Package ‘Rchoice’

May 27, 2020

Title Discrete Choice (Binary, Poisson and Ordered) Models with Random Parameters

Version 0.3-2

Author Mauricio Sarrias <msarrias86@gmail.com>

Maintainer Mauricio Sarrias <msarrias86@gmail.com>

Description An implementation of simulated maximum likelihood method for the estimation of Binary (Probit and Logit), Ordered (Probit and Logit) and Poisson models with random parameters for cross-sectional and longitudinal data.

Depends R (>= 3.1.0), Formula, maxLik

Imports msm, plm, plotrix, stats, graphics

Suggests car, lmtest, memisc, pglm, sandwich

License GPL (>= 2)

URL http://msarrias.com/rchoice-package-in-r.html

LazyData no

RoxygenNote 7.1.0

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-27 15:30:07 UTC

R topics documented:

AIC.Rchoice *Akaike's Information Criterion*

Description

Calculate Akaike's information Criterion (AIC) or the Bayesian information Criterion (BIC) for a model of class Rchoice.

Usage

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

## S3 method for class 'Rchoice'
BIC(object, ...)
```

Arguments

object	a fitted model of class Rchoice,
...	additional arguments to be passed to or from other functions,
k	a numeric value, use as penalty coefficient for number of parameters in the fitted model,

Value

a numeric value with the corresponding AIC or BIC value.

See Also

Rchoice

Examples

```
## Probit model
data("Workmroz")
probit <- Rchoice(lfp ~ k5 + k618 + age + wc + hc + lwg + inc,
                  data = Workmroz , family = binomial('probit'))
summary(probit)

AIC(probit)
BIC(probit)
```

`Articles` *Doctoral Publications*

Description

Data from research by Long(1990) that analizes the scientist's level of publications.

Usage

```
data(Articles)
```

Format

A data frame with 915 observations on the following 6 variables.

`art` Articles during last 3 years of Ph.D.

`fem` 1 if female scientist; else 0

`mar` 1 if married; else 0

`kid5` Number of children 5 or younger

`phd` Prestige of Ph.D. department

`ment` Articles by mentor during last 3 years

Source

- Long, J. S. (1990). The origins of sex differences in science. Social Forces, 68(4), 1297-1316.
- Long, J. S. (1997). Regression models for categorical and limited dependent variables (Vol. 7). Sage.
- Long, J. S., & Freese, J. (2006). Regression models for categorical and limited dependent variables using Stata. Stata Press, College Station, TX.

Examples

```
data(Articles)
```

`Attitudes` *Attituded toward working mothers*

Description

In 1997 and 1989, the General Social Survey asked respondents to evaluate the following statement: "A working mother can establish just as warm and secure a relationship with her children as a mother who does not work".

Usage

```
data(Attitudes)
```

Format

A data frame with 2293 observations on the following 10 variables.

`warm` 1 = Strongly disagree, 2 = disagree, 3 = agree, 4 = strongly agree

`yr89` survey year: 1 = 1989; 0 = 1977

`male` 1 = male; 0 = female

`white` 1 = white; 0 = nonwhite

`age` age in years

`ed` years of education

`prst` occupational prestige

Source

- Clogg, C. C., & Shihadeh, E. S. (1994). Statistical models for ordinal variables. Thousand Oaks, CA: Sage Publications.
- Long, J. S. (1997). Regression models for categorical and limited dependent variables (Vol. 7). Sage.
- Long, J. S., & Freese, J. (2006). Regression models for categorical and limited dependent variables using Stata. Stata Press, College Station, TX.

Examples

```
data(Attitudes)
```

bread.Rchoice *Bread for sandwiches*

Description

Computes the “bread” of the sandwich covariance matrix for a model of class `Rchoice`

Usage

```
## S3 method for class 'Rchoice'
bread(x, ...)
```

Arguments

`x`	a fitted model of class `Rchoice`,
`...`	Other arguments when `bread` is applied to another class object.

Details

For more information see `bread` from the package **sandwich**.

Value

the covariance matrix times observations

References

Zeileis A (2006), Object-oriented Computation of Sandwich Estimators. Journal of Statistical Software, 16(9), 1–16.

Examples

```
## Probit model
data("Workmroz")
probit <- Rchoice(lfp ~ k5 + k618 + age + wc + hc + lwg + inc,
                  data = Workmroz , family = binomial('probit'))
summary(probit)

library("sandwich")
bread(probit)
```

effect.Rchoice *Get the conditional individual coefficients*

Description

This a helper function to obtain the individuals' conditional estimate of the random parameters or compensating variations.

Usage

```
effect.Rchoice(x, par = NULL, effect = c("cv", "ce"), wrt = NULL, ...)
```

Arguments

`x`	a object of class `Rchoice`,
`par`	a string giving the name of the variable with random parameter,
`effect`	a string indicating what should be computed: the conditional expectation of the individual coefficients "`ce`", or the conditional expectation of the individual compensating variations "`cv`",
`wrt`	a string indicating repect to which variable the compensating variation should be computed,
`...`	further arguments. Ignored.

Value

A named list where "mean" contains the individuals' conditional mean for the random parameter or compensating variation, and where 'sd.est' contains their standard errors.

References

- Greene, W. H. (2012). Econometric Analysis, Seventh Edition. Pearson Hall.
- Train, K. (2009). Discrete Choice Methods with Simulation. Cambridge university press.

See Also

`Rchoice` for the estimation of different discrete choice models with individual parameters.

Examples

```
## Not run:
## Probit Model with Random Effects and Random Parameters
data('Unions', package = 'pglm')
Unions$lwage <- log(Unions$wage)
union.ran <- Rchoice(union ~ age + exper + rural + lwage,
                     data = Unions[1:2000, ],
                     family = binomial('probit'),
                     ranp = c(constant = "n", lwage = "t"),
                     R = 10,
                     panel = TRUE,
                     index = "id",
                     print.init = TRUE)

## Get the individuals' conditional mean and their standard errors for lwage
bi.wage <- effect.Rchoice(union.ran, par = "lwage", effect = "ce")
summary(bi.wage$mean)
summary(bi.wage$sd.est)

## End(Not run)
```

estfun.Rchoice *Gradient for observations*

Description

It extracts the gradient for each observations evaluated at the estimated parameters for a model of class Rchoice

Usage

```
## S3 method for class 'Rchoice'
estfun(x, ...)
```

Arguments

x	a fitted model of class Rchoice,
...	Other arguments when estfun is applied to another class object

Details

For more information see estfun from package **sandwich**.

Value

the gradient matrix of dimension n times k

References

Zeileis A (2006), Object-oriented Computation of Sandwich Estimators. Journal of Statistical Software, 16(9), 1–16.

`getSummary.Rchoice` *Get Model Summaries for Use with "mtable"*

Description

A generic function to collect coefficients and summary statistics from a `Rchoice` object. It is used in `mtable`

Usage

```
getSummary.Rchoice(obj, alpha = 0.05, ...)
```

Arguments

`obj`	a `Rchoice` object,
`alpha`	level of the confidence intervals,
`...`	further arguments,

Details

For more details see package **memisc**.

`Health` *German Health Care Data*

Description

German Health Care Data, unbalanced panel.

Usage

```
data(Health)
```

Format

A data frame with 27326 observations on the following 27 variables.

`id` person identification number

`female` female =1, male =0

`year` calendar year of the observation

`age` age in years

`hsat` health satisfaction, 0 (low),...,10 (high)

`handdum` handicapped = 1, 0 otherwise

`handper` degree of handicap in percent; 0,100 `hhinc` household nomial monthly net income in German marks

`hhkids` children under age 16 in the household = 1; otherwise = 0

`educ` years of schooling

`married` married =1, otherwise = 0

`haupts` highest schooling degree is Hauptschul degree = 1; otherwise = 0

`reals` highest schooling degree is Realschul degree = 1, otherwise = 0

`fachhs` highest schooling degree is Polytechical degree = 1; otherwise = 0

`abitur` highest schooling degree is Abitur = 1; otherwise = 0

`univ` highest schooling degree is university degree =1; otherwise = 0

`working` employed =1; otherwise = 0

`bluec` blue-collar employee = 1; otherwise = 0

`whitec` white-collar employeee =1; otherwise = 0

`self` self-employed = 1; otherwise = 0

`beamt` civil servant = 1; otherwise = 0

`docvis` number of doctor visits in last three months

`hospvis` number of hospital visits in last calendar year

`public` insured in public health =1; otherwise = 0

`addon` insured by add-on insurance =1; otherwise = 0

`hsat2` 40 observations on hsat recorded between 6 and 7 were changed to 7

`newhsat` recording of hsat, (0-2) = 0, (3-5)=1, (6-8)=2, (9)=3 (10)=4

Source

Riphahn, R. T., Wambach, A., & Million, A. (2003). Incentive effects in the demand for health care: a bivariate panel count data estimation. Journal of applied econometrics, 18(4), 387-405.

References

Greene, W. H. (2003). Econometric analysis. Pearson Education India.

Examples

```
data(Health)
```

plot.Rchoice *Plot of the distribution of conditional expectation of random parameters.*

Description

Plot the distribution of the conditional expectation of the random parameters or compensating variations for objects of class `Rchoice`.

Usage

```
## S3 method for class 'Rchoice'
plot(
  x,
  par = NULL,
  effect = c("ce", "cv"),
  wrt = NULL,
  type = c("density", "histogram"),
  adjust = 1,
  main = NULL,
  col = "indianred1",
  breaks = 10,
  ylab = NULL,
  xlab = NULL,
  ind = FALSE,
  id = NULL,
  ...
)
```

Arguments

`x` a object of class `Rchoice`,

`par` a string giving the name of the variable with random parameter,

`effect` a string indicating what should be plotted: the conditional expectation of the individual coefficients `"ce"`, or the conditional expectation of the individual compensating variations `"cv"`,

`wrt` a string indicating repect to which variable should be computed the compensating variation,

`type` a string indicating the type of distribution: it can be a `histogram` or a `density` of the conditional expectation,

`adjust` bandwidth for the kernel density,

`main` an overall title for the plot,

`col` color for the graph,

`breaks` number of breaks for the histrogram if `type = "histogram"`, `ylab` a title for the y axis,

`xlab` a title for the x axis,

`ind` a boolean. If `TRUE`, a 95 As default, the conditional expectation of `par` for the first 10 individual is plotted,

`id` only relevant if `ind` is not `NULL`. This is a vector indicating the individuals for which the confidence intervals are plotted,

`...` further arguments. Ignored.

Author(s)

Mauricio Sarrias

References

- Greene, W. H. (2012). Econometric analysis, Seventh Edition. Pearson Hall.
- Train, K. (2009). Discrete choice methods with simulation. Cambridge university press.

See Also

`Rchoice` for the estimation of different discrete choice models with individual parameters.

Examples

```
## Not run:
## Probit Model with Random Effects and Random Parameters
data('Unions', package = 'pglm')
Unions$lwage <- log(Unions$wage)
union.ran <- Rchoice(union ~ age + exper + rural + lwage,
                     data = Unions[1:2000, ],
                     family = binomial('probit'),
                     ranp = c(constant = "n", lwage = "t"),
                     R = 10,
                     panel = TRUE,
                     index = "id",
                     print.init = TRUE)

## Plot the distribution of the conditional mean for lwage
plot(union.ran, par = "lwage", type = "density")

## Plot the conditional mean for the first 20 individuals
plot(union.ran, par = "lwage", ind =  TRUE, id = 1:20, col = "blue")

## Plot the compensating variation
plot(union.ran, par = "lwage", effect = "cv", wrt = "rural", type = "histogram")

## End(Not run)
```

Rchoice *Estimate discrete choice model with random parameters*

Description

Estimation of discrete choice models such as Binary (logit and probit), Poisson and Ordered (logit and probit) model with random coefficients for cross-sectional and panel data using simulated maximum likelihood.

Usage

```
Rchoice(
  formula,
  data,
  subset,
  weights,
  na.action,
  family,
  start = NULL,
  ranp = NULL,
  R = 40,
  haltons = NA,
  seed = 10,
  correlation = FALSE,
  panel = FALSE,
  index = NULL,
  mvar = NULL,
  print.init = FALSE,
  init.ran = 0.1,
  gradient = TRUE,
  ...
)

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

## S3 method for class 'Rchoice'
model.matrix(object, ...)

## S3 method for class 'Rchoice'
coef(object, ...)

## S3 method for class 'Rchoice'
nObs(x, ...)

## S3 method for class 'Rchoice'
fitted(object, ...)
```

```
## S3 method for class 'Rchoice'
residuals(object, ...)

## S3 method for class 'Rchoice'
df.residual(object, ...)

## S3 method for class 'Rchoice'
update(object, new, ...)

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

## S3 method for class 'Rchoice'
print(
  x,
  digits = max(3, getOption("digits") - 3),
  width = getOption("width"),
  ...
)

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

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

Arguments

`formula` a symbolic description of the model to be estimated. The `formula` consists in two parts. The first one is reserved for standard variables with fixed and random parameters. The second one is reserved for variables that enter in the mean of the random parameters. See for example `rFormula`,

`data` the data. It may be a `pdata.frame` object or an ordinary `data.frame`,

`subset` an optional vector specifying a subset of observations,

`weights` an optional vector of weigths,

`na.action` a function wich indicated what should happen when the data contains `NA`'s,

`family` the distribution to be used. It might be `family = binomial("probit")` for a Probit Model, `family = binomial("logit")` for a Logit model, `family = ordinal("probit")` for an Ordered Probit Model, `family = ordinal("logit")` for a Ordered Logit Model for an Ordered Logit Model, and `family = "poisson"` for a Poisson Model, `start` a vector of starting values,

`ranp` a named vector whose names are the random parameters and values the distribution: `"n"` for normal, `"ln"` for log-normal, `"cn"` for truncated normal, `"u"` for uniform, `"t"` for triangular, `"sb"` for Johnson Sb,

`R` the number of draws if `ranp` is not `NULL`,

`haltons` only relevant if `ranp` is not `NULL`. If not `NULL`, halton sequence is used instead of pseudo-random numbers. If `haltons=NA`, some default values are used for the prime of the sequence and for the number of element dropped. Otherwise, `haltons` should be a list with elements `prime` and `drop`,

`seed` the seed for the pseudo-random draws. This is only relevant if `haltons = NULL`,

`correlation` only relevant if `ranp` is not `NULL`. If `TRUE`, the correlation between random parameters is taken into account,

`panel` if `TRUE` a panel data model is estimated,

`index` a string indicating the ‘id’ for individuals in the data. This argument is not required if data is a `pdata.frame` object,

`mvar` only valid if `ranp` is not `NULL`. This is a named list, where the names correspond to the variables with random parameters, and the values correspond to the variables that enter in the mean of each random parameters,

`print.init` if `TRUE`, the initial values for the optimization procedure are printed,

`init.ran` initial values for standard deviation of random parameters. Default is 0.1,

`gradient` if `FALSE`, numerical gradients are used for the optimization procedure of models with random parameters,

`...` further arguments passed to `maxLik`,

`x, object` and object of class `Rchoice`,

`new` an updated formula for the update method,

`digits` number of digits,

`width` width,

Details

The models are estimated using the `maxLik` function from `maxLik` package.

If `ranp` is not `NULL`, the random parameter model is estimated. A random parameter model or random coefficient models permits regression parameter to vary across individuals according to some distribution. A fully parametric random parameter model specifies the latent variable y^* conditional on regressors x and given parameters β_i to have conditional density $f(y|x, \beta_i)$ where β_i are iid with density $g(\beta_i|\theta_i)$. The density is assumed a priori by the user by the argument `ranp`. If the parameters are assumed to be normally distributed β_i $N(\beta, \Sigma)$, then the random parameter are constructed as:

$$\beta_{ir} = \beta + L\omega_{ir}$$

where $LL' = \Sigma$ and ω_{ir} is the r-th draw from standard normal distribution for individual i.

Once the model is specified by the argument `family`, the model is estimated using Simulated Maximum Likelihood (SML). The probabilities, given by $f(y|x, \beta_i)$, are simulated using `R` pseudo-draws if `halton=NULL` or `R` halton draws if `halton = NA`. The user can also specified the primes and the number of dropped elements for the halton draws. For example, if the model consists of two random parameters, the user can specify `haltons = list("prime" = c(2,3),"drop" = c(11,11))`.

A random parameter hierarchical model can be estimated by including heterogeneity in the mean of the random parameters:

$$\beta_{ir} = \beta + \pi' s_i + L\omega_{ir}$$

Rchoice manages the variables in the hierarchical model by the `formula` object: all the hierarchical variables (s_i) are included after the `|` symbol. The argument `mvar` indicate which variables enter in each random parameter. See examples below

Value

An object of class “`Rchoice`”, a list elements:

`coefficients`	the named vector of coefficients,
`family`	type of model,
`link`	distribution of the errors,
`logLik`	a set of values of the maximum likelihood procedure,
`mf`	the model framed used,
`formula`	the formula (a Formula object),
`time`	`proc.time()` minus the start time,
`freq`	frequency of dependent variable,
`draws`	type of draws used,
`R.model`	`TRUE` if a random parameter model is fitted,
`R`	number of draws used,
`bi`	an array of dimension $N \times R \times K$ with the individual parameters,
`Qir`	matrix of dimension $N \times R$ representing $P_{ir}/\sum_r P_{ir}$,
`ranp`	vector indicating the variables with random parameters and their distribution,
`probabilities`	the fitted probabilities for each individuals,
`residuals`	the residuals,
`call`	the matched call.

Author(s)

Mauricio Sarrias <`msarrias86@gmail.com`>

References

Greene, W. H. (2012). Econometric Analysis. 7 edition. Prentice Hall.

Train, K. (2009). Discrete Choice Methods with Simulation. Cambridge university press.

See Also

`plot.Rchoice`, `effect.Rchoice`

Examples

```
## Probit model
data("Workmroz")
probit <- Rchoice(lfp ~ k5 + k618 + age + wc + hc + lwg + inc,
                  data = Workmroz, family = binomial('probit'))
summary(probit)

## Poisson model
data("Articles")
poisson <- Rchoice(art ~ fem + mar + kid5 + phd + ment, data = Articles, family = poisson)
summary(poisson)

## Ordered probit model
data("Health")
oprobit <- Rchoice(newhsat ~ age + educ + hhinc + married + hhkids,
data = Health, family = ordinal('probit'), subset = year == 1988)
summary(oprobit)

## Poisson Model with Random Parameters
## Not run:
poisson.ran <- Rchoice(art ~ fem + mar + kid5 + phd + ment,
                       data = Articles,  family = poisson,
                       ranp = c(kid5 = "n", phd = "n", ment = "n"))
summary(poisson.ran)

## Poisson Model with Correlated Random Parameters
poissonc.ran <- Rchoice(art ~ fem + mar + kid5 + phd + ment,
                       data = Articles,
                       ranp = c(kid5 = "n", phd = "n", ment = "n"),
                       family = poisson,
                       correlation =  TRUE)
summary(poissonc.ran)

## Hierarchical Poisson Model
poissonH.ran <- Rchoice(art ~ fem + mar + kid5 + phd + ment | fem + phd,
                       data = Articles,
                       ranp = c(kid5 = "n", phd = "n", ment = "n"),
                       mvar = list(phd = c("fem"), ment = c("fem", "phd")),
                       family = poisson,
                       R = 10)
summary(poissonH.ran)

## Probit Model with Random Effects and Random Parameters
data('Unions', package = 'pglm')
Unions$lwage <- log(Unions$wage)
union.ran <- Rchoice(union ~ age + exper + rural + lwage,
                     data = Unions[1:2000, ],
                     family = binomial('probit'),
                     ranp = c(constant = "n", lwage = "t"),
                     R = 10,
                     panel = TRUE,
                     index = "id",
                     print.init = TRUE)
summary(union.ran)

## Ordered Probit Model with Random Effects and Random Parameters
oprobit.ran <- Rchoice(newhsat ~ age + educ + married + hhkids + linc,
                     data = Health[1:2000, ],
                     family = ordinal('probit'),
                     ranp = c(constant = "n", hhkids = "n", linc = "n"),
                     panel = TRUE,
                     index = "id",
                     R = 100,
                     print.init = TRUE)
summary(oprobit.ran)

## End(Not run)
```

---

rFormula *Model formula for Rchoice models*

Description

Two kind of variables are used in models with individual heterogenetiy: the typical variables that enter in the latent process and those variables that enter in the random parameter (Hierarchical Model). `rFormula` deal with this type of models using suitable methods to extract the elements of the model.

Usage

```
rFormula(object)

is.rFormula(object)

## S3 method for class 'rFormula'
model.frame(formula, data, ..., lhs = NULL, rhs = NULL)

## S3 method for class 'rFormula'
model.matrix(object, data, rhs = NULL, ...)
```

Arguments

`object`	a formula form the `rFormula` function, for the `model.matrix` method, a `rFormula` object,
`formula`	a `rFormula` object,
`data`	a `data.frame`,
`...`	further arguments.

lhs	see Formula,
rhs	see Formula,

vcov.Rchoice *vcov method for Rchoice objects*

Description

The vcov method for Rchoice objects extracts the covariance matrix of the coefficients or the random parameters. It also allows to get the standard errors for the variance-covariance matrix of the random parameters

Usage

```
## S3 method for class 'Rchoice'
vcov(
  object,
  what = c("coefficient", "ranp"),
  type = c("cov", "cor", "sd"),
  se = FALSE,
  digits = max(3, getOption("digits") - 2),
  ...
)

cov.Rchoice(x)

cor.Rchoice(x)

se.cov.Rchoice(x, sd = FALSE, digits = max(3, getOption("digits") - 2))
```

Arguments

object
: a fitted model of class Rchoice,

what
: indicates which covariance matrix has to be extracted. The default is coefficient, in this case the vcov behaves as usual. If what = "ranp" the covariance matrix of the random parameters is returned as default,

type
: if the model is estimated with random parameters, then this argument indicates what matrix should be returned. If type = "cov", then the covariance matrix of the random parameters is returned; if type = "cor" then the correlation matrix of the random parameters is returned; if type = "sd" then the standard deviation of the random parameters is returned,

se
: if TRUE type = "cov" then the standard error of the covariance matrix of the random parameters is returned; if TRUE type = "sd" the standard error of the standard deviation of the random parameter is returned. This argument if valid only if the model is estimated using correlated random parameters,

digits
: number of digits,

`...` further arguments

`x` a fitted model of class `Rchoice`,

`sd` if `TRUE`, then the standard deviation of the random parameters are returned,

Details

This new interface replaces the `cor.Rchoice`, `cov.Rchoice` and `se.cov.Rchoice` functions which are deprecated.

See Also

`Rchoice` for the estimation of discrete choice models with random parameters.

`Workmroz` *Labor Force Participation*

Description

Data extracted by Mroz(1987) from the 197 Panel Study of Income Dynacmis. The sample consists of 753 white, married women between the ages of 30 and 60.

Usage

```
data(Workmroz)
```

Format

A data frame with 753 observations on the following 9 variables.

`lfp` 1 if wife is in the paid labor force; else 0

`k5` Number of children ages 5 and younger

`k618` Number of children ages 6 to 18

`age` Wife's age in years

`wc` 1 if wife attended college; else 0

`hc` 1 if husband attended college; else 0

`lwg` Log of wife's estimated wage rate

`inc` Family income excluding wife's wage

`linc` Log of Family income excluding wife's wage

Source

Mroz, T. A. (1987). The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions. Econometrica, 55(4), 765-799

Examples

```
data(Workmroz)
```

Index