are appropriate when all predictors are categorical and there are many replicates within each covariate class.

Value

Returns a list with table, with expected E and observed O, and the chi-square test chisq and p-value (p.value) for the Pearson goodness-of-fit test. The observed and expected count are listed in the order of the unique levels formed by the design matrix.

When sample weights are present, the goodness-of-fit test is a modified F-test as suggested by Archer et al. (2007).

usage

gof(object)

arguments

object instance of blm or lexpit

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

References

Archer KJ, Lemeshow S, Hosmer DW. Goodness-of-fit tests for logistic regression models when data are collected using a complex sampling design. *Computational Statistics & Data Analysis*. 2007;51:4450–4464.

See Also

blm, lexpit

Examples

```
data(ccdata)

ccdata$packyear <- ccdata$packyear+runif(nrow(ccdata))

# UNWEIGHTED GOF
fit <- blm(y~female+packyear,data = ccdata)
gof(fit)

# WEIGHTED GOF
fit <- blm(y~female+packyear,data = ccdata, weight = ccdata$w)
gof(fit)
```

---

gof.pearson *Pearson's goodness-of-fit statistics for* blm *and* lexpit *objects.*

Description

Computes the deviance and Pearson chi-squared statistics for the fit from a blm or lexpit model. These tests are appropriate when all predictors are categorical and there are many replicates within each covariate class.

Value

Returns a list with expected E and observed O and the chi-square test chisq and p-value (p.value) for the Pearson goodness-of-fit test. The observed and expected count are listed in the order of the unique levels formed by the design matrix.

usage

gof.pearson(object)

arguments

object instance of blm or lexpit

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

See Also

blm, lexpit

Examples

```
data(ccdata)

fit <- blm(y~female+I(packyear>20),data = ccdata,
weight = ccdata$w, strata = ccdata$strata)

gof.pearson(fit)
```

`lexpit` *Fit a linear-expit regression model*

Description

A direct probability model for regression with a binary outcome from observational data. Covariate effects are the sum of additive terms and an expit term, which allows some explanatory variables to be additive and others non-linear.

Usage

```
lexpit(formula.linear,formula.expit,data,na.action=na.omit,
weights=NULL,strata=NULL,par.init=NULL,
warn = FALSE,
control.lexpit=list(max.iter=1000,tol=1E-7),...)
```

Arguments

`formula.linear`	formula for linear model for binary outcome, `event~x1+x2+...`
`formula.expit`	formula for expit model, linear in expit, `event~z1+z2+...`
`data`	data.frame containing the variables of `formula.linear` and `formula.expit`
`na.action`	function specifying how missing data should be handled, na.action
`weights`	Vector of weights equal to the number of observations. For population-based case-control study, weights are the inverse sampling fractions for controls.
`strata`	vector indicating the stratification for weighted regression with stratified observational data
`par.init`	list (optional) of initial parameters for `linear` and `expit` terms.
`warn`	logical indicator whether to include warnings during algorithm fitting. Default of `FALSE` suppresses warnings when testing for feasible parameters.
`control.lexpit`	list with control parameters for optimization algorithm
`...`	Additional arguments passed to `constrOptim`

Details

`lexpit` model uses a two-stage optimization procedure. At the first stage linear terms the solutions to the maximum of a pseudo log-likelihood using a constrained optimization algorithm with an adaptive barrier method, `constrOptim` (Lange, 2010). The second stage maximizes the pseudo log-likelihood with respect to the expit terms using iterative reweighted least squares with an offset term for the linear component of the model.

Variance estimates are based on Taylor linearization (Shah, 2002). When `weights` are not NULL, it is assumed that the study is a case-control design.

Value

Returns an object of class `lexpit`.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

References

Kovalchik S, Varadhan R (2013). Fitting Additive Binomial Regression Models with the R Package blm. *Journal of Statistical Software*, 54(1), 1-18. URL: http://www.jstatsoft.org/v54/i01/.

Lange, K. (2010) *Numerical Analysis for Statisticians*, Springer.

Shah, BV. (2002) Calculus of Taylor deviations. Joint Statistical Meetings.

See Also

constrOptim, nlm

Examples

```
data(ccdata)

fit <- lexpit(y~female,y~packyear,weights = ccdata$w,
        strata=ccdata$strata,data=ccdata)

summary(fit)

# LEXPIT MODEL FOR BLADDER CANCER RISK BY AGE 70
formula.linear <- bladder70~female * smoke_status
formula.expit <- bladder70~redmeat+fiber.centered+I(fiber.centered^2)

# ADDITIVE EFFECTS FOR GENDER AND SMOKING
# LOGISTIC EFFECTS FOR FIBER AND REDMEAT CONSUMPTION
data(aarp)

fit <- lexpit(formula.linear, formula.expit, aarp, weight=aarp$w)
logLik(fit)

model.formula(fit)
```

```
# SUMMARY
summary(fit)
confint(fit)

# FITTED ABSOLUTE RISK PER 1,000 PERSONS
head(predict(fit)*1000)
```

`lexpit-class` *Class "lexpit"*

Description

Class for linear-expit regression (lexpit).

Objects from the Class

Objects can be created by calls of the form `new("lexpit", ...)`.

Slots

`coef.linear`: vector of fitted linear coefficients

`coef.expit`: vector of fitted expit coefficients

`vcov.linear`: matrix of variance-covariate estimates for linear `coef`

`vcov.expit`: matrix of variance-covariate estimates for expit `coef`

`formula.linear`: model formula for linear component

`formula.expit`: model formula for expit component

`df.residual`: residual degrees of freedom

`p`: number of linear parameters

`q`: number of expit parameters

`data`: data frame used in fitting, after applying `na.action`

`which.kept`: vector of index of values in original data source that were used in the model fitting

`y`: response vector for fitted model

`weights`: vector of weights used in model fitting

`strata`: stratification factor for weighted regression.

`converged`: logical message about convergence status at the end of algorithm

`par.init`: initial parameter values for optimization algorithm

`loglik` value of log-likelihood (normalized for weighted likelihood) under full model

`loglik.null` value of log-likelihood (normalized for weighted likelihood) under null model

`barrier.value` value of the barrier function at the optimum

`control.lexpit` list with control parameters for optimization algorithm

Methods

show `signature(object = "lexpit")`: Display point estimates of `lexpit` object.

print `signature(x = "lexpit",...)`: Display point estimates of `lexpit` object.

summary `signature(object = "lexpit",...)`: List of estimates and convergence information.

coef `signature(object = "lexpit")`: Extractor for fitted coefficients.

logLik `signature(object = "lexpit")`: Extractor for log-likelihood of `lexpit` model.

model.formula `signature(object = "lexpit")`: Extractor for formula of `lexpit` object.

vcov `signature(object = "lexpit")`: Extractor for variance-covariance based on Taylor series large-sample Hessian approximation with the pseudo-likelihood of the constrained optimization.

resid `signature(object = "lexpit")`: Extractor for residuals.

predict `signature(object = "lexpit")`: Returns vector of linear predictors for each subject of the fitted model.

confint `signature(object = "lexpit", parm, level = 0.95,...)`: Returns confidence interval (at a given `level`) for the specified regression parameters.

See Also

`lexpit`, `constrOptim`

`logit` *Logit function*

Description

Returns the logit. Where,

$$logit(x) = \log(x/(1-x))$$

Usage

```
logit(x)
```

Arguments

`x`	numeric vector

Value

Numeric that is the logit of `x`.

See Also

`expit`

Examples

```
logit(1:10)
```

`logLik` *Log-likelihood of* `blm` *and* `lexpit` *objects.*

Description

Method to access the log-likelihood of the fitted `blm` or `lexpit` model.

Details

The return object is of the `logLik` class. This method is registered with the stats4 package and can therefore be used with applicable methods like AIC and BIC. Note that when weights are used in the model estimation, the `logLik` is a pseduo-log-likelihood.

Methods

logLik `signature(object = "blm",...)`: Extract log-likelihood. Returns object of `logLik` class.

logLik `signature(object = "lexpit",...)`: Extract log-likelihood. Returns object of `logLik` class.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

See Also

logLik.lm

Examples

```
data(ccdata)

fit <- lexpit(y~female, y~packyear, data = ccdata,
        weight = ccdata$w, strata = ccdata$strata)

logLik(fit)

AIC(fit)
```

LRT *Performs likelihood-ratio test for lexpit and BLM models of cohort data*

Description

Computes the likelihood ratio test for the significance of the specified variable in a lexpit or BLM model fit to cohort data. This method is only valid for study designs that use simple random sampling.

Usage

```
LRT(object, var)
```

Arguments

object	a model of the lexpit or blm class.
var	character name of term.label to be tested

Value

A matrix with the LRT statistic and p-value for the test of the significance of the specified variable given all other variables in the model.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

See Also

constrOptim

Examples

```
cohort <- data.frame(
x1 = runif(500),
x2 = runif(500)
)

cohort$event <- rbinom(n=nrow(cohort),size=1,
prob=0.25+0.1*cohort$x1+.1*cohort$x2)

fit <- blm(event~x1+x2, data=cohort)

summary(fit)

LRT(fit, "x1")
```

```
LRT(fit, "x2")
```

`model.formula` *Get formula call for* `blm` *and* `lexpit` *objects.*

Description

Extract vector of formula of the fit of a `blm` or the formulas for the `lexpit` model.

Methods

model.formula `signature(object = "blm")`: Extractor for formula of `blm` object.

model.formula `signature(object = "lexpit")`: Extractor for formulas of `lexpit` object. Returns a list containing the `linear` and `expit` formulas.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

`predict` *Get risk predictions for* `blm` *and* `lexpit` *objects.*

Description

Computes vector of risk predictions for the dataset used to fit the model. As with method `predict.glm`, standard errors for fitted values can be requested and predictions for the covariates of the data frame `newdata` can be computed rather than the default computation of all fitted values for the data frame used for model fitting.

Methods

predict `signature(object = "blm", newdata, se = FALSE)`: Risk predictions for fit design matrix.

predict `signature(object = "lexpit", newdata, se = FALSE)`: Risk predictions for fit design matrix.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
data(ccdata)

fit <- lexpit(y~female, y~packyear, data = ccdata,
        weight = ccdata$w, strata = ccdata$strata)

predict(fit)[1:10]
```

`print` *Print coefficients of* `blm` *and* `lexpit` *model fit.*

Description

Prints the regression coefficients of the fit of a `blm` or `lexpit` model.

Methods

print `signature(x = "blm")`: Call and coefficient estimates.

print `signature(x = "lexpit")`: Call and coefficient estimates.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

`resid` *Get residuals from* `blm` *and* `lexpit` *objects.*

Description

Extract residuals of model fit.

Methods

resid `signature(object = "blm")`: Extractor for residuals of `blm` object.

resid `signature(object = "lexpit")`: Extractor for residuals of `blm` object.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

`risk.exposure.plot` *Risk-exposure scatter plot*

Description

Calculates the weighted average crude risk against the average exposure level for a continuous exposure. Each point corresponds to overlapping subgroups of 20 percent of the sample ordered from lowest to highest exposure and a sliding window of 1

Usage

```
risk.exposure.plot(object, scale=1,...)
```

Arguments

`object`	list or data.frame with `risk` and `x` covariate. Return object of crude.risk
`scale`	multiplicative factor to modify scale of crude risk estimates
`...`	additional arguments passed to scatter.smooth

Details

The risk-exposure scatter plot is intended to explore the possible functional relationship between risk and exposure.

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
data(aarp)

risk <- crude.risk(bladder70~redmeat,
  weights = aarp$w,
  data = aarp)

risk.exposure.plot(risk,
   xlab = "Avg. Red Meat Consumption")
```

Rsquared *Compute R-squared measures of model fit for* blm *and* lexpit *objects.*

Description

Returns McFadden's unadjusted and adjusted R-squared measures for models of a binary outcome.

Usage

```
Rsquared(object)
```

Arguments

object	object of class blm or lexpit

Value

List of R2 and R2adj.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
data(ccdata)

fit <- blm(y~female+packyear,data = ccdata,
weight = ccdata$w, strata = ccdata$strata)

Rsquared(fit)
```

show *Show* blm *and* lexpit *model fit.*

Description

Print estimates of a blm or lexpit model fit.

Methods

show signature(object = "blm"): Call and coefficient estimates.

show signature(object = "lexpit"): Call and coefficient estimates.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

summary *Summary of* blm *and* lexpit *model fit.*

Description

A list of estimates and convergence status of a blm or lexpit model fit.

Methods

summary signature(object = "blm"): Matrix of estimates and convergence information.

summary signature(object = "lexpit"): Matrix of estimates and convergence information. The matrix returned has the named components:

Est. vector of estimated regression coefficients. For lexpit model estimates are split into est.linear and est.expit components of list

Std. Err standard error of model estimates

t-value t-value of model estimates

p-value p-value (two-sided) of model estimates

Author(s)

S. Kovalchik <s.a.kovalchik@gmail.com>

See Also

blm, lexpit

Examples

```
data(ccdata)

fit <- blm(y~female+packyear,data = ccdata,
weight = ccdata$w, strata = ccdata$strata)

summary(fit)

fit.lexpit <- lexpit(y~female, y~packyear,data = ccdata,
weight = ccdata$w, strata = ccdata$strata)

summary(fit.lexpit)
```

vcov *Get variance-covariance from* blm *and* lexpit *objects.*

Description

Returns Hessian-based variance-covariance matrix of the fit of a blm or lexpit model. If any constraints are active, only the augmented Lagrangian takes this into account in the Hessian computation, so if augmented is FALSE, i.e. the adaptive barrier method of optimization is used, the covariance-variance might be inaccurate.

Methods

vcov signature(object = "blm"): Extractor for variance-covariance of MLEs.

vcov signature(object = "lexpit"): Extractor for variance-covariance of MLEs.

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

which.at.boundary *Covariate patterns at the boundary for* blm *and* lexpit *objects.*

Description

Returns matrix of covariate types with a predicted probability at the lower or upper boundary defined by the specified criterion or NA if no boundary constraints are present.

Value

Returns all rows of design matrix whose predicted risk are less than or equal to criterion or greater than or equal to 1 - criterion.

usage

which.at.boundary (object, criterion = 1e-06)

arguments

object model fit of class blm or lexpit

criterion numeric distance from 0 (or 1) that is considered to be at the boundary

Author(s)

Stephanie Kovalchik <s.a.kovalchik@gmail.com>

Examples

```
data(ccdata)

fit <- blm(y~female+packyear,data = ccdata,
weight = ccdata$w, strata = ccdata$strata)

which.at.boundary(fit)
```

Index