Package 'oglmx'

May 5, 2018

Type Package

Title Estimation of Ordered Generalized Linear Models

Version 3.0.0.0

Date 2018-05-05

Author Nathan Carroll

Maintainer Nathan Carroll <nathan.carroll@ur.de>

Description

Ordered models such as ordered probit and ordered logit presume that the error variance is constant across observations. In the case that this assumption does not hold estimates of marginal effects are typically biased (Weiss (1997)). This package allows for generalization of ordered probit and ordered logit models by allowing the user to specify a model for the variance. Furthermore, the package includes functions to calculate the marginal effects. Wrapper functions to estimate the standard limited dependent variable models are also included.

License GPL-2

Depends maxLik

Imports stats

Suggests glmx, lmtest

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-05 11:24:43 UTC

R topics documented:

`oglmx-package` *Estimation of Ordered Generalized Linear Models Package for estimation of ordered generalized linear models.*

Description

Ordered models such as ordered probit and ordered logit presume that the error variance is constant across observations. In the case that this assumption does not hold estimates of marginal effects are typically biased (Weiss (1997)). This package allows for generalization of ordered probit and ordered logit models by allowing the user to specify a model for the variance. Furthermore, the package includes functions to calculate the marginal effects. Wrapper functions to estimate the standard limited dependent variable models are also included.

Details

Package:	oglmx
Type:	Package
Title:	Estimation of Ordered Generalized Linear Models
Version:	3.0.0.0
Date:	2018-05-05
Author:	Nathan Carroll
Maintainer:	Nathan Carroll <nathan.carroll@ur.de>
Description:	Ordered models such as ordered probit and ordered logit presume that the error variance is constant across obse
License:	GPL-2
Depends:	maxLik
Imports:	stats
Suggests:	glmx, lmtest

Index of help topics:

```
AIC.oglmx               Calculate Akaike Information Criterion
```

Author(s)

Nathan Carroll

Maintainer: Nathan Carroll <nathan.carroll@ur.de>

AIC.oglmx *Calculate Akaike Information Criterion*

Description

Calculates the Akaike Information Criterion for objects of class `oglmx`. Calculate using the formula $-2*loglikelihood + k*npar$ where $npar$ represents the number of parameters in the model and k is the cost of additional parameters, equal to 2 for the AIC, it is $k = \log(n)$ with n the number of observations for the BIC.

Usage

```
  ## S3 method for class 'oglmx'
AIC(object, ..., k = 2)
```

Arguments

`object`	object of class `oglmx`
`...`	additional arguments. Currently ignored.
`k`	the penalty per parameter to be used.

Details

When comparing models by maximium likelihood estimation the smaller the value of the AIC the better.

Value

A numeric value with the AIC.

Author(s)

Nathan Carroll, `<nathan.carroll@ur.de>`

See Also

`AIC`.

---

continuous.margin.mean *Calculate marginal effects for continuous variables.*

Description

Calculate marginal effects for continuous variables. Functions calculate for variables in the mean equation and in the variance equation, for a variable in both equations the effects should be summed.

Usage

```
continuous.margin.mean(paramvec,etas,link,std.dev)

continuous.margin.sd(paramvec,etas,link,std.dev,gstd.dev)
```

Arguments

paramvec	Coefficients related to variables for which marginal effects are desired.
etas	Inputs to link functions.
link	specifies the link function for the estimated model.
std.dev	The calculated standard deviation of the error terms.
gstd.dev	The calculated derivative of the standard deviation of the error terms.

Value

Numeric vector of marginal effects.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

margins.oglmx

---

discrete.margin_meanonly

Calculate marginal effects for binary variables.

Description

Calculate marginal effects for binary variables. Functions calculate for variables that are only in the mean equation, only in the variance equation, and variables in both.

Usage

```
discrete.margin_meanonly(beta, X, whichVars, etas, link, std.dev)

discrete.margin_varonly(delta, Z, whichVars, sdmodel, etas, link, std.dev)

discrete.margin_both(beta, X, delta, Z, BothEqLocs, sdmodel, etas, link, std.dev)
```

Arguments

beta	Coefficients for the mean equation.
X	Variable values for the mean equation.
whichVars	Numeric vector stating indexes of variables that are binary and marginal effects are desired.
etas	Inputs to link functions.
link	specifies the link function for the estimated model.

`std.dev`	The calculated standard deviation of the error terms.
`delta`	Coefficients for the variance equation.
`Z`	Variable values for the variance equation.
`sdmodel`	Expression used to calculate standard deviation.
`BothEqLocs`	Dataframe describing locations of binary variables that are in both the mean and variance equations.

Value

Numeric vector of marginal effects. Has as attributes calculated components that are used to calculate derivatives of marginal effects.

Author(s)

Nathan Carroll, `<nathan.carroll@ur.de>`

See Also

`margins.oglmx`

D_continuous.margin.mean_mean

Calculate derivatives of marginal effects for continuous variables.

Description

Calculates derivatives of marginal effects with respect to the estimated parameters for variables that are treated as continuous. Required to calculate standard errors of marginal effects.

Usage

```
D_continuous.margin.mean_mean(whichMargins, whichXest, X, paramvec, etas,
                              link, std.dev)

D_continuous.margin.mean_var(Z, paramvec, etas, link, std.dev, gstd.dev)

D_continuous.margin.mean_alpha(estThresh, outcomematrix, paramvec, etas,
                               link,std.dev)

D_continuous.margin.var_mean(X, paramvec, etas, link,std.dev, gstd.dev)

D_continuous.margin.var_var(whichMargins, whichZest, Z, paramvec, etas,
                            link, std.dev, gstd.dev, hstd.dev)

D_continuous.margin.var_alpha(estThresh, outcomematrix, paramvec, etas,
                              link, std.dev, gstd.dev)
```

Arguments

whichMargins	Numeric vector indicating indexes of parameters in the relevant matrix for which margins are desired.
whichXest	Logical vector indicating the variables in X for which the relevant parameters were estimated.
X	Data matrix containing variables in mean equation.
paramvec	Coefficients related to variables for which marginal effects are desired.
etas	Inputs to link functions.
link	specifies the link function for the estimated model.
std.dev	The calculated standard deviation of the error terms.
Z	Data matrix containing variables in variance equation.
whichZest	Logical vector indicating the variables in Z for which the relevant parameters were estimated.
gstd.dev	The calculated derivative of the standard deviation of the error terms.
hstd.dev	The calculated second derivative of the standard deviation of the error terms.
estThresh	Logical vector indicating which threshold parameters were estimated.
outcomematrix	A matrix that indicates the outcome variable.

Value

Numeric matrix of derivatives of marginal effects with respect to estimated parameters.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

margins.oglmx

D_discrete.margin_meanonly.mean

Calculate derivatives of marginal effects for binary variables.

Description

Calculates derivatives of marginal effects with respect to the estimated parameters for binary variables. Required to calculate standard errors of marginal effects.

Usage

```
D_discrete.margin_meanonly.mean(whichVars, whichXest, X, fouretas, link, std.dev)

D_discrete.margin_mean.var(whichZest, Z, fouretas, link, std.dev, gstd.dev)

D_discrete.margin_mean.alpha(estThresh, outcomematrix, fouretas, std.dev, link)

D_discrete.margin_var.mean(whichXest, X, fouretas, link, StdDevs)

D_discrete.margin_varonly.var(whichVars, whichZest, Z,fouretas, ZDinputs, link,
                              StdDevs, gsdmodel)

D_discrete.margin_var.alpha(estThresh, outcomematrix, fouretas, StdDevs, link)

D_discrete.margin_meanvar.mean(whichXest, X, BothEqLocs, fouretas, StdDevs, link)

D_discrete.margin_meanvar.var(whichZest, Z, BothEqLocs, fouretas, ZDinputs, link,
                              StdDevs,gsdmodel)
```

Arguments

whichVars	Numeric vector stating indexes of variables that are binary and marginal effects are desired.
whichXest	Logical vector indicating the variables in X for which the relevant parameters were estimated.
X	Data matrix containing variables in mean equation.
fouretas	Inputs to link functions.
link	specifies the link function for the estimated model.
std.dev	The calculated standard deviation of the error terms.
Z	Data matrix containing variables in variance equation.
whichZest	Logical vector indicating the variables in Z for which the relevant parameters were estimated.
gstd.dev	The calculated derivative of the standard deviation of the error terms.
estThresh	Logical vector indicating which threshold parameters were estimated.
outcomematrix	A matrix that indicates the outcome variable.
ZDinputs	Values of inputs to function that gives standard deviation when binary variable is equal to 0 and 1.
StdDevs	Values of standard deviation when binary variable is equal to 0 and 1.
gsdmodel	Expression used to calculate derivative of standard deviation.
BothEqLocs	Dataframe describing locations of binary variables that are in both the mean and variance equations.

Value

Numeric matrix of derivatives of marginal effects with respect to estimated parameters.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

`margins.oglmx`

`formula.oglmx` *Obtain model formula for an* `oglmx` *object.*

Description

Given an object of class `oglmx` the function describes the estimated model via an expression of class `formula`. The function serves to provide a name of a model to the `lrtest` function in the `lmtest` package.

Usage

```
## S3 method for class 'oglmx'
formula(x, ... )
```

Arguments

`x`	object of class `oglmx`.
`...`	additional arguments, currently ignored.

Value

an object of class `formula`.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

`oglmx`, codelrtest, codeformula.

`getEtas` *Construct ingredients for probability calculation.*

Description

The probability of a particular outcome j for observation i is given by:

$$F\left(\frac{\alpha_{j+1} - x_i\beta}{g\left(z_i\delta\right)}\right) - F\left(\frac{\alpha_j - x_i\beta}{g\left(z_i\delta\right)}\right)$$

where F is the link function, the αs refer to threshold values and g is the function that describes the model for the variance. This function calculates the two inputs to the link function in the above expression given precalculated values of the mean of the latent variable given parameters and the standard deviation given parameters.

Usage

```
getEtas(thresholds,xb,std.dev)

getEtas.Exp(thresholds,xb_matrix,sd_matrix)
```

Arguments

`thresholds`	Numeric matrix of dimension (number of observations * 2). Columns refer to the right and left threshold corresponding to the desired outcome.
`xb, xb_matrix`	Numeric vector/matrix of expected values of the latent variable.
`std.dev, sd_matrix`	Numeric vector/matrix of standard deviations of the error term given variables.

Value

`eta_1`	Numeric vector/matrix corresponding to the right threshold.
`eta_0`	Numeric vector/matrix corresponding to the left threshold.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

`oglmx`

InternalFunctions *Various functions not intended for user.*

Description

Functions used in the process of estimating parameters and standard errors of ordered generalized linear models.

Usage

```
updateComponents(Env,Parameters)

oglmx.maxlik(inputenv,start)

loglikelihood.oglmx(Env)

score_oglmx(Env)

hessian_oglmx(Env)

calcBHHHmatrix(Env)

mergeformulas(formula1,formula2)

calcstartvalues(whichparameter,gfunc,threshvec)

getThresholds(outcomematrix,thresholdvector)

Probability(eta_1,eta_0,link)
```

Arguments

`Env, inputenv`	environment, typically constructed by the `oglmx.fit` function, that contains all relevant information for the optimisation process.
`Parameters, start`	numeric vector of length equal to the number of estimated parameters.
`formula1, formula2`	items of class `formula`.
`whichparameter`	logical
`gfunc`	expression, function used to model the variance
`threshvec, thresholdvector`	numeric vectors of threshold values
`outcomematrix`	numeric matrix with binary variables indicating the outcome for each observation

eta_1,eta_0	input values for the link function
link	string value indicating which link function is to be used

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

oglmx, getEtas

---

logit.reg *Fit Logit Model.*

Description

Wrapper function for oglmx to estimate the binary response logit model.

Usage

```
logit.reg(formula, data, start = NULL, weights=NULL, beta = NULL,
          analhessian = TRUE, na.action, savemodelframe = FALSE, robust = FALSE)
```

Arguments

formula	an object of class formula: a symbolic description of the model used to explain the mean of the latent variable. The response variable should be a numeric vector or factor variable with two values.
data	a data frame containing the variables in the model.
start	either NULL or a numeric vector specifying start values for each of the estimated parameters, passed to the maximisation routine.
weights	either NULL or a numeric vector of length equal to the number of rows in the data frame. Used to apply weighted maximum likelihood estimation.
beta	NULL or numeric vector. Used to prespecify elements of the parameter vector for the equation of the mean of the latent variable. Vector should be of length one or of length equal to the number of explanatory variables in the mean equation. If of length one the value is presumed to correspond to the constant. If of length greater than one then NA should be entered for elements of the vector to be estimated.
analhessian	logical. Indicates whether the analytic Hessian should be calculated and used, default is TRUE, if set to FALSE a finite-difference approximation of the Hessian is used.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The factory-fresh default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.

`savemodelframe`	logical. Indicates whether the model frame(s) should be saved for future use. Default is `FALSE`. Should be set to `TRUE` if intending to estimate Average Marginal Effects.
`robust`	logical. If set to `TRUE` the outer product or BHHH estimate of the meat in the sandwich of the variance-covariance matrix is calculated. If calculated standard errors will be calculated using the sandwich estimator by default when calling `summary`.

Value

object of class "`oglmx`", see `oglmx`.

Author(s)

Nathan Carroll, <`nathan.carroll@ur.de`>

See Also

`glm` for alternative method to estimate a logit model. `oglmx`. To obtain marginal effects see `margins.oglmx`.

`logLik.oglmx` *Extract log likelihood value*

Description

Return the log likelihood value for objects of class `oglmx` and `summary.oglmx`

Usage

```
## S3 method for class 'oglmx'
logLik(object, ... )

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

Arguments

`object`	object of class `oglmx` or `summary.oglmx`.
`...`	additional arguments, currently ignored.

Value

A single numeric value, the log likelihood for the estimated model.

Author(s)

Carroll, Nathan <`nathan.carroll@ur.de`>

See Also

logLik, oglmx

---

margins.oglmx *Calculate marginal effects for* oglmx *objects.*

Description

This function constructs marginal effects and calculates standard errors for all models estimated by the oglmx function. Standard errors are obtained using the delta method.

Usage

```
margins.oglmx(object, atmeans = TRUE, AME = FALSE, location = NULL, outcomes = "All",
              ascontinuous = FALSE, Vars = NULL)

## S3 method for class 'margins.oglmx'
print(x, ... )
```

Arguments

object	object of class "oglmx".
Vars	vector specifying variables for which marginal effects are desired.
outcomes	either character string "All", the default option, or a numeric vector indicating the outcomes for which the marginal effect is desired.
atmeans	logical. If TRUE then the marginal effects are calculated at the means of the variables in the equations for the mean and variance of the latent variable.
AME	logical. If TRUE the marginal effects are averaged across observations.
ascontinuous	logical. If TRUE binary variables are treated as if continuous to calculate marginal effects.
location	NULL, a numeric vector, or a list containing two numeric vectors. Allows the user to specify the values of the explanatory variables at which the marginal effect is to be calculated. For a homoskedastic model the input should be a numeric vector of length equal to the number of variables in the model matrix. For a heterskedastic model the input should be a list, the first element should be a vector of length equal to the number of variables in the mean equation and the second is a vector of length equal to the number of variables in the variance equation.
...	additional arguments to print method. Currently ignored.
x	object of class margins.oglmx.

Value

an object of class `margins.oglmx`. The object consists of a list containing data matrices, each matrix corresponding to an outcome for which the marginal effect was desired. Columns of each matrix correspond to the estimated marginal effect, its standard error, t-statistics and two sided p-value.

Author(s)

Nathan Carroll, `<nathan.carroll@ur.de>`

`McFaddensR2.oglmx` *Calculate McFadden's R-Squared.*

Description

Model evaluation methods based on the analogue of squared residuals do not work well when the outcome variable is discrete and ordered. A popular pseudo-R^2 measure due to McFadden (1973) is given by:

$$R^2 = 1 - \log L_{fit} / \log L_0$$

where $\log L_{fit}$ is the log-likelihood for the fitted model and $\log L_0$ is the log-likelihood from an intercept only model that estimates the probability of each alternative to be the sample average. This function calculates this term for objects of class `oglmx`.

Usage

```
McFaddensR2.oglmx(object)
```

Arguments

`object`	object of type `oglmx`

Value

numeric value between 0 and a theoretical maximum of 1.

oglmx *Fit Ordered Generalized Linear Model.*

Description

oglmx is used to estimate models for which the outcome variable is discrete and the mean and/or variance of the underlying latent variable can be modelled as a linear combination of explanatory variables. Standard models such as probit, logit, ordered probit and ordered logit are included in the diverse set of models estimated by the function.

Usage

```
oglmx(formulaMEAN, formulaSD=NULL, data, start=NULL, weights=NULL,
      link="probit", constantMEAN=TRUE, constantSD=TRUE, beta=NULL,
      delta=NULL, threshparam=NULL, analhessian=TRUE,
      sdmodel=expression(exp(z)), SameModelMEANSD=FALSE, na.action,
      savemodelframe=TRUE, Force=FALSE, robust=FALSE)

oglmx.fit(outcomeMatrix, X, Z, w, beta, delta, threshparam, link, start,
          sdmodel, optmeth="maxLik", analhessian, robust)
```

Arguments

formulaMEAN	an object of class formula: a symbolic description of the model used to explain the mean of the latent variable. The response variable should be a numeric vector or factor variable such that the numerical assignments for the levels of the factor have ordinal meaning.
formulaSD	either NULL or an object of class formula: a symbolic description of the model used to explain the variance of the latent variable.
data	a data frame containing the variables in the model.
start	either NULL or a numeric vector specifying start values for each of the estimated parameters, passed to the maximisation routine.
weights	either NULL or a numeric vector of length equal to the number of rows in the data frame. Used to apply weighted maximum likelihood estimation.
link	specifies a link function for the model to be estimated, accepted values are "probit", "logit", "cauchit", "loglog" and "cloglog"
constantMEAN	logical. Should an intercept be included in the model of the mean of the latent variable? Can be overwritten and set to FALSE using the formulaMEAN argument by writing 0 + as the first element of the equation.
constantSD	logical. Should an intercept be included in the model of the variance of the latent variable? Can be overwritten and set to FALSE using the formulaSD argument by writing 0 + as the first element of the equation.
beta	NULL or numeric vector. Used to prespecify elements of the parameter vector for the equation of the mean of the latent variable. Vector should be of length one

or of length equal to the number of explanatory variables in the mean equation. If of length one the value is presumed to correspond to the constant if a constant is included or the first element of the parameter vector. If of length greater than one then `NA` should be entered for elements of the vector to be estimated.

`delta` — `NULL` or numeric vector. Used to prespecify elements of the parameter vector for the equation of the variance of the latent variable. Vector should be of length one or of length equal to the number of explanatory variables in the variance equation. If of length one the value is presumed to correspond to the constant if a constant is included or the first element of the parameter vector. If of length greater than one then `NA` should be entered for elements of the vector to be estimated.

`threshparam` — `NULL` or numeric vector. Used to prespecify the threshold parameters of the model. Vector should be of length equal to the number of outcomes minus one. `NA` should be entered for threshold parameters to be estimated by the model.

`analhessian` — logical. Indicates whether the analytic Hessian should be calculated and used, default is TRUE, if set to FALSE a finite-difference approximation of the Hessian is used.

`sdmodel` — object of mode "`expression`". The expression defines function that transforms the linear model for the standard deviation into the standard deviation. The expression should be written as a function of variable `z`. The default value is `expression(exp(z))`.

`SameModelMEANSD` — logical. Indicates whether the matrix used to model the mean of the latent variable is identical to that used to model the variance. If `formulaSD=NULL` and `SameModelMEANSD=TRUE` a model with heteroskedasticity is estimated. If `SameModelMEANSD=FALSE` and `formulaSD==formulaMEAN` value is overridden. Used to reduce memory requirements when models are identical.

`na.action` — a function which indicates what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The factory-fresh default is `na.omit`. Another possible value is `NULL`, no action. Value `na.exclude` can be useful.

`savemodelframe` — logical. Indicates whether the model frame(s) should be saved for future use. Default is `FALSE`. Should be set to `TRUE` if intending to estimate Average Marginal Effects.

`Force` — logical. If set to `FALSE` (the default) the function stops if the response variable has more than twenty categories. Should be changed to `TRUE` if a model with more than twenty categories is desired.

`robust` — logical. If set to `TRUE` the outer product or BHHH estimate of the meat in the sandwich of the variance-covariance matrix is calculated. If calculated standard errors will be calculated using the sandwich estimator by default when calling `summary`.

`outcomeMatrix, X, Z` — `X` is a data matrix for the right hand side of the mean equation, `outcomeMatrix` is a matrix that indicates the outcome variable and `Z` is a data matrix for the variance equation.

`w` — `w` specifies a vector of weights for the `oglmx.fit` function.

`optmeth` — `optmeth` specifies a method for the maximisation of the likelihood, currently "maxLik" is the only available option.

Value

An object of class "oglmx" with the following components:

`link`	link function used in the estimated model.
`sdmodel`	Expression for the model for the standard deviation, default is exp(z).
`call`	the call used to generate the results.
`factorvars`	vector listing factor variables included in the model
`Outcomes`	numeric vector listing the values of the different outcomes.
`NoVarModData`	dataframe. Contains data required to estimate the no information model used in calculation of McFadden's R-squared measure.
`NOutcomes`	the number of distinct outcomes in the response variable.
`Hetero`	logical. If `TRUE` indicates that the estimated model includes a model for the variance of the error term, i.e. heteroskedasticity.
`formula`	two element list. Each element is an object of type `formula` related to the mean and standard deviation equation respectively.
`modelframes`	If `savemodelframe` set to `FALSE` then returns `NULL`, otherwise returns a list with two elements, the model frames for the mean and variance equations.
`BothEq`	Omitted in the case of a homoskedastic model. Dataframe listing variables that are contained in both the mean and variance equations.
`varMeans`	a list containing two numeric vectors. The vectors list the mean values of the variables in the mean and variance equation respectively. Stored for use in a call of `margins.oglmx` to obtain marginal effects at means.
`varBinary`	a list containing two numeric vectors. The vectors indicate whether the variables in the mean and variance equations are binary indicators. Stored for use in a call of `margins.oglmx` to obtain marginal effects at means.
`loglikelihood`	log-likelihood for the estimated model. Includes as attributes the log-likelihood for the constant only model and the number of observations.
`coefficients`	vector of estimated parameters.
`gradient`	numeric vector, the value of the gradient of the log-likelihood function at the obtained parameter vector. Should be approximately equal to zero.
`no.iterations`	number of iterations of maximisation algorithm.
`returnCode`	code returned by the `maxLik` optimisation routine. For details of meaning see `maxNR`.
`hessian`	hessian matrix of the log-likelihood function evaluated at the obtained parameter vector.
`allparams`	a list containing three numeric vectors, the vectors contain the parameters from the mean equation, the variance equation and the threshold parameters respectively. Includes the prespecified and estimated parameters together.
`Est.Parameters`	list containing three logical vectors. Indicates which parameters in the parameter vectors were estimated.
`BHHHhessian`	Omitted if `robust = FALSE` and weights were not included. The BHHH variance-covariance estimate.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

References

Cameron, A. C. & Trivedi, P. K. (2005) *Microeconometrics : methods and applications* Cambridge University Press

Wooldridge, J. M. (2002) *Econometric analysis of cross section and panel data* The MIT Press

See Also

maxLik, margins.oglmx, polr.

Examples

```
# create random sample, three variables, two binary.
set.seed(242)
n<-250
x1<-sample(c(0,1),n,replace=TRUE,prob=c(0.75,0.25))
x2<-vector("numeric",n)
x2[x1==0]<-sample(c(0,1),n-sum(x1==1),replace=TRUE,prob=c(2/3,1/3))
z<-rnorm(n,0.5)
# create latent outcome variable
latenty<-0.5+1.5*x1-0.5*x2+0.5*z+rnorm(n,sd=exp(0.5*x1-0.5*x2))
# observed y has four possible values: -1,0,1,2
# threshold values are: -0.5, 0.5, 1.5.
y<-vector("numeric",n)
y[latenty< -0.5]<--1
y[latenty>= -0.5 & latenty<0.5]<- 0
y[latenty>= 0.5 & latenty<1.5]<- 1
y[latenty>= 1.5]<- 2
dataset<-data.frame(y,x1,x2)
# estimate standard ordered probit
results.oprob<-oglmx(y ~ x1 + x2 + z, data=dataset,link="probit",constantMEAN=FALSE,
                     constantSD=FALSE,delta=0,threshparam=NULL)
coef(results.oprob) # extract estimated coefficients
summary(results.oprob)
# calculate marginal effects at means
margins.oglmx(results.oprob)
# estimate ordered probit with heteroskedasticity
results.oprobhet<-oglmx(y ~ x1 + x2 + z, ~ x1 + x2, data=dataset, link="probit",
                     constantMEAN=FALSE, constantSD=FALSE,threshparam=NULL)
summary(results.oprobhet)
library("lmtest")
# likelihood ratio test to compare model with and without heteroskedasticity.
lrtest(results.oprob,results.oprobhet)
# calculate marginal effects at means.
margins.oglmx(results.oprobhet)
# scale of parameter values is meaningless. Suppose instead two of the
# three threshold values were known, then can include constants in the
# mean and standard deviation equation and the scale is meaningful.
```

```
results.oprobhet1<-oglmx(y ~ x1 + x2 + z, ~ x1 + x2, data=dataset, link="probit",
                        constantMEAN=TRUE, constantSD=TRUE,threshparam=c(-0.5,0.5,NA))
summary(results.oprobhet1)
margins.oglmx(results.oprobhet1)
# marginal effects are identical to results.oprobithet, but using the true thresholds
# means the estimated parameters are on the same scale as underlying data.
# can choose any two of the threshold values and get broadly the same result.
results.oprobhet2<-oglmx(y ~ x1 + x2 + z, ~ x1 + x2, data=dataset, link="probit",
                         constantMEAN=TRUE, constantSD=TRUE,threshparam=c(-0.5,NA,1.5))
summary(results.oprobhet2)
margins.oglmx(results.oprobhet2)
# marginal effects are again identical. Parameter estimates do change.
```

ologit.reg *Fit an ordered Logit model.*

Description

Wrapper function for oglmx to estimate an ordered Logit model.

Usage

```
ologit.reg(formula, data, start = NULL, weights=NULL, beta = NULL, threshparam = NULL,
           analhessian = TRUE, na.action, savemodelframe = FALSE, robust = FALSE,
            Force = FALSE)
```

Arguments

formula	an object of class formula: a symbolic description of the model used to explain the mean of the latent variable. The response variable should be a numeric vector or factor variable such that the numerical assignments for the levels of the factor have ordinal meaning.
data	a data frame containing the variables in the model.
start	either NULL or a numeric vector specifying start values for each of the estimated parameters, passed to the maximisation routine.
weights	either NULL or a numeric vector of length equal to the number of rows in the data frame. Used to apply weighted maximum likelihood estimation.
beta	NULL or numeric vector. Used to prespecify elements of the parameter vector for the equation of the mean of the latent variable. Vector should be of length one or of length equal to the number of explanatory variables in the mean equation. NA should be entered for elements of the vector to be estimated.
threshparam	numeric vector. Used to prespecify the threshold parameters of the model. Vector should be of length equal to the number of outcomes minus one. NA should be entered for threshold parameters to be estimated by the model.

analhessian	logical. Indicates whether the analytic Hessian should be calculated and used, default is TRUE, if set to FALSE a finite-difference approximation of the Hessian is used.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The factory-fresh default is `na.omit`. Another possible value is `NULL`, no action. Value `na.exclude` can be useful.
savemodelframe	logical. Indicates whether the model frame(s) should be saved for future use. Default is `FALSE`. Should be switched to `TRUE` if intending to estimate Average Marginal Effects.
robust	logical. If set to `TRUE` the outer product or BHHH estimate of the meat in the sandwich of the variance-covariance matrix is calculated. If calculated standard errors will be calculated using the sandwich estimator by default when calling `summary`.
Force	logical. If set to `FALSE` (the default) the function stops if the response variable has more than twenty categories. Should be changed to `TRUE` if a model with more than twenty categories is desired.

Details

object of class "`oglmx`", see `oglmx`.

Value

object of class "`oglmx`", see `oglmx`.

Author(s)

Nathan Carroll, `<nathan.carroll@ur.de>`

See Also

`polr` for alternative method to estimate an ordered logit model. `oglmx`. To obtain marginal effects see `margins.oglmx`.

---

oprobit.reg *Fit Ordered Probit Model.*

Description

Wrapper function for `oglmx` to estimate an ordered Probit model.

Usage

```
oprobit.reg(formula, data, start = NULL, weights=NULL, beta = NULL, threshparam = NULL,
          analhessian = TRUE, na.action, savemodelframe = FALSE, robust = FALSE,
            Force = FALSE)
```

Arguments

`formula`	an object of class `formula`: a symbolic description of the model used to explain the mean of the latent variable. The response variable should be a numeric vector or factor variable such that the numerical assignments for the levels of the factor have ordinal meaning.
`data`	a data frame containing the variables in the model.
`start`	either `NULL` or a numeric vector specifying start values for each of the estimated parameters, passed to the maximisation routine.
`weights`	either `NULL` or a numeric vector of length equal to the number of rows in the data frame. Used to apply weighted maximum likelihood estimation.
`beta`	`NULL` or numeric vector. Used to prespecify elements of the parameter vector for the equation of the mean of the latent variable. Vector should be of length one or of length equal to the number of explanatory variables in the mean equation. If of length one the value is presumed to correspond to the first element of the parameter vector. If of length greater than one then `NA` should be entered for elements of the vector to be estimated.
`threshparam`	`NULL` or numeric vector. Used to prespecify the threshold parameters of the model. Vector should be of length equal to the number of outcomes minus one. `NA` should be entered for threshold parameters to be estimated by the model.
`analhessian`	logical. Indicates whether the analytic Hessian should be calculated and used, default is TRUE, if set to FALSE a finite-difference approximation of the Hessian is used.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The factory-fresh default is `na.omit`. Another possible value is `NULL`, no action. Value `na.exclude` can be useful.
`savemodelframe`	logical. Indicates whether the model frame(s) should be saved for future use. Default is `FALSE`. Should be switched to `TRUE` if intending to estimate Average Marginal Effects.
`robust`	logical. If set to `TRUE` the outer product or BHHH estimate of the meat in the sandwich of the variance-covariance matrix is calculated. If calculated standard errors will be calculated using the sandwich estimator by default when calling `summary`.
`Force`	logical. If set to `FALSE` (the default) the function stops if the response variable has more than twenty categories. Should be changed to `TRUE` if a model with more than twenty categories is desired.

Value

object of class "`oglmx`", see `oglmx`.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

polr for alternative method to estimate an ordered probit model. oglmx. To obtain marginal effects see margins.oglmx.

---

probit.reg — *Fit Probit Model.*

Description

Wrapper function for oglmx to estimate the binary response probit model.

Usage

```
probit.reg(formula, data, start = NULL, weights=NULL, beta = NULL,
          analhessian = TRUE, na.action, savemodelframe = FALSE, robust = FALSE)
```

Arguments

formula	an object of class formula: a symbolic description of the model used to explain the mean of the latent variable. The response variable should be a numeric vector or factor variable with two values.
data	a data frame containing the variables in the model.
start	either NULL or a numeric vector specifying start values for each of the estimated parameters, passed to the maximisation routine.
weights	either NULL or a numeric vector of length equal to the number of rows in the data frame. Used to apply weighted maximum likelihood estimation.
beta	NULL or numeric vector. Used to prespecify elements of the parameter vector for the equation of the mean of the latent variable. Vector should be of length one or of length equal to the number of explanatory variables in the mean equation. If of length one the value is presumed to correspond to the constant. If of length greater than one then NA should be entered for elements of the vector to be estimated.
analhessian	logical. Indicates whether the analytic Hessian should be calculated and used, default is TRUE, if set to FALSE a finite-difference approximation of the Hessian is used.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The factory-fresh default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.
savemodelframe	logical. Indicates whether the model frame(s) should be saved for future use. Default is FALSE. Should be switched to TRUE if intending to estimate Average Marginal Effects.
robust	logical. If set to TRUE the outer product or BHHH estimate of the meat in the sandwich of the variance-covariance matrix is calculated. If calculated standard errors will be calculated using the sandwich estimator by default when calling summary.

Value

object of class "oglmx", see oglmx.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

glm for alternative method to estimate a probit model. oglmx. To obtain marginal effects see margins.oglmx.

---

scoreMean *Calculate derivatives of loglikelihood*

Description

Functions used to calculate the first and second derivatives of the log-likelihood with respect to the estimated parameters.

Usage

```
scoreMean(eta_1,eta_0,std.dev,prob,link)

scoreVar(eta_1,eta_0,std.dev,gstd.dev,prob,link)

scoreThresh(estThresh,outcomematrix,eta_1,eta_0,std.dev,prob,link)

hessMean_Mean(eta_1,eta_0,std.dev,prob,link)

hessMean_Var(eta_1,eta_0,std.dev,gstd.dev,prob,link)

hessVar_Var(eta_1,eta_0,std.dev,gstd.dev,hstd.dev,prob,link)

hessMean_Thresh(estThresh,outcomematrix,eta_1,eta_0,std.dev,prob,link)

hessVar_Thresh(estThresh,outcomematrix,eta_1,eta_0,std.dev,gstd.dev,prob,link)

hessThresh_Thresh(estThresh,outcomematrix,eta_1,eta_0,std.dev,prob,link)
```

Arguments

eta_1	numeric vector or matrix. Refers to the input to the link function to calculate the probability at the right threshold of the outcome.
eta_0	numeric vector or matrix. Refers to the input to the link function to calculate the probability at the left threshold of the outcome.

std.dev	numeric vector or matrix. The standard deviation of the error term for the observations given the data and parameters.
prob	numeric vector or matrix. Probability of the outcome given the parameters and data.
link	character, indicates link function for the estimated model.
estThresh	numeric vector indicating which of the threshold values are estimated.
outcomematrix	numeric matrix indicating the outcome for each observation.
gstd.dev	numeric vector or matrix. The first derivative of standard deviation of the error term for the observations given the data and parameters.
hstd.dev	numeric vector or matrix. The second derivative of standard deviation of the error term for the observations given the data and parameters.

Value

numeric vector or matrix, depending on the structure of the inputs. Derivatives of the log-likelihood with respect to constants in the mean and variance equations and the threshold values.

Author(s)

Nathan Carroll, <nathan.carroll@ur.de>

See Also

oglmx

---

summary.oglmx *Summarizing Ordered Discrete Outcome Model Fits*

Description

summary method for class "oglmx"

Usage

```
## S3 method for class 'oglmx'
summary(object, tol = 1e-20, ... )

## S3 method for class 'summary.oglmx'
print(x, ... )
```

Arguments

object	an object of class "oglmx"
tol	argument passed to qr.solve, defines the tolerance for detecting linear dependencies in the hessian matrix to be inverted.
...	additional arguments, currently ignored.
x	object of class summary.oglmx.

Value

- `regtype` character string describing the type of model estimated.
- `loglikelihood` log-likelihood for the estimated model.
- `estimate` matrix with four columns and number of rows equal to the number of estimated parameters. Columns of the matrix correspond to estimated coefficients, standard errors, t-statistics and (two-sided) p-values.
- `estimateDisplay` the same data as in `estimate` but separated into a list with elements for each type of parameter estimate. The first element is for parameters in the mean equation, second element for parameters in the variance equation and the final element is for threshold parameters.
- `no.iterations` number of iterations used in function that maximises the log-likelihood.
- `McFaddensR2` McFadden's R^2 aka Pseudo-R^2. Calculated as:

$$R^2 = 1 - \log L_{fit} / \log L_0$$

  where $\log L_{fit}$ is the log-likelihood for the fitted model and $\log L_0$ is the log-likelihood from an intercept only model that estimates the probability of each alternative to be the sample average.
- `AIC` Akaike Information Criterion, calculated as:

$$AIC = 2k - 2 \log L_{fit}$$

  where k is the number of estimated parameters.
- `coefficients` named vector of estimated parameters.

Author(s)

Carroll, Nathan <nathan.carroll@ur.de>

References

McFadden, D. (1973) *Conditional Logit Analysis of Qualitative Choice Behavior* in *Frontiers in Econometrics*. P.Zarembka (Ed.), New York, Academic Press.

`vcov.oglmx` *Calculate Variance-Covariance Matrix for an oglmx Object*

Description

Returns the variance-covariance matrix of the estimated parameters of an oglmx object.

Usage

```
## S3 method for class 'oglmx'
vcov(object, tol = 1e-20, ... )
```

Arguments

`object`	an object of class "oglmx"
`tol`	argument passed to qr.solve, defines the tolerance for detecting linear dependencies in the hessian matrix to be inverted.
`...`	further arguments, currently ignored.

Value

A matrix of the estimated covariances between the parameter estimates obtained from inverting the Hessian at the returned parameter values in an oglmx object.

Author(s)

Nathan Carroll, `<nathan.carroll@ur.de>`

Index