Package: GLMcat (via r-universe)

November 4, 2024

Title Generalized Linear Models for Categorical Responses

Version 0.2.7

Description In statistical modeling, there is a wide variety of regression models for categorical dependent variables (nominal or ordinal data); yet, there is no software embracing all these models together in a uniform and generalized format. Following the methodology proposed by Peyhardi, Trottier, and Guédon (2015) <doi:10.1093/biomet/asv042>, we introduce 'GLMcat', an R package to estimate generalized linear models implemented under the unified specification (r, F, Z). Where r represents the ratio of probabilities (reference, cumulative, adjacent, or sequential), F the cumulative cdf function for the linkage, and Z, the design matrix.

License GPL-3

Encoding UTF-8

Depends R (>= 2.10)

LazyData true

RoxygenNote 7.2.3

LinkingTo Rcpp, BH, RcppEigen

Imports Rcpp, stats, stringr, ordinal

Suggests knitr, rmarkdown, testthat (>= 3.0.0), dplyr, ggplot2, gridExtra, gtools, tidyr

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/ylleonv/GLMcat

BugReports https://github.com/ylleonv/GLMcat/issues

Config/pak/sysreqs libicu-dev

Repository https://ylleonv.r-universe.dev

RemoteUrl https://github.com/ylleonv/glmcat

RemoteRef HEAD

RemoteSha c923a6137df3f1746be2c6f07e6cbf3919673bde

Contents

`accidents` *Accidents Dataset*

Description

This dataset contains information about various accidents, including details such as accident severity, road and weather conditions, light conditions, and the number of casualties.

Usage

```
accidents
```

Format

A data frame with 109,577 rows and 12 variables:

accident_severity Factor with levels `Slight`, `Serious`, `Fatal`

road_type Factor with levels `Dual carriageway`, `One way street`, `Roundabout`, `Single carriageway`, `Slip road`

weather_conditions Factor with levels `Fine + high winds`, `Fine no high winds`, `Fog or mist`, `Raining + high winds`, `Raining no high winds`, `Snowing`

light_conditions Factor with levels `Darkness`, `Daylight`

day_of_week Factor with levels Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

number_of_casualties Numeric, number of casualties in the accident

urban_or_rural_area Factor with levels Urban, Rural

speed_limit Numeric, speed limit at the accident location

junction_detail Factor with levels Not at junction or within 20 metres, T or staggered junction, Crossroads, Roundabout, Other junction, Private drive or entrance

carriageway_hazards Factor with levels Any animal in carriageway (except ridden horse), Data missing or out of range, None, Other object on road, Pedestrian in carriageway - not injured, Previous accident, Vehicle load on road

weather Factor with levels Fine + high winds, Fine no high winds, Fog or mist, Raining + high winds, Raining no high winds, Snowing

road Factor with levels Dual carriageway, One way street, Roundabout, Single carriageway, Slip road

Source

Data from 2019, openly available at https://www.data.gov.uk/, accessed in September 2023.

Examples

```
data(accidents)
```

---

anova.glmcat *Anova for a fitted* glmcat *model object*

Description

Compute an analysis of deviance table for one fitted glmcat model object.

Usage

```
## S3 method for class 'glmcat'
anova(object, ...)
```

Arguments

object	an object of class "glmcat".
...	additional arguments.

coef.glmcat *Model coefficients of a fitted* glmcat *model object*

Description

Returns the coefficient estimates of the fitted glmcat model object.

Usage

```
## S3 method for class 'glmcat'
coef(object, na.rm = FALSE, ...)
```

Arguments

object	an fitted object of class glmcat.
na.rm	TRUE for NA coefficients to be removed, default is FALSE.
...	additional arguments affecting the coef method.

confint.glmcat *Confidence intervals for parameters of a fitted* glmcat *model object*

Description

Computes confidence intervals from a fitted glmcat model object for all the parameters.

Usage

```
## S3 method for class 'glmcat'
confint(object, parm, level, ...)
```

Arguments

object	an fitted object of class glmcat.
parm	a numeric or character vector indicating which regression coefficients should be displayed
level	the confidence level.
...	other parameters.

control_glmcat *Control parameters for* glmcat *models*

Description

Set control parameters for glmcat models.

Usage

```
control_glmcat(maxit = 25, epsilon = 1e-06, beta_init = NA)
```

Arguments

maxit	the maximum number of the Fisher's Scoring Algorithm iterations. Defaults to 25.
epsilon	a double to change update the convergence criterion of GLMcat models.
beta_init	an appropriate sized vector for the initial iteration of the algorithm.

discrete_cm *Discrete Choice Models*

Description

Family of models for Discrete Choice. Fits discrete choice models which require data in long form. For each individual (or decision maker), there are multiple observations (rows), one for each of the alternatives the individual could have chosen. A group of observations of the same individual is a "case". It is important to note that each case represents a single statistical observation although it comprises multiple observations.

Usage

```
discrete_cm(
  formula,
  case_id,
  alternatives,
  reference,
  alternative_specific = NA,
  data,
  cdf = list(),
  intercept = "standard",
  normalization = 1,
  control = list(),
  na.action = "na.omit",
  find_nu = FALSE
)
```

Arguments

formula	a symbolic description of the model to be fit. An expression of the form y ~ predictors is interpreted as a specification that the response y is modeled by a linear predictor specified symbolically by model. A particularity for the formula is that for the case-specific variables, the user can define a specific effect for a category (in the parameter 'alternative_specific').
case_id	a string with the name of the column that identifies each case.
alternatives	a string with the name of the column that identifies the vector of alternatives the individual could have chosen.
reference	a string indicating the reference category.
alternative_specific	a character vector with the name of the explanatory variables that are different for each case, these are the alternative-specific variables. By default, the case-specific variables are the explanatory variables that are not identified here but are part of the formula.
data	a dataframe (in long format) object in R, with the dependent variable as a factor.
cdf	a parameter specifying the inverse distribution function to be used as part of the link function. If the distribution has no parameters to specify, it should be entered as a string indicating the name. The default value is 'logistic'. If there are parameters to specify, a list must be entered. For example, for Student's distribution, it would be 'list("student", df=2)'. For the non-central distribution of Student, it would be 'list("noncentralt", df=2, mu=1)'.
intercept	if set to "conditional", the design will be equivalent to the conditional logit model.
normalization	the quantile to use for the normalization of the estimated coefficients where the logistic distribution is used as the base cumulative distribution function.
control	a list specifying additional control parameters. - 'maxit': the maximum number of iterations for the Fisher scoring algorithm. - 'epsilon': a double value to fix the epsilon value. - 'beta_init': an appropriately sized vector for the initial iteration of the algorithm.
na.action	an argument to handle missing data. Available options are na.omit, na.fail, and na.exclude. It comes from the stats library and does not include the na.pass option.
find_nu	a logical argument to indicate whether the user intends to utilize the Student CDF and seeks an optimization algorithm to identify an optimal degrees of freedom setting for the model.

Details

Family of models for Discrete Choice

Note

For these models, it is not allowed to exclude the intercept.

Examples

```
library(GLMcat)
data(TravelChoice)

discrete_cm(formula = choice ~ hinc + gc + invt,
            case_id = "indv", alternatives = "mode", reference = "air",
            data = TravelChoice,
            cdf = "logistic")

#' Model with alternative specific effects for gc and invt:
discrete_cm(formula = choice ~ hinc + gc + invt,
            case_id = "indv", alternatives = "mode", reference = "air",
            data = TravelChoice, alternative_specific = c("gc", "invt"),
            cdf = "logistic")

 #' A more specific design was studied by Louvierte et al. (2000, p. 157) and Greene (2003, p. 730).
 #' These analyses set the effect of the variables hinc and psize exclusively for the category air
discrete_cm(formula = choice ~ hinc[air] + psize[air] + gc + ttme,
            case_id = "indv",
            alternatives = "mode",
            reference = "car",
            alternative_specific = c("gc", "ttme"),
            data = TravelChoice)
```

DisturbedDreams *Severity of disturbed dreams*

Description

Boy's disturbed dreams benchmark dataset drawn from a study that cross-classified boys by their age, and the severity (not severe, severe 1, severe 2, very severe) of their disturbed dreams (Maxwell, 1961).

Usage

```
data(DisturbedDreams)
```

Format

A dataframe containing :

Age Individuals age

Level Severity level: Not.severe, Severe.1, Severe.2, Very.severe.

References

Maxwell, A.E. (1961) *Analyzing qualitative data*, Methuen London, 73.

Examples

```
data(DisturbedDreams)
```

`extractAIC.glmcat` *Extract AIC from a fitted* `glmcat` *model object*

Description

Method to compute the (generalized) Akaike An Information Criterion for a fitted object of class `glmcat`.

Usage

```
## S3 method for class 'glmcat'
extractAIC(fit, ...)
```

Arguments

`fit`	an fitted object of class `glmcat`.
`...`	further arguments (currently unused in base R).

Examples

```
model <- glmcat(formula = Level ~ Age, data = DisturbedDreams,
                ref_category = "Very.severe", ratio = "cumulative")
extractAIC(model)
```

`glmcat` *Generalized linear models for categorical responses*

Description

Estimate generalized linear models implemented under the unified specification (ratio,cdf,Z) where `ratio` represents the ratio of probabilities (reference, cumulative, adjacent, or sequential), `cdf` the cumulative distribution function for the linkage, and Z the design matrix which must be specified through the `parallel` and the `threshold` arguments.

Usage

```
glmcat(
  formula,
  data,
  ratio = c("reference", "cumulative", "sequential", "adjacent"),
  cdf = list(),
  parallel = NA,
  categories_order = NA,
  ref_category = NA,
  threshold = c("standard", "symmetric", "equidistant"),
  control = list(),
  normalization = 1,
  na.action = "na.omit",
  find_nu = FALSE,
  ...
)
```

Arguments

formula	formula a symbolic description of the model to be fit. An expression of the form ‘y ~ predictors‘ is interpreted as a specification that the response ‘y‘ is modeled by a linear predictor specified by ‘predictors‘.
data	a dataframe object in R, with the dependent variable as a factor.
ratio	a string indicating the ratio (equivalently to the family) options are: reference, adjacent, cumulative and sequential. It is mandatory for the user to specify the desired ratio option as there is no default value.
cdf	The inverse distribution function to be used as part of the link function. - If the distribution has no parameters to specify, then it should be entered as a string indicating the name, e.g., ‘cdf = "normal"‘. The default value is ‘cdf = "logistic"‘. - If there are parameters to specify, then a list must be entered. For example, for Student’s distribution: ‘cdf = list("student", df=2)‘. For the non-central distribution of Student: ‘cdf = list("noncentralt", df=2, mu=1)‘.
parallel	a character vector indicating the name of the variables with a parallel effect. If a variable is categorical, specify the name and the level of the variable as a string, e.g., ‘"namelevel"‘.
categories_order	a character vector indicating the incremental order of the categories, e.g., ‘c("a", "b", "c")‘ for ‘a < b < c‘. Alphabetical order is assumed by default. Order is relevant for adjacent, cumulative, and sequential ratio.
ref_category	a string indicating the reference category. This option is suitable for models with reference ratio.
threshold	a restriction to impose on the thresholds. Options are: ‘standard‘, ‘equidistant‘, or ‘symmetric‘. This is valid only for the cumulative ratio.
control	a list of control parameters for the estimation algorithm. - ‘maxit‘: The maximum number of iterations for the Fisher scoring algorithm. - ‘epsilon‘: A double to change the convergence criterion of GLMcat models. - ‘beta_init‘: An appropriately sized vector for the initial iteration of the algorithm.

normalization	the quantile to use for the normalization of the estimated coefficients when the logistic distribution is used as the base cumulative distribution function.
na.action	an argument to handle missing data. Available options are 'na.omit', 'na.fail', and 'na.exclude'. It does not include the 'na.pass' option.
find_nu	a logical argument to indicate whether the user intends to utilize the Student CDF and seeks an optimization algorithm to identify an optimal degrees of freedom setting for the model.
...	additional arguments. Note: If the 'reference' ratio is used, you'll get a warning if the variable is an ordered factor. Note: If any other 'radio' is used, it will issue a warning if the response is not ordered, and the variables order will default to the alphanumeric natural order.

Details

Fitting models for categorical responses

This function fits generalized linear models for categorical responses using the unified specification framework introduced by Peyhardi, Trottier, and Guédon (2015).

References

Peyhardi J, Trottier C, Guédon Y (2015). "A new specification of generalized linear models for categorical responses." *Biometrika*, 102(4), 889–906. doi:10.1093/biomet/asv042.

See Also

summary.glmcat

Examples

```
data(DisturbedDreams)
ref_log_com <- glmcat(formula = Level ~ Age, data = DisturbedDreams,
    ref_category = "Very.severe",
    cdf = "logistic", ratio = "reference")
```

---

logLik.glmcat *Log-likelihood of a fitted* glmcat *model object*

Description

Extract Log-likelihood of a fitted glmcat model object.

Usage

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

Arguments

object	an fitted object of class glmcat.
...	additional arguments affecting the loglik.

---

nobs.glmcat *Number of observations of a fitted* glmcat *model object*

Description

Extract the number of observations of the fitted glmcat model object.

Usage

```
## S3 method for class 'glmcat'
nobs(object, ...)
```

Arguments

object	an fitted object of class glmcat.
...	additional arguments affecting the nobs method.

---

plot.glmcat *Plot method for a fitted* glmcat *model object*

Description

plot of the log-likelihood profile for a fitted glmcat model object.

Usage

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

Arguments

x	an object of class glmcat.
...	additional arguments.

predict.glmcat *Predict method for a a fitted* glmcat *model object*

Description

Obtains predictions of a fitted glmcat model object.

Usage

```
## S3 method for class 'glmcat'
predict(object, newdata, type, ...)
```

Arguments

object	a fitted object of class glmcat.
newdata	optionally, a data frame in which to look for the variables involved in the model. If omitted, the fitted linear predictors are used.
type	the type of prediction required. The default is "prob" which gives the probabilities, the other option is "linear.predictor" which gives predictions on the scale of the linear predictor.
...	further arguments. The default is "prob" which gives the probabilities, the other option is "linear.predictor" which gives predictions on the scale of the linear predictor.

print.anova.glmcat *Printing Anova for* glmcat *model fits*

Description

print.anova method for GLMcat objects.

Usage

```
## S3 method for class 'anova.glmcat'
print(x, digits = max(getOption("digits") - 2, 3), ...)
```

Arguments

x	an object of class "glmcat".
digits	the number of digits in the printed table.
...	additional arguments affecting the summary produced.

print.glmcat *Print method for a fitted* glmcat *model object*

Description

print method for a fitted glmcat model object.

Usage

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

Arguments

x	an object of class glmcat.
...	additional arguments.

Examples

```
model <- glmcat(formula = Level ~ Age, data = DisturbedDreams,
                ref_category = "Very.severe", ratio = "cumulative")
print(model)
```

print.summary.glmcat *Printing a fitted* glmcat *model object*

Description

print.summary method for GLMcat objects.

Usage

```
## S3 method for class 'summary.glmcat'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	an object of class "glmcat".
digits	the number of digits in the printed table.
...	additional arguments affecting the summary produced.

step.glmcat *Stepwise for a* glmcat *model object*

Description

Stepwise for a glmcat model object based on the AIC.

Usage

```
## S3 method for class 'glmcat'
step(object, scope, scale, direction, trace, keep, steps, k, ...)
```

Arguments

object	an fitted object of class glmcat.
scope	defines the range of models examined in the stepwise search (same as in the step function of the stats package). This should be either a single formula, or a list containing components upper and lower, both formulae.
scale	the scaling parameter (if applicable).
direction	the mode of the stepwise search.
trace	to print the process information.
keep	a logical value indicating whether to keep the models from all steps.
steps	the maximum number of steps.
k	additional arguments (if needed).
...	additional arguments passed to the function.

summary.glmcat *Summary method for a fitted* glmcat *model object*

Description

Summary method for a fitted ‘glmcat‘ model object.

Usage

```
## S3 method for class 'glmcat'
summary(object, normalized = FALSE, correlation = FALSE, ...)
```

Arguments

object	an fitted object of class ‘glmcat‘.
normalized	if ‘TRUE‘, the summary method yields the normalized coefficients.
correlation	if ‘TRUE‘, prints the correlation matrix.
...	additional arguments affecting the summary produced.

Examples

```
mod1 <- discrete_cm(formula = choice ~ hinc + gc + invt,
                    case_id = "indv", alternatives = "mode", reference = "air",
                    data = TravelChoice,  alternative_specific = c("gc", "invt"),
                    cdf = "normal", normalization = 0.8)
summary(mod1, normalized = TRUE)
```

---

terms.glmcat *Terms of a fitted* glmcat *model object*

Description

Returns the terms of a fitted glmcat model object.

Usage

```
## S3 method for class 'glmcat'
terms(x, ...)
```

Arguments

x	an object of class glmcat.
...	additional arguments.

TravelChoice *Travel Mode Choice*

Description

The data set contains 210 observations on mode choice for travel between Sydney and Melbourne, Australia.

Usage

```
data(TravelChoice)
```

Format

A dataframe containing :

indv Id of the individual

mode available options: air, train, bus or car

choice a logical vector indicating as TRUE the transportation mode chosen by the traveler As category-specific variables:

invt travel time in vehicle

gc generalized cost measure

ttme terminal waiting time for plane, train and bus; 0 for car

invc in vehicle cost As case-specific variables:

hinc household income

psize traveling group size in mode chosen

Source

Download from on-line (18/09/2020) complements to Greene, W.H. (2011) Econometric Analysis, Prentice Hall, 7th Edition, Table F18-2.

References

Greene, W.H. and D. Hensher (1997) *Multinomial logit and discrete choice models in* Greene, W. H. (1997) *LIMDEP version 7.0 user's manual revised*, Plainview, New York econometric software, Inc .

Examples

```
data(TravelChoice)
```

---

vcov.glmcat *Variance-Covariance Matrix for a fitted* glmcat *model object*

Description

Returns the variance-covariance matrix of the main parameters of a fitted glmcat model object.

Usage

```
## S3 method for class 'glmcat'
vcov(object,...)
```

Arguments

object	an object of class glmcat.
...	additional arguments.

Index