# Package 'gmnl'

July 8, 2026

**Type** Package

**Title** Multinomial Logit Models with Random Parameters

**Version** 1.1-4

**Date** 2026-07-08

**Description** An implementation of maximum simulated likelihood method for the estimation of multinomial logit models with random coefficients as presented by Sarrias and Daziano (2017) <doi:10.18637/jss.v079.i02>. Specifically, it allows estimating models with continuous heterogeneity such as the mixed multinomial logit and the generalized multinomial logit. It also allows estimating models with discrete heterogeneity such as the latent class and the mixed-mixed multinomial logit model.

**Depends** R (>= 3.6.0), maxLik, Formula

**Imports** plotrix, msm, mlogit, truncnorm, stats, graphics, utils

**Suggests** AER, lmtest, car, memisc, testthat

**URL** https://www.msarrias.com/description.html

**License** GPL (>= 2)

**RoxygenNote** 7.3.3

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Mauricio Sarrias [aut, cre] (ORCID: <https://orcid.org/0000-0001-5932-4817>), Ricardo Daziano [aut], Yves Croissant [ctb]

**Maintainer** Mauricio Sarrias <msarrias86@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-07-08 15:00:02 UTC

# Contents

AIC.gmnl . . . . . 2
bread.gmnl . . . . . 3
cov.gmnl . . . . . 4
effect.gmnl . . . . . 5
estfun.gmnl . . . . . 7
getSummary.gmnl . . . . . 7
gFormula . . . . . 8
gmnl . . . . . 9
plot.gmnl . . . . . 15
vcov.gmnl . . . . . 17
wtp.gmnl . . . . . 18

**Index** **20**

---

`AIC.gmnl` *Akaike's Information Criterion*

---

## Description

Calculate the Akaike's information Criterion (AIC) or the Bayesian information Criterion (BIC) for an object of class `gmnl`.

## Usage

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

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

## Arguments

| | |
|---|---|
| `object` | a fitted model of class `gmnl`. |
| `...` | 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. |

## Details

For more information see `AIC` or `BIC`

## Value

A numeric value with the corresponding AIC or BIC value.

## See Also

gmnl for the estimation of multinomial logit models with observed and unobserved individual heterogeneity.

## Examples

```
## Estimate MNL model
data("TravelMode", package = "AER")
library(mlogit)
TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
                  alt.levels = c("air", "train", "bus", "car"), chid.var = "individual")

mnl <- gmnl(choice ~ wait + vcost + travel + gcost | 0 , data = TM)
AIC(mnl)
BIC(mnl)
```

---

# bread.gmnl *Bread for Sandwiches*

---

## Description

Computes the "bread" of the sandwich covariance matrix for objects of class gmnl.

## Usage

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

## Arguments

| | |
|---|---|
| x | a fitted model of class gmnl. |
| ... | 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.

cov.gmnl *Functions for Correlated Random Parameters*

## Description

These are a set of functions that help to extract the variance-covariance matrix, the correlation matrix, and the standard error of the random parameters for models of class gmnl.

## Usage

```
cov.gmnl(x, Q = NULL)

cor.gmnl(x, Q = NULL)

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

## Arguments

| | |
|---|---|
| x | an object of class gmnl where ranp is not NULL. |
| Q | this argument is only valid if the "mm" (MM-MNL) model is estimated. It indicates the class for which the variance-covariance matrix is computed. |
| sd | if TRUE, then the standard deviations of the random parameters along with their standard errors are computed. |
| digits | the number of digits. |

## Details

The variance-covariance matrix is computed using the Cholesky decomposition $LL' = \Sigma$.

se.cov.gmnl function is a wrapper for the deltamethod function of the **msm** package.

## Value

cov.gmnl returns a matrix with the variance of the random parameters if the model is fitted with random coefficients. If the model is fitted with correlation = TRUE, then the variance-covariance matrix is returned.

If correlation = TRUE in the fitted model, then se.cov.gmnl returns a coefficient matrix for the elements of the variance-covariance matrix or the standard deviations if sd = TRUE.

## Author(s)

Mauricio Sarrias <msarrias86@gmail.com>

## References


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

## See Also

gmnl for the estimation of different multinomial models with individual heterogeneity.

## Examples

```
## Not run:
## Examples using Electricity data set from mlogit package
library(mlogit)
data("Electricity", package = "mlogit")
Electr <- mlogit.data(Electricity, id.var = "id", choice = "choice",
                      varying = 3:26, shape = "wide", sep = "")

## Estimate a MIXL model with correlated random parameters
Elec.cor <- gmnl(choice ~ pf + cl + loc + wk + tod + seas| 0, data = Electr,
                 subset = 1:3000,
                 model = 'mixl',
                 R = 10,
                 panel = TRUE,
                 ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n"),
                 correlation = TRUE)

## Use functions for correlated random parameters
cov.gmnl(Elec.cor)
se.cov.gmnl(Elec.cor)
se.cov.gmnl(Elec.cor, sd = TRUE)
cor.gmnl(Elec.cor)

## End(Not run)
```

---

`effect.gmnl` *Get the Conditional Individual Coefficients*

---

## Description

This a helper function to obtain the individuals' conditional estimate of the either random parameters or willingness-to-pay.

## Usage

```
effect.gmnl(x, par = NULL, effect = c("ce", "wtp"), wrt = NULL, ...)
```

## Arguments

| | |
|---|---|
| x | an object of class gmnl. |
| par | a string giving the name of the variable with a random parameter. |
| effect | a string indicating what should be computed: the conditional expectation of the individual coefficients "ce", or the conditional expectation of the willingness-to-pay "wtp". |

| | |
|---|---|
| wrt | a string indicating with respect to which variable the willingness-to-pay should be computed. |
| ... | further arguments. Ignorred. |

## Value

A named list where "mean" contains the individuals' conditional mean for the random parameter or willingness-to-pay, and where "sd.est" contains standard errors.

## 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

gmnl for the estimation of multinomial Logit models with individual parameters.

## Examples

```
## Not run:
## Data
data("TravelMode", package = "AER")
library(mlogit)
TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
                  alt.levels = c("air", "train", "bus", "car"), chid.var = "individual")

## MIXL model with observed heterogeneity
mixl.hier <- gmnl(choice ~ vcost + gcost + travel + wait | 1 | 0 | income + size - 1,
                  data = TM,
                  model = "mixl",
                  ranp = c(travel = "t", wait = "n"),
                  mvar = list(travel = c("income","size"), wait = c("income")),
                  R = 30,
                  haltons = list("primes"= c(2, 17), "drop" = rep(19, 2)))

## Get the individuals' conditional mean and their standard errors for lwage
bi.travel <- effect.gmnl(mixl.hier, par = "travel", effect = "ce")
summary(bi.travel$mean)
summary(bi.travel$sd.est)

## Get the individuals' conditional WTP of travel with respect to gcost
wtp.travel <- effect.gmnl(mixl.hier, par = "travel", effect = "wtp", wrt = "gcost")
summary(wtp.travel$mean)
summary(wtp.travel$sd.est)

## End(Not run)
```

## estfun.gmnl *Gradient for Observations*

### Description

It extracts the gradient for each observation evaluated at the estimated parameters for an object of class gmnl.

### Usage

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

### Arguments

| | |
|---|---|
| x | a fitted model of class gmnl. |
| ... | other arguments. Ignored. |

### 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.gmnl *Get Model Summaries for Use with "mtable"*

### Description

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

### Usage

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

### Arguments

| | |
|---|---|
| obj | a gmnl object, |
| alpha | level of the confidence intervals, |
| ... | further arguments, |

### Details

For more details see package **memisc**

---

gFormula *Model Formula for Multinomial Logit Models*

---

### Description

Four kind of variables are used in multinomial choice models with individual heterogeneity: alternative specific and individual specific variables; variables for the mean of the random parameters (deterministic taste variations), and variables for the scale function. gFormula deals with this type of models using suitable methods to extract the elements of the model.

### Usage

```
gFormula(object)

is.gFormula(object)

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

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

### Arguments

| | |
|---|---|
| object | a formula for the gFormula function, for the model.matrix method, a gFormula object, |
| formula | a gFormula object, |
| data | a data.frame, |
| ... | further arguments. |
| lhs | see Formula, |
| rhs | see Formula, |
| Q | number of classes for the latent class model, |

gmnl *Estimate Multinomial Logit Models with Observed and Unobserved Individual Heterogeneity.*

## Description

Estimate different types of multinomial logit models with observed and unobserved individual heterogeneity, such as MIXL, S-MNL, G-MNL, LC and MM-MNL models. These models are estimated using Maximum Simulated Likelihood. It supports both cross-sectional and panel data.

## Usage

```
gmnl(
  formula,
  data,
  subset,
  weights,
  na.action,
  model = c("mnl", "mixl", "smnl", "gmnl", "lc", "mm"),
  start = NULL,
  ranp = NULL,
  R = 40,
  Q = 2,
  haltons = NA,
  mvar = NULL,
  seed = 12345,
  correlation = FALSE,
  bound.err = 2,
  panel = FALSE,
  hgamma = c("direct", "indirect"),
  reflevel = NULL,
  init.tau = 0.1,
  init.gamma = 0.1,
  notscale = NULL,
  print.init = FALSE,
  gradient = TRUE,
  typeR = TRUE,
  ...
)

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

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

## S3 method for class 'summary.gmnl'
print(
  x,
  digits = max(3, getOption("digits") - 2),
  width = getOption("width"),
  ...
)

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

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

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

## S3 method for class 'gmnl'
residuals(object, outcome = TRUE, ...)

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

## S3 method for class 'gmnl'
fitted(object, outcome = TRUE, ...)

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

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

## Arguments

| | |
|---|---|
| formula | a symbolic description of the model to be estimated. The formula is divided in five parts, each of them separated by the symbol \|. The first part is reserved for alternative-specific variables with a generic coefficient. The second part corresponds to individual-specific variables with an alternative specific coefficients. The third part corresponds to alternative-specific variables with an alternative-specific coefficident. The fourth part is reserved for time-invariant variables that modify the mean of the random parameters. Finally, the fifth part is reserved for time-invariant variables that enter in the scale coefficient or in the probability assignment in models with latent classes. |
| data | the data of class mlogit.data. |

| | |
|---|---|
| subset | an optional vector specifying a subset of observations. |
| weights | an optional vector of weights. Default to 1. |
| na.action | a function wich indicated what should happen when the data contains NA's. |
| model | a string indicating which model is estimated. The options are "mnl" for the Multinomial Logit Model, "mixl" for the Mixed Logit Model, "smnl" for the Scaled Multinomial Logit Model, "gmnl" for the Generalized Multinomial Logit Model, "lc" for the Latent Class Multinomial Logit Model, and "mm" for the Mixed-Mixed Multinomial Logit 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 Sb Johnson. |
| R | the number of draws of pseudo-random numbers if ranp is not NULL. |
| Q | number of classes for LC or MM-MNL models. |
| haltons | only relevant if ranp is not NULL. If haltons = NULL, pseudo-random numbers are used instead of Halton sequences. If haltons=NA, the first $K$ primes are used to generates the Halton draws, where $K$ is the number of random parameters, and 15 of the initial sequence of elements are dropped. Otherwise, haltons should be a list with elements prime and drop. |
| 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. |
| seed | seed for the random number generator. Default is seed = 12345. |
| correlation | only relevant if ranp is not NULL. If true, the correlation across random parameters is taken into account. |
| bound.err | only relevant if model is smnl or gmnl. It indicates at which values the draws for the scale parameter are truncated. By default bound.err = 2, therefore a truncated normal distribution with truncation at -2 and +2 is used. |
| panel | if TRUE a panel data model is estimated. |
| hgamma | a string indicated how to estimate the parameter gamma. If "direct", then $\gamma$ is estimated directly, if "indirect" then $\gamma^*$ is estimated, where $\gamma = \exp(\gamma^*)/(1+\exp(\gamma^*))$. |
| reflevel | the base alternative. |
| init.tau | initial value for the $\tau$ parameter. |
| init.gamma | initial value for $\gamma$. |
| notscale | only relevant if model is smnl or gmnl. It is a vector indicating which variables should not be scaled. |
| print.init | if TRUE, the initial values for the optimization procedure are printed. |
| gradient | if TRUE, analytical gradients are used for the optimization procedure. |
| typeR | if TRUE, truncated normal draws are used for the scale parameter, if FALSE the procedure suggested by Greene (2010) is used. |

| | |
|---|---|
| ... | additional arguments to be passed to maxLik, which depend in the maximization routine. |
| x, object | and object of class gmnl. |
| digits | the number of digits. |
| width | width. |
| new | an updated formula for the update method. |
| outcome | if TRUE, then the fitted and residuals methods return a vector that corresponds to the chosen alternative, otherwise it returns a matrix where each column corresponds to each alternative. |

## Details

Let the utility to person $i$ from choosing alternative $j$ on choice occasion $t$ be:

$$U_{ijt} = \beta_i x_{ijt} + \epsilon_{ijt}$$

where $\epsilon_{ijt}$ is i.i.d extreme value, and $\beta_i$ vary across individuals. Each model estimated by gmnl depends on how $\beta_i$ is specified. The options are the following:

1. S-MNL if $\beta_i = \sigma_i \beta$, where the scale $\sigma_i$ varies across individuals.
2. MIXL if $\beta_i = \beta + s\eta_i$, where $\eta_i$ is a draw from some distribution. For example, if $\beta_i \sim N(\beta, s^2)$, then $\eta_i \sim N(0, 1)$.
3. GMNL if $\beta_i = \sigma_i\beta + \gamma s\eta_i + \sigma_i(1-\gamma)s\eta_i$, where $\sigma_i$ is the scale parameter, and $\gamma$ is a parameter that controls how the variance of residual taste heterogeneity varies with scale.
4. LC if $\beta_i = \beta_q$ with probability $w_{iq}$ for $q = 1, ..., Q$, where $Q$ is the total number of classes.
5. MM-MIXL if $\beta_i \sim f(\beta_q, \Sigma_q)$ with probability $w_{iq}$ for $q = 1, ..., Q$, where $Q$ is the total number of classes.

Observed heterogeneity can be also accommodated in the random parameters when the MIXL is estimated by including individual-specific covariates. Specifically, the vector of random coefficients is

$$\beta_i = \beta + \Pi z_i + L\eta_i$$

where $z_i$ is a set of characteristics of individual $i$ that influence the mean of the taste parameters; and $\Pi$ is matrix of parameters. To estimate this model, the fourth part of the formula should be specified along with the mvar argument.

One can also allow the mean of the scale to differ across individuals by including individual-specific characteristics. Thus, the scale parameters can be written as

$$\exp(\bar{\sigma} + \delta h_i + \tau \upsilon_i)$$

where $h_i$ is a vector of attributes of individual $i$. To estimate this model, the fifth part of the formula should include the variables that enter $h_i$.

For models with latent classes, the class assignment is modeled as a semi-parametric multinomial logit format

$$w_{iq} = \frac{\exp(\gamma_q)}{\sum_{q=1}^{Q} \exp(\gamma_q)}$$

for $q = 1, ..., Q, \gamma_1 = 0$. Latent class models (LC and MM-MIXL) requires at least that a constant should be specified in the fifth part of the `formula`. If the class assignment, $w_{iq}$, is also determined by socio-economic characteristics, these variables can be also included in the fifth part.

Models that involve random parameters are estimated using Maximum Simulated Likelihood using the `maxLik` function of `maxLik` package.

## Value

An object of class "`gmnl`" with the following elements

| | |
|---|---|
| `coefficients` | the named vector of coefficients, |
| `logLik` | a set of values of the maximum likelihood procedure, |
| `mf` | the model framed used, |
| `formula` | the formula (a `gFormula` object), |
| `time` | `proc.time()` minus the start time, |
| `freq` | frequency of dependent variable, |
| `draws` | type of draws used, |
| `model` | the fitted model, |
| `R` | number of draws used, |
| `ranp` | vector indicating the variables with random parameters and their distribution, |
| `residuals` | the residuals, |
| `correlation` | whether the model is fitted assuming that the random parameters are correlated, |
| `bi` | matrix of conditional expectation of random parameters, |
| `Q` | number of classes, |
| `call` | the matched call. |

## Author(s)

Mauricio Sarrias <msarrias86@gmail.com>

## References


- Keane, M., & Wasi, N. (2013). Comparing alternative models of heterogeneity in consumer choice behavior. Journal of Applied Econometrics, 28(6), 1018-1045.
- Fiebig, D. G., Keane, M. P., Louviere, J., & Wasi, N. (2010). The generalized multinomial logit model: accounting for scale and coefficient heterogeneity. Marketing Science, 29(3), 393-421.
- Greene, W. H., & Hensher, D. A. (2010). Does scale heterogeneity across individuals matter? An empirical assessment of alternative logit models. Transportation, 37(3), 413-428.
- Train, K. (2009). Discrete choice methods with simulation. Cambridge University Press.


## See Also

`mlogit`, `mlogit.data`, `maxLik`, `Rchoice`

## Examples

```
## Examples using the Fishing data set from the AER package
data("TravelMode", package = "AER")
library(mlogit)
TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
                  alt.levels = c("air", "train", "bus", "car"), chid.var = "individual")
## Not run:
## S-MNL model, ASCs not scaled
smnl <- gmnl(choice ~ wait + vcost + travel + gcost| 1, data = TM,
             model = "smnl", R = 100,
             notscale = c(1, 1, 1, rep(0, 4)))
summary(smnl)

## MIXL model with observed heterogeneity
mixl.hier <- gmnl(choice ~ vcost + gcost + travel + wait | 1 | 0 | income + size - 1,
                  data = TM,
                  model = "mixl",
                  ranp = c(travel = "t", wait = "n"),
                  mvar = list(travel = c("income","size"), wait = c("income")),
                  R = 30,
                  haltons = list("primes"= c(2, 17), "drop" = rep(19, 2)))
summary(mixl.hier)

## Examples using the Electricity data set from the mlogit package
data("Electricity", package = "mlogit")
Electr <- mlogit.data(Electricity, id.var = "id", choice = "choice",
                      varying = 3:26, shape = "wide", sep = "")

## Estimate a MIXL model with correlated random parameters
Elec.cor <- gmnl(choice ~ pf + cl + loc + wk + tod + seas| 0, data = Electr,
                 subset = 1:3000,
                 model = 'mixl',
                 R = 10,
                 panel = TRUE,
                 ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n"),
                 correlation = TRUE)
summary(Elec.cor)
cov.gmnl(Elec.cor)
se.cov.gmnl(Elec.cor)
se.cov.gmnl(Elec.cor, sd = TRUE)
cor.gmnl(Elec.cor)

## Estimate a G-MNL model, where ASCs are also random
Electr$asc2 <- as.numeric(Electr$alt == 2)
Electr$asc3 <- as.numeric(Electr$alt == 3)
Electr$asc4 <- as.numeric(Electr$alt == 4)

Elec.gmnl <- gmnl(choice ~ pf + cl + loc + wk + tod + seas + asc2 + asc3 + asc4 | 0,
                  data = Electr,
                  subset = 1:3000,
                  model = 'gmnl',
                  R = 30,
```

```
                  panel = TRUE,
                  notscale = c(rep(0, 6), 1, 1, 1),
                  ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n",
                  asc2 = "n", asc3 = "n", asc4 = "n"))
summary(Elec.gmnl)

## Estimate a LC model with 2 classes
Elec.lc <- gmnl(choice ~ pf + cl + loc + wk + tod + seas| 0 | 0 | 0 | 1,
               data = Electr,
               subset = 1:3000,
               model = 'lc',
               panel = TRUE,
               Q = 2)
summary(Elec.lc)

## Estimate a MM-MIXL model
Elec.mm <- gmnl(choice ~ pf + cl + loc + wk + tod + seas| 0 | 0 | 0 | 1,
                  data = Electr,
                  subset = 1:3000,
                  model = 'mm',
                  R = 30,
                  panel = TRUE,
                  ranp = c(pf = "n", cl = "n", loc = "n", wk = "n", tod = "n",
                  seas = "n"),
                  Q = 2,
                  iterlim = 500)
summary(Elec.mm)

## End(Not run)
```

---

`plot.gmnl` *Plot of the Distribution of the Conditional Expectation of Random Parameters*

---

## Description

Methods for `gmnl` objects which provide a plot of the distribution of the conditional expectation of the random parameters or the distribution of the conditional willigness-to-pay.

## Usage

```
## S3 method for class 'gmnl'
plot(
  x,
  par = NULL,
  effect = c("ce", "wtp"),
  wrt = NULL,
  type = c("density", "histogram"),
  adjust = 1,
```

```
  main = NULL,
  col = "indianred1",
  breaks = 10,
  ylab = NULL,
  xlab = NULL,
  ind = FALSE,
  id = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| x | an object of class gmnl. |
| par | a string giving the name of the variable with random parameter. |
| effect | a string indicating whether the conditional expectation, "ce", or the WTP, "wtp" should be plotted. |
| wrt | a string indicating with respect to which variable the WTP should be computed if effect = "wtp". |
| type | a string indicating the type of distribution: it can be a histogram or a density of the conditional expectation of the random coefficients or WTP. |
| 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% interval of conditional distribution for each individual is plotted. 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 whom the user want to plot the conditional coefficients. |
| ... | further arguments to be passed to plot or plotCI. |

## 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

gmnl for the estimation of different multinomial models with individual heterogeneity and effect.gmnl.

## Examples

```
## Not run:
## Examples using the Electricity data set from the mlogit package
library(mlogit)
data("Electricity", package = "mlogit")
Electr <- mlogit.data(Electricity, id.var = "id", choice = "choice",
                      varying = 3:26, shape = "wide", sep = "")

## Estimate a MIXL model with correlated random parameters
Elec.cor <- gmnl(choice ~ pf + cl + loc + wk + tod + seas| 0, data = Electr,
                 subset = 1:3000,
                 model = 'mixl',
                 R = 10,
                 panel = TRUE,
                 ranp = c(cl = "n", loc = "n", wk = "n", tod = "n", seas = "n"),
                 correlation = TRUE)

## Plot the density of the conditional expectation distribution of loc
plot(Elec.cor, par = "loc", effect = "ce", type = "density", col = "grey")

## Plot the conditional expectation of loc for each individual
plot(Elec.cor, par = "loc", effect = "ce", ind = TRUE, id = 1:30)

## Plot the WTP for cl
plot(Elec.cor, par = "loc", effect = "wtp", wrt = "pf")

## End(Not run)
```

---

`vcov.gmnl` *vcov method for gmnl objects*

---

## Description

The vcov method for gmnl 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 'gmnl'
vcov(
  object,
  what = c("coefficient", "ranp"),
  type = c("cov", "cor", "sd"),
  se = FALSE,
  Q = NULL,
  digits = max(3, getOption("digits") - 2),
  ...
)
```

### Arguments

| | |
|---|---|
| object | a fitted model of class gmnl, |
| 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, |
| Q | this argument is only valid if the "mm" (MM-MNL) model is estimated. It indicates the class for which the variance-covariance matrix is computed, |
| digits | number of digits, |
| ... | further arguments |

### Details

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

### See Also

gmnl for the estimation of multinomial logit models with random parameters.

---

## wtp.gmnl *Compute Willingness-to-pay*

---

### Description

Compute the willingness-to-pay.

### Usage

```
wtp.gmnl(object, wrt = NULL, digits = max(3, getOption("digits") - 2))
```

### Arguments

| | |
|---|---|
| object | an object of class gmnl. |
| wrt | a string indicating the variable with respect to which the WTP is computed, |
| digits | number of significant digits to be used for most numbers. |

## Details

For each coefficient, this function computes both the point estimate and standard error of WTP with respect to the variable specified in the argument `wrt`. Specifically, let $\beta_k$ be the coefficient for variable $k$, then

$$WTP_k = -\beta_k/\beta_p$$

where $\beta_p$ is the coefficient for the variable specified with the argument `wrt`. Note that, `wtp.gmnl` does not include the negative sign.

`wtp.gmnl` function is a wrapper for the `deltamethod` function of the **msm** package.

## Value

A coefficient matrix with the WTP point estimates and standard errors.

## 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

`deltamethod` for the estimation of the standard errors.

## Examples

```
## Examples using the Electricity data set from the mlogit package
library(mlogit)
data("Electricity", package = "mlogit")
Electr <- mlogit.data(Electricity, id.var = "id", choice = "choice",
                      varying = 3:26, shape = "wide", sep = "")

## Estimate a conditional logit model
clogit <- gmnl(choice ~ pf + cl + loc + wk + tod + seas| 0,
               data = Electr)
wtp.gmnl(clogit, wrt = "pf")
```

# Index


AIC, *2*
AIC.gmnl, 2

BIC, *2*
BIC.gmnl *(*AIC.gmnl*)*, 2
bread, *3*
bread.gmnl, 3

coef.gmnl *(*gmnl*)*, 9
cor.gmnl *(*cov.gmnl*)*, 4
cov.gmnl, 4

deltamethod, *4*, *19*
df.residual.gmnl *(*gmnl*)*, 9

effect.gmnl, 5, *16*
estfun, *7*
estfun.gmnl, 7

fitted.gmnl *(*gmnl*)*, 9
Formula, *8*

getSummary.gmnl, 7
gFormula, 8
gmnl, *3*, *5*, *6*, 9, *16*, *18*

is.gFormula *(*gFormula*)*, 8

logLik.gmnl *(*gmnl*)*, 9

maxLik, *12*, *13*
mlogit, *13*
mlogit.data, *10*, *13*
model.frame.gFormula *(*gFormula*)*, 8
model.matrix.gFormula *(*gFormula*)*, 8
model.matrix.gmnl *(*gmnl*)*, 9

nObs.gmnl *(*gmnl*)*, 9

plot.gmnl, 15
print.gmnl *(*gmnl*)*, 9
print.summary.gmnl *(*gmnl*)*, 9

residuals.gmnl *(*gmnl*)*, 9

se.cov.gmnl *(*cov.gmnl*)*, 4
summary.gmnl *(*gmnl*)*, 9

update.gmnl *(*gmnl*)*, 9

vcov.gmnl, 17

wtp.gmnl, 18