Package ‘mlogit’
April 20, 2018

Version 0.3-0
Date 2018-04-20
Title Multinomial Logit Models
Depends R (>= 2.10), Formula, maxLik
Imports statmod, zoo, MASS, lmtest
Suggests knitr, car, nnet, lattice, AER, ggplot2, texreg
Description Maximum Likelihood estimation of random utility discrete choice models (logit and probit).
VignetteBuilder knitr
License GPL (>= 2)
URL https://cran.r-project.org/package=mlogit,
     https://r-forge.r-project.org/projects/mlogit/
Author Yves Croissant [aut, cre]
Maintainer Yves Croissant <yves.croissant@univ-reunion.fr>
Repository CRAN
Repository/R-Forge/Project mlogit
Repository/R-Forge/Revision 54
Repository/R-Forge/DateTimeStamp 2018-04-20 04:07:04
Date/Publication 2018-04-20 12:35:12 UTC
NeedsCompilation no

R topics documented:

Beef ................................................................. 2
Car ................................................................. 3
Catsup ............................................................. 4
correlation ......................................................... 5
Cracker ............................................................ 6
distribution ......................................................... 7
Beef  

Choice of a beef product

Description

panel data

number of observations: 5 scenario for 47 individuals

observation: individuals

country: United States

Usage

data(Beef)
**Format**

A dataframe containing:

- **id**: individuals identifier
- **product**: one of **fresh**, **lean**, **dietLean**, **organic** and **noPurchase**
- **scenario**: scenario identifier
- **choice**: one if the product is chosen, 0 otherwise
- **price**: the price of the product

**Source**


**References**


---

**Stated Preferences for Car Choice**

<table>
<thead>
<tr>
<th>Car</th>
</tr>
</thead>
</table>

**Description**

- a cross-section
- **number of observations**: 4654
- **observation**: individuals
- **country**: United States

**Usage**

data(Car)

**Format**

A dataframe containing:

- **choice**: choice of a vehicle among 6 propositions
- **college**: college education?
- **hsg2**: size of household greater than 2?
- **coml5**: commute lower than 5 miles a day?
Catsup

Description
a cross-section

number of observations : 2798
observation : individuals
country : United States

Usage
data(Catsup)

Format
A dataframe containing :

id individuals identifiers
choice one of heinz41, heinz32, heinz28, hunts32
disp.z is there a display for brand z?
feat.z is there a newspaper feature advertisement for brand z?
price.z price of brand z
Source


References


correlation

Correlation structure of the random parameters

Description

Functions that extract the correlation structure of a mlogit object

Usage

cor.mlogit(x)
cov.mlogit(x)

Arguments

x an mlogit object with random parameters and correlation=TRUE.

Value

A numerical matrix which returns either the correlation or the covariance matrix of the random parameters.

Author(s)

Yves Croissant
Choice of Brand for Crackers

Description

a cross-section

number of observations: 3292
observation: individuals
country: United States

Usage

data(Cracker)

Format

A dataframe containing:

id  individuals identifiers
choice one of sunshine, keebler, nabisco, private
disp.z is there a display for brand z?
feat.z is there a newspaper feature advertisement for brand z?
price.z price of brand z

Source


References

Functions used to describe the characteristics of estimated random parameters

Description

rpar objects contain all the relevant information about the distribution of random parameters. These functions enable to obtain easily descriptive statistics, density, probability and quantiles of the distribution.

Usage

med(x, ...)
stdev(x, ...)
rg(x, ...)
qrpar(x, ...)
prpar(x, ...)
drpar(x, ...)

## S3 method for class 'rpar'
mean(x, norm = NULL, ...)
## S3 method for class 'rpar'
med(x, norm = NULL, ...)
## S3 method for class 'rpar'
stdev(x, norm = NULL, ...)
## S3 method for class 'rpar'
rg(x, norm = NULL, ...)
## S3 method for class 'mlogit'
mean(x, par = NULL, norm = NULL, ...)
## S3 method for class 'mlogit'
med(x, par = NULL, norm = NULL, ...)
## S3 method for class 'mlogit'
stdev(x, par = NULL, norm = NULL, ...)
## S3 method for class 'mlogit'
rg(x, par = NULL, norm = NULL, ...)

## S3 method for class 'rpar'
stdev(x, norm = NULL, ...)
## S3 method for class 'rpar'
qrpar(x, norm = NULL, ...)
## S3 method for class 'rpar'
prpar(x, norm = NULL, ...)
## S3 method for class 'rpar'
drpar(x, norm = NULL, ...)

## S3 method for class 'mlogit'
qrpar(x, par = 1, y = NULL, norm = NULL, ...)
## S3 method for class 'mlogit'
prpar(x, par = 1, y = NULL, norm = NULL, ...)
## S3 method for class 'mlogit'
drpar(x, par = 1, y = NULL, norm = NULL, ...)

Arguments

- `x`: an `mlogit` or an `rpar` object,
- `norm`: the variable used for normalization if any: for the `mlogit` method, this should be the name of the parameter, for the `rpar` method the absolute value of the parameter,
- `par`: the required parameter(s) for the `mlogit` methods (either the name or the position of the parameter(s). If `NULL`, all the random parameters are used.
- `y`: values for which the function has to be evaluated,
- `...`: further arguments.

Details

`mean`, `med`, `stdev` and `rg` compute respectively the mean, the median, the standard deviation and the range of the random parameter. `qrpar`, `prpar`, `drpar` return functions that compute the quantiles, the probability and the density of the random parameters (note that `sd` and `range` are not generic functions in R and that `median` is, but without `...`).

Value

- a numeric vector for `qrpar`, `drpar` and `prpar`, a numeric vector for `mean`, `stdev` and `med` and a numeric matrix for `rg`.

Author(s)

Yves Croissant

See Also

`mlogit` for the estimation of random parameters logit models and `rpar` for the description of `rpar` objects.

---

**effects.mlogit**

*Marginal effects of the covariates*

Description

The effects method for `mlogit` objects computes the marginal effects of the selected covariate on the probabilities of choosing the alternatives

Usage

```r
## S3 method for class 'mlogit'
effects(object, covariate = NULL,
        type = c("aa", "ar", "rr", "ra"), data = NULL, ...)
```
Arguments

- **object**: a `mlogit` object,
- **covariate**: the name of the covariate for which the effect should be computed,
- **type**: the effect is a ratio of two marginal variations of the probability and of the covariate; these variations can be absolute "a" or relative "r". This argument is a string that contains two letters, the first refers to the probability, the second to the covariate,
- **data**: a `data.frame` containing the values for which the effects should be calculated. The number of lines of this `data.frame` should be equal to the number of alternatives,
- **...**: further arguments.

Value

If the covariate is alternative specific, a $J \times J$ matrix is returned, $J$ being the number of alternatives. Each line contains the marginal effects of the covariate of one alternative on the probability to choose any alternative. If the covariate is individual specific, a vector of length $J$ is returned.

Author(s)

Yves Croissant

See Also

- `mlogit` for the estimation of multinomial logit models.

Examples

```r
data("Fishing", package = "mlogit")
Fish <- mlogit.data(Fishing, varying = c(2:9), shape = "wide", choice = "mode")
m <- mlogit(mode ~ price | income | catch, data = Fish)
# compute a data.frame containing the mean value of the covariates in
# the sample
z <- with(Fish, data.frame(price = tapply(price, index(m)$alt, mean),
                         catch = tapply(catch, index(m)$alt, mean),
                         income = mean(income)))
# compute the marginal effects (the second one is an elasticity
effects(m, covariate = "income", data = z)
effects(m, covariate = "price", type = "rr", data = z)
effects(m, covariate = "catch", type = "ar", data = z)
```
Electricity

*Stated preference data for the choice of electricity suppliers*

**Description**

- **panel data**
- **number of observations**: 4308
- **observation**: households
- **country**: United States

**Usage**

`data(Electricity)`

**Format**

A dataframe containing:

- **choice**: the choice of the individual, one of 1, 2, 3, 4,
- **id**: the individual index,
- **pfi**: fixed price at a stated cents per kWh, with the price varying over suppliers and experiments, for scenario $i=(1, 2, 3, 4)$,
- **cli**: the length of contract that the supplier offered, in years (such as 1 year or 5 years.) During this contract period, the supplier guaranteed the prices and the buyer would have to pay a penalty if he/she switched to another supplier. The supplier could offer no contract in which case either side could stop the agreement at any time. This is recorded as a contract length of 0
- **loci**: is the supplier a local company,
- **wki**: is the supplier a well-known company
- **todi**: a time-of-day rate under which the price is 11 cents per kWh from 8am to 8pm and 5 cents per kWh from 8pm to 8am. These TOD prices did not vary over suppliers or experiments: whenever the supplier was said to offer TOD, the prices were stated as above.
- **seasi**: a seasonal rate under which the price is 10 cents per kWh in the summer, 8 cents per kWh in the winter, and 6 cents per kWh in the spring and fall. Like TOD rates, these prices did not vary. Note that the price is for the electricity only, not transmission and distribution, which is supplied by the local regulated utility.

**Source**


**References**

Kenneth Train’s home page: [http://elsa.berkeley.edu/~train/](http://elsa.berkeley.edu/~train/).
Examples of mixed logit and multinomial probit models

Description
This file contains several fitted mixed logit and multinomial probit models. These examples are used in the vignettes and in the man page and are stored in this file because they are pretty long to compute.

Usage
data(Examples)

Format
These examples are obtained using the "Electricity" and the "Train", "Mode" and "Fishing" data set.

Fishing Choice of Fishing Mode

Description
a cross-section
number of observations: 1182
observation: individuals
country: United States

Usage
data(Fishing)

Format
A dataframe containing:
mode recreation mode choice, one of: beach, pier, boat and charter
price.beach price for beach mode
price.pier price for pier mode
price.boat price for private boat mode
price.charter price for charter boat mode
catch.beach catch rate for beach mode
catch.pier catch rate for pier mode
catch.boat catch rate for private boat mode
catch.charter catch rate for charter boat mode
income monthly income
Source

References

Game

<table>
<thead>
<tr>
<th>Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranked data for gaming platforms</td>
</tr>
</tbody>
</table>

Description

a cross-section

number of observations : 91
observation : individuals
country : Netherlands

Usage
data(Game)

Format

A dataframe containing :

ch.Platform where platform is one of Xbox, PlayStation, PSPortable, GameCube, GameBoy and PC. This variables contain the ranking of the platforms from 1 to 6,

own.Platform these 6 variables are dummies which indicate whether the given platform is already owned by the respondent,
age the age of the respondent,
hours hours per week spent on gaming.

Details

The data are also provided in long format (use in this case data(Game2)). In this case, the alternative and the choice situation are respectively indicated in the platform and chid variables.

Source

References
**Description**

a cross-section

*number of observations*: 250

*observation*: households

*country*: California

**Usage**

data(HC)

**Format**

A dataframe containing:

- **depvar**: heating system, one of gcc (gas central heat with cooling), ecc (electric central resistance heat with cooling), erc (electric room resistance heat with cooling), hpc (electric heat pump which provides cooling also), g (gas central heat without cooling), ec (electric central resistance heat without cooling), er (electric room resistance heat without cooling)
- **ich.z**: installation cost of the heating portion of the system
- **icca**: installation cost for cooling
- **och.z**: operating cost for the heating portion of the system
- **occa**: operating cost for cooling
- **income**: annual income of the household

**References**

Kenneth Train’s home page: [http://elsa.berkeley.edu/~train/](http://elsa.berkeley.edu/~train/).
Usage

data(Heating)

Format

A dataframe containing:

- **idcase**  id
- **depvar**  heating system, one of gc (gas central), gr (gas room), ec (electric central), er (electric room), hp (heat pump)
- **ic.z**  installation cost for heating system z (defined for the 5 heating systems)
- **oc.z**  annual operating cost for heating system z (defined for the 5 heating systems)
- **pb.z**  ratio oc.z/ic.z
- **income**  annual income of the household
- **agehed**  age of the household head
- **rooms**  numbers of rooms in the house

References

Kenneth Train’s home page: [http://elsa.berkeley.edu/~train/](http://elsa.berkeley.edu/~train/).

---

**hmftest**  

**Hausman-McFadden Test**

Description

Test the IIA hypothesis (independence of irrelevant alternatives) for a multinomial logit model.

Usage

hmftest(x, ...)

## S3 method for class 'mlogit'

hmftest(x, z, ...)

## S3 method for class 'formula'

hmftest(x, alt.subset, ...)

Arguments

- **x**  an object of class `mlogit` or a formula,
- **z**  an object of class `mlogit` or a subset of alternatives for the `mlogit` method. This should be the same model as `x` estimated on a subset of alternatives,
- **alt.subset**  a subset of alternatives,
- **...**  further arguments passed to `mlogit` for the `formula` method.
Details

This is an implementation of the Hausman’s consistency test for multinomial logit models. If the independance of irrelevant alternatives applies, the probability ratio of every two alternatives depends only on the characteristics of these alternatives. Consequently, the results obtained on the estimation with all the alternatives or only on a subset of them are consistent, but more efficient in the first case. On the contrary, only the results obtained from the estimation on a relevant subset are consistent. To compute this test, one needs a model estimated with all the alternatives and one model estimated on a subset of alternatives. This can be done by providing two objects of class \texttt{mlogit}, one object of class \texttt{mlogit} and a character vector indicating the subset of alternatives, or a formula and a subset of alternatives.

Value

an object of class "htest".

Author(s)

Yves Croissant

References


Examples

```r
## from Greene's Econometric Analysis p. 731

data("TravelMode",package="AER")
TravelMode <- mlogit.data(TravelMode,choice="choice",shape="long", 
alt.var="mode",chid.var="individual")

## Create a variable of income only for the air mode

TravelMode$avinc <- with(TravelMode,(mode=='air')+income)

## Estimate the model on all alternatives, with car as the base level
## like in Greene's book.

#x <- mlogit(choice=wait+gcost+avinc,TravelMode,reflevel="car")
x <- mlogit(choice=wait+gcost+avinc,TravelMode)

## Estimate the same model for ground modes only (the variable avinc
## must be dropped because it is 0 for every observation

g <- mlogit(choice=wait+gcost,TravelMode,reflevel="car", 
alt.subset=c("car","bus","train"))

## Compute the test
```
JapaneseFDI

Japanese Foreign Direct Investment in European Regions

Description

a cross-section

number of observations : 25764

country : Europe

Usage

data(JapaneseFDI)

Format

A dataframe containing :

firm the investment id

country the country

region the region (nuts1 nomenclature)

choice a dummy indicating the chosen region

choice.c the chosen country

wage wage rate in the region

unemp unemployment rate in the region

elig is the country eligible to european subsidies

area the area of the region

scrate social charge rate (country level)

ctaxrate corporate tax rate (country level)

gdp regional gdp

harris harris’ market potential

krugman krugman’s market potential

domind domestic industry count

japind japan industry count

network network count

Source

.

References

Ketchup

Choice of Brand for Ketchup

Description

a cross-section

*number of observations*: 4956

*observation*: individuals

*country*: United States

Usage

data(Ketchup)

Format

A dataframe containing:

- **hid**: individuals identifiers
- **id**: purchase identifiers
- **choice**: one of heinz, hunts, delmonte, stb (store brand)
- **price.z**: price of brand z

Source


References


logsum

*Compute the log-sum or inclusive value/utility*

Description

The `logsum` function computes the inclusive value, or inclusive utility, which is used to compute the surplus and to estimate the two steps nested logit model.
Usage

\texttt{logsum(coef, X = NULL, formula = NULL, data = NULL,}
\texttt{ type = NULL, output = c("chid", "obs"))}

Arguments

- \texttt{coef}: a numerical vector or a \texttt{mlogit} object, from which the \texttt{coef} vector is extracted,
- \texttt{X}: a matrix or a \texttt{mlogit} object from which the \texttt{model.matrix} is extracted,
- \texttt{formula}: a formula or a \texttt{mlogit} object from which the formula is extracted,
- \texttt{data}: a \texttt{data.frame} or a \texttt{mlogit} object from which the \texttt{model.frame} is extracted,
- \texttt{type}: either "group" or "global": if a group argument has been provided in the \texttt{mlogit.data}, the inclusive values are by default computed for every group, otherwise, a unique global inclusive value is computed for each choice situation,
- \texttt{output}: the shape of the results: if "chid", the results is a vector (if \texttt{type} = "global") or a matrix (if \texttt{type} = "region") with row number equal to the number of choice situation, if "obs" a vector of length equal to the number of lines of the data in long format is returned.

Details

The inclusive value, or inclusive utility, or log-sum is the log of the denominator of the probabilities of the multinomial logit model. If a "group" variable is provided in the \texttt{mlogit.data} function, the denominator can either be the one of the multinomial model or those of the lower model of the nested logit model.

If only one argument (\texttt{coef}) is provided, it should a \texttt{mlogit} object and in this case, the coefficients and the \texttt{model.matrix} are extracted from this model.

In order to provide a different \texttt{model.matrix}, further arguments could be used. \texttt{X} is a matrix or a \texttt{mlogit} from which the \texttt{model.matrix} is extracted. The \texttt{formula-data} interface can also be used to construct the relevant \texttt{model.matrix}.

Value

either a vector or a matrix.

Author(s)

Yves Croissant

See Also

\texttt{mlogit} for the estimation of a multinomial logit model.
Description

Two kinds of variables are used in logit models: alternative specific and individual specific variables. mFormula provides a relevant class to deal with this specificity and suitable methods to extract the elements of the model.

Usage

mFormula(object)
## S3 method for class 'formula'
mFormula(object)
is.mFormula(object)
## S3 method for class 'mFormula'
model.matrix(object, data, ...)
## S3 method for class 'mFormula'
model.frame(formula, data, ..., lhs = NULL, rhs = NULL)

Arguments

object for the mFormula function, a formula, for the update and model.matrix methods, a mFormula object,
formula a mFormula object,
data a data.frame,
 lhs see Formula
 rhs see Formula
... further arguments.

Details

Let J being the number of alternatives. The formula may include alternative-specific and individual specific variables. For the latter, J-1 coefficients are estimated for each variable. For the former, only one (generic) coefficient or J different coefficient may be estimated.

A mFormula is a formula for which the right hand side may contain three parts: the first one contains the alternative specific variables with generic coefficient, i.e. a unique coefficient for all the alternatives; the second one contains the individual specific variables for which one coefficient is estimated for all the alternatives except one of them; the third one contains the alternative specific variables with alternative specific coefficients. The different parts are separated by a “|” sign. If a standard formula is written, it is assumed that there are only alternative specific variables with generic coefficients.

The intercept is necessarily alternative specific (a generic intercept is not identified because only utility differences are relevant). Therefore, it deals with the second part of the formula. As it is usual in R, the default behaviour is to include an intercept. A model without an intercept (which is
mlogit

Multinomial logit model

Description

Estimation by maximum likelihood of the multinomial logit model, with alternative-specific and/or individual specific variables.

Value

an object of class mformula.

Author(s)

Yves Croissant

Examples

data("Fishing", package = "mlogit")
Fish <- mlogit.data(Fishing, varying = c(2:9), shape = "wide", choice = "mode")

# a formula with to alternative specific variables (price and catch) and
# an intercept
f1 <- mFormula(mode ~ price + catch)
head(model.matrix(f1, Fish), 2)

# same, with an individual specific variable (income)
f2 <- mFormula(mode ~ price + catch | income)
head(model.matrix(f2, Fish), 2)

# same, without an intercept
f3 <- mFormula(mode ~ price + catch | income + 0)
head(model.matrix(f3, Fish), 2)

# same as f2, but now, coefficients of catch are alternative specific
f4 <- mFormula(mode ~ price | income | catch)
head(model.matrix(f4, Fish), 2)
Usage

mlogit(formula, data, subset, weights, na.action, start = NULL,
   alt.subset = NULL, reflevel = NULL,
   nests = NULL, un.nest.el = FALSE, unscaled = FALSE,
   heterosc = FALSE, rpar = NULL, probit = FALSE,
   R = 40, correlation = FALSE, halton = NULL,
   random.nb = NULL, panel = FALSE, estimate = TRUE,
   seed = 10, ...)

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

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

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

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

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

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

## S3 method for class 'mlogit'
fitted(object,
   type = c("outcome", "probabilities",
            "linpred", "parameters"),
   outcome = NULL, ...)

## S3 method for class 'mlogit'
predict(object, newdata, returnData = FALSE, ...)

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

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

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

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

## S3 method for class 'mlogit'
coef(object, fixed = FALSE, ...)

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

Arguments

x, object an object of class mlogit

formula a symbolic description of the model to be estimated,

new an updated formula for the update method,
newdata  | a data.frame for the predict method,  
returnData | if TRUE, the data is returned as an attribute,  
data | the data: an mlogit.data object or an ordinary data.frame,  
subset | an optional vector specifying a subset of observations,  
weights | an optional vector of weights,  
na.action | a function which indicates what should happen when the data contains 'NA's,  
start | a vector of starting values,  
alt.subset | a vector of character strings containing the subset of alternative on which the model should be estimated,  
reflevel | the base alternative (the one for which the coefficients of individual-specific variables are normalized to 0),  
nests | a named list of characters vectors, each names being a nest, the corresponding vector being the set of alternatives that belong to this nest,  
un.nest.el | a boolean, if TRUE, the hypothesis of unique elasticity is imposed for nested logit models,  
unscaled | a boolean, if TRUE, the unscaled version of the nested logit model is estimated,  
heterosc | a boolean, if TRUE, the heteroscedastic logit model is estimated,  
rpar | a named vector whose names are the random parameters and values the distribution: 'n' for normal, 'l' for log-normal, 't' for truncated normal, 'u' for uniform,  
probit | if TRUE, a multinomial porbit model is estimated, 
R | the number of function evaluation for the gaussian quadrature method used if heterosc=TRUE, the number of draws of pseudo-random numbers if rpar is not NULL,  
correlation | only relevant if rpar is not NULL, if true, the correlation between random parameters is taken into account,  
halton | only relevant if rpar is not NULL, if not NULL, halton sequence is used instead of pseudo-random numbers. If halton=NA, some default values are used for the prime of the sequence (actually, the primes are used in order) and for the number of elements dropped. Otherwise, halton should be a list with elements prime (the primes used) and drop (the number of elements dropped).  
random.nb | only relevant if rpar is not NULL, a user-supplied matrix of random,  
panel | only relevant if rpar is not NULL and if the data are repeated observations of the same unit; if TRUE, the mixed-logit model is estimated using panel techniques,  
estimate | a boolean indicating whether the model should be estimated or not: if not, the model.frame is returned,  
seed | ,  
digits | the number of digits,  
width | the width of the printing,  
outcome | a boolean which indicates, for the fitted and the residuals methods whether a matrix (for each choice, one value for each alternative) or a vector (for each choice, only a value for the alternative chosen) should be returned,
type

one of outcome (probability of the chosen alternative), probabilities (probabilities for all the alternatives), parameters for individual-level random parameters.

fixed

if FALSE (the default), constant coefficients are not returned.

... further arguments passed to mlogit.data or mlogit.optim.

Details

For how to use the formula argument, see mFormula.

The data argument may be an ordinary data.frame. In this case, some supplementary arguments should be provided and are passed to mlogit.data. Note that it is not necessary to indicate the choice argument as it is deduced from the formula.

The model is estimated using the mlogit.optim function.

The basic multinomial logit model and three important extenstions of this model may be estimated. If heterosc=TRUE, the heteroscedastic logit model is estimated. J-1 extra coefficients are estimated that represent the scale parameter for J-1 alternatives, the scale parameter for the reference alternative being normalized to 1. The probabilities don’t have a closed form, they are estimated using a gaussian quadrature method.

If nests is not NULL, the nested logit model is estimated.

If rpar is not NULL, the random parameter model is estimated. The probabilities are approximated using simulations with R draws and halton sequences are used if halton is not NULL. Pseudo-random numbers are drawn from a standard normal and the relevant transformations are performed to obtain numbers drawn from a normal, log-normal, censored-normal or uniform distribution. If correlation=TRUE, the correlation between the random parameters are taken into account by estimating the components of the cholesky decomposition of the covariance matrix. With G random parameters, without correlation G standard deviations are estimated, with correlation G * (G + 1) /2 coefficients are estimated.

Value

An object of class "mlogit", a list with elements:

coefficients  the named vector of coefficients,
logLik        the value of the log-likelihood,
hessian       the hessian of the log-likelihood at convergence,
gradients      the gradient of the log-likelihood at convergence,
call           the matched call,
est.stat       some information about the estimation (time used, optimisation method),
freq           the frequency of choice,
residuals      the residuals,
fitted.values  the fitted values,
formula        the formula (a mFormula object),
extended.formula the formula (a formula object),
model          the model frame used,
index          the index of the choice and of the alternatives.
Author(s)

Yves Croissant

References


See Also

mlogit.data to shape the data. multinom from package nnet performs the estimation of the multinomial logit model with individual specific variables. mlogit.optim for details about the optimization function.

Examples

```r
## Cameron and Trivedi's Microeconometrics p.493 There are two
## alternative specific variables : price and catch one individual
## specific variable (income) and four fishing mode : beach, pier, boat,
## charter

data("Fishing", package = "mlogit")
Fish <- mlogit.data(Fishing, varying = c(2:9), shape = "wide", choice = "mode")

## a pure "conditional" model
summary(mlogit(mode ~ price + catch, data = Fish))

## a pure "multinomial model"
summary(mlogit(mode ~ 0 | income, data = Fish))

## which can also be estimated using multinom (package nnet)
library("nnet")
summary(multinom(mode ~ income, data = Fishing))

## a "mixed" model
m <- mlogit(mode ~ price + catch | income, data = Fish)
summary(m)

## same model with charter as the reference level
m <- mlogit(mode ~ price + catch | income, data = Fish, reflevel = "charter")

## same model with a subset of alternatives : charter, pier, beach
```
m <- mlogit(mode ~ price+ catch | income, data = Fish, 
     alt.subset = c("charter", "pier", "beach"))

## model on unbalanced data i.e. for some observations, some 
## alternatives are missing

# a data.frame in wide format with two missing prices 
Fishing2 <- Fishing
Fishing2[1, "price.pier"] <- Fishing2[3, "price.beach"] <- NA
mlogit(mode~price+catch|income, Fishing2, shape="wide", choice="mode", varying = 2:9)

# a data.frame in long format with three missing lines 
data("TravelMode", package = "AER")
Tr2 <- TravelMode[-c(2, 7, 9),]
mlogit(choice=wait+gcost|income+size, Tr2, shape = "long", 
     chid.var = "individual", alt.var="mode", choice = "choice")

## An heteroscedastic logit model

data("TravelMode", package = "AER")
hl <- mlogit(choice ~ wait + travel + vcost, TravelMode, 
     shape = "long", chid.var = "individual", alt.var = "mode", 
     method = "bfgs", heterosc = TRUE, tol = 10)

## A nested logit model

TravelMode$saveincome <- with(TravelMode, income * (mode == "air"))
TravelMode$time <- with(TravelMode, travel + wait)/60
TravelMode$timeair <- with(TravelMode, time * I(mode == "air"))
TravelMode$income <- with(TravelMode, income / 10)

# Hensher and Greene (2002), table 1 p.8-9 model 5 
TravelMode$incomeother <- with(TravelMode, ifelse(mode %in% c('air', 'car'), income, 0))
nl <- mlogit(choice=gcost+wait+incomeother, TravelMode, 
     shape="long", alt.var='mode', 
     nests=list(public=c('train', 'bus'), other=c('car','air'))) 

# same with a comon nest elasticity (model 1) 
nl2 <- update(nl, un.nest.el = TRUE)

## a probit model

## Not run:
pr <- mlogit(choice ~ wait + travel + vcost, TravelMode, 
     shape = "long", chid.var = "individual", alt.var = "mode", 
     probit = TRUE)

## End(Not run)

## a mixed logit model

## Not run: 
rpl <- mlogit(mode ~ price+ catch | income, Fishing, varying = 2:9,
shape = 'wide', rpar = c(price= 'n', catch = 'n'),
correlation = TRUE, halton = NA,
 R = 10, tol = 10, print.level = 0)

summary(rpl)
rpar(rpl)
cor.mlogit(rpl)
cov.mlogit(rpl)
rpar(rpl, "catch")
summary(rpar(rpl, "catch"))

## End(Not run)

# a ranked ordered model
data("Game", package = "mlogit")
g <- mlogit(ch~own|hours, Game, choice='ch', varying = 1:12,
ranked=TRUE, shape="wide", reflevel="PC")

---

**mlogit.data**

data.frame for logit model

**Description**

shape a data.frame in a suitable form for the use of the mlogit function.

**Usage**

mlogit.data(data, choice = NULL, shape = c("long", "wide"), varying = NULL,
sep=""," alt.var = NULL, chid.var = NULL, alt.levels = NULL,
id.var = NULL, group.var = NULL, opposite = NULL,
drop.index = FALSE, ranked = FALSE, subset = NULL, ...)

## S3 method for class 'mlogit.data'
mean(x, ...)

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

## S3 method for class 'mlogit.data'
formula(x, ...)

**Arguments**

data a data.frame,
x a mlogit.data or a pseries object,
choice the variable indicating the choice made: it can be either a logical vector, a numerical vector with 0 where the alternative is not chosen, a factor with level 'yes' when the alternative is chosen
shape the shape of the data.frame: whether long if each row is an alternative or wide if each row is an observation,
mlogit.data

varying the indexes of the variables that are alternative specific,
sep the separator of the variable name and the alternative name (only relevant for a wide data.frame),
alt.var the name of the variable that contains the alternative index (for a long data.frame only) or the name under which the alternative index will be stored (the default name is alt),
chid.var the name of the variable that contains the choice index or the name under which the choice index will be stored,
alt.levels the name of the alternatives: if null, for a wide data.frame, they are guessed from the variable names and the choice variable (both should be the same), for a long data.frame, they are guessed from the alt.var argument,
id.var the name of the variable that contains the individual index if any,
group.var the name of the variable that contains the group index if any,
opposite returns the opposite of the specified variables,
drop.index should the index variables be dropped from the data.frame,
ranked a logical value which is true if the response is a rank,
subset a logical expression which defines the subset of observations to be selected,
... further arguments passed to reshape.

Value

A mlogit.data object, which is a data.frame in long format, i.e. one line for each alternative. It has a index attribute, which is a data.frame that contains the index of the choice made (‘chid’), the index of the alternative (‘alt’) and, if any, the index of the individual (‘id’). The choice variable is a boolean which indicates the choice made. This function use reshape if the data.frame is in wide format.

Author(s)

Yves Croissant

See Also

reshape

Examples

# ModeChoice is a long data.frame
data("TravelMode", package = "AER")
TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
   alt.levels = c("air", "train", "bus", "car"))

# Same but the alt variable called mode is provided
TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
   alt.var = "mode")
mlogit.optim

Non-linear minimization routine

```r
alt.var = "mode")

# Same but the chid variable called individual is provided
TM <- mlogit.data(TravelMode, choice = "choice",
   shape = "long", id.var = "individual",
   alt.levels = c("air", "train", "bus", "car"))

# Same but with two own provided variables
TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
   id.var = "individual", alt.var = "mode")

# Same but with two own provided variables which are deleted from the
# data.frame
TM <- mlogit.data(TravelMode, choice = "choice", shape = "long",
   id.var = "individual", alt.var = "mode", drop.index = TRUE)

# Train is a wide data.frame with columns 'choiceid' is the choice
# index, the alternatives are named "ch1" and "ch2", the opposite of
# the variables is returned
data("Train", package = "mlogit")
Train <- mlogit.data(Train, choice = "choice", shape = "wide",
   varying = 4:11, alt.levels = c("A", "B"), sep = "-_",
   opposite = c("price", "time", "change", "comfort"))

# Car is a wide data.frame
data("Car", package = "mlogit")
Car <- mlogit.data(Car, varying = 5:70, shape = "wide", sep = "",
   choice = "choice", alt.levels = 1:6)

data("HC", package = "mlogit")
HC <- mlogit.data(HC, choice = "depvar", varying=c(2:8, 10:16), shape="wide")

# Game is a data.frame in wide format for which the response is a
# ranking variable
data("Game", package = "mlogit")
G <- mlogit.data(Game, shape="wide", varying = 1:12, alt.var = 'platform',
   drop.index = TRUE, choice="ch", ranked =TRUE)

# Game2 contains the same data, but in long format
data("Game2", package = "mlogit")
G2 <- mlogit.data(Game2, shape='long', choice="ch", alt.var = 'platform', ranked = TRUE)
```
mlogit.optim

Description

This function performs efficiently the optimization of the likelihood functions for multinomial logit models.

Usage

mlogit.optim(logLik, start, method = c("bfgs", "nr", "bhhh"), iterlim = 2000,
            tol = 1e-06, ftol = 1e-08, steptol = 1e-10,
            print.level = 0, constPar = NULL, ...)

Arguments

logLik the likelihood function to be maximized,
start the initial value of the vector of coefficients,
method the method used, one of 'nr' for Newton-Ralphson, 'bhhh' for Berndt-Hausman-Hall-Hall and 'bfgs'.
iterlim the maximum number of iterations,
tol the value of the criteria for the gradient,
ftol the value of the criteria for the function,
steptol the value of the criteria for the step,
print.level one of (0, 1, 2), the details of the printing messages. If 'print.level=0', no information about the optimization process is provided, if 'print.level=1' the value of the likelihood, the step and the stoping criteria is printing, if 'print.level=2' the vectors of the parameters and the gradient are also printed.
constPar a numeric or a character vector which indicates that some parameters should be treated as constant,
... further arguments passed to f.

Details

The optimization is performed by updating, at each iteration, the vector of parameters by the amount step * direction, where step is a positive scalar and direction = H^-1 * g, where g is the gradient and H^-1 is an estimation of the inverse of the hessian. The choice of H^-1 depends on the method chosen:

if method= 'nr', H is the hessian (i.e. is the second derivates matrix of the likelihood function),
if method = 'bhhh', H is the outer-product of the individual contributions of each individual to the gradient,
if method = 'bfgs', H^-1 is updated at each iteration using a formula that uses the variations of the vector of parameters and the gradient. The initial value of the matrix is the inverse of the outer-product of the gradient (i.e. the bhh estimator of the hessian).

The initial step is 1 and, if the new value of the function is less than the previous value, it is divided by two, until a higher value is obtained.

The routine stops when the gradient is sufficiently close to 0. The criteria is g * H^-1 * g which is compared to the tol argument. It also may stops if the number of iterations equals iterlim.
The function \( f \) has an initial value argument which is the initial value of the likelihood. The function is then evaluated a first time with a step equals to one. If the value is lower than the initial value, the step is divided by two until the likelihood increases. The gradient is then computed and the function returns as attributes the gradient is the step. This method is more efficient than other functions available for R:

For the \texttt{optim} and the \texttt{maxLik} functions, the function and the gradient should be provided as separate functions. But, for multinomial logit models, both depends on the probabilities which are the most time-consuming elements of the model to compute.

For the \texttt{nlm} function, the function returns the gradient as an attribute. The gradient is therefore computed at each iteration, even when the function is computed with a step that is unable to increase the value of the likelihood.

Previous versions of \texttt{mlogit} depended on the \texttt{maxLik} package. We kept the same interface, namely the \texttt{start,method,iterlim,tol,print.level} and \texttt{constPar} arguments.

The default method is \texttt{'bfgs'}, which is known to perform well, even if the likelihood function is not well behaved and the default value for \texttt{print.level}=1, which means moderate printing.

A special default behavior is performed if a simple multinomial logit model is estimated. Indeed, for this model, the likelihood function is concave, the analytical hessian is simple to write and the optimization is straightforward. Therefore, in this case, the default method is \texttt{'nr'} and \texttt{print.level}=0.

Value

a list that contains the followings elements:

- **optimum**: the value of the function at the optimum, with attributes:
  - \texttt{grad} a matrix that contains the contribution of each individual to the gradient,
  - \texttt{gradient} the gradient and, if \texttt{method='nr} hessian the hessian,
- **coefficients**: the vector of the parameters at the optimum,
- **est.stat**: a list that contains some information about the optimization:
  - \texttt{nb.iter} the number of iterations,
  - \texttt{eps} the value of the stopping criteria,
  - \texttt{method} the method of optimization method used,
  - \texttt{message}

Author(s)

Yves Croissant

---

**MobilePhones**  
*Stated Preferences survey for mobile phones*

**Description**

a cross-section from 2003

*number of observations*: 11184

*observation*: individuals

*country*: Netherland
Mode Choice

Description

a cross-section

number of observations : 453

observation : individuals

Usage

data(Mode)

Format

A dataframe containing :

choice one of car, carpool, bus or rail

cost.z cost of mode z

time.z time of mode z

Source


References

References

Kenneth Train’s home page: http://elsa.berkeley.edu/~train/.

ModeCanada

Mode Choice for the Montreal-Toronto Corridor

Description

A cross-section

number of observations: 3880
observation: individuals

Usage

data(ModeCanada)

Format

A dataframe containing:

- **case**: the individual index
- **alt**: the alternative, one of train, car, bus and air,
- **choice**: one if the mode is chosen, zero otherwise,
- **cost**: monetary cost,
- **ivt**: in vehicule time,
- **ovt**: out vehicule time,
- **frequency**: frequency,
- **income**: income,
- **urban**: urban,
- **noalt**: the number of alternatives available.

References


Examples

```r
data("ModeCanada", package = "mlogit")
bususers <- with(ModeCanada, case[choice == 1 & alt == "bus"])
ModeCanada <- subset(ModeCanada, ! case %in% bususers)
ModeCanada <- subset(ModeCanada, noalt == 4)
ModeCanada <- subset(ModeCanada, alt != "bus")
ModeCanada$alt <- ModeCanada$alt[drop = TRUE]
KoppWen00 <- mlogit.data(ModeCanada, shape='long', chid.var = 'case',
                        alt.var = 'alt', choice='choice',
                        drop.index=TRUE)
pcl <- mlogit(choice~freq+cost+ivt+ovt, KoppWen00, reflevel='car',
             nests='pcl', constPar=c('iv.train.air'))
```

Description

- a cross-section
- number of observations: 9480
- country: United States
- economic topic: microeconomics
- econometrics topic: discrete choice

Usage

```r
data("NOx")
```

Format

A dataframe containing:

- `chid` the plant id,
alt  the alternative,
id  the owner id,
choice  the chosen alternative,
available  a dummy indicating that the alternative is available,
env  the regulatory environment, one of 'regulated', 'deregulated' and 'public',
post  dummy for post-combustion pollution control technology,
cm  dummy for combustion modification technology,
lmb  dummy for low NOx burners technology,
age  age of the plant (in deviation from the mean age).
vcost  variable cost,
kcost  capital cost,

Source

References

plot.mlogit

Plot of the distribution of estimated random parameters

Description
Methods for rpar and mlogit objects which provide a plot of the distribution of one or all of the estimated random parameters.

Usage

## S3 method for class 'mlogit'
plot(x, par = NULL, norm = NULL,
     type = c("density", "probability"), ...)

## S3 method for class 'rpar'
plot(x, norm = NULL, type = c("density","probability"), ...)

Arguments

x  a mlogit or a rpar object,
type  the function to be plotted, whether the density or the probability density function,
par  a subset of the random parameters ; if NULL, all the parameters are selected,
norm  the coefficient’s name for the mlogit method or the coefficient’s value for the rpar method used for normalization,
...  further arguments, passed to plot.rpar for the mlogit method and to plot for the rpar method.
Details

For the rpar method, one plot is drawn. For the mlogit method, one plot for each selected random parameter is drawn.

Author(s)

Yves Croissant

See Also

mlogit for the estimation of random parameters logit models and rpar for the description of rpar objects and distribution for functions which return informations about the distribution of random parameters.

---

RiskyTransport | Risky Transportation Choices

Description

a cross-section

number of observations: 5405
country: Sierra Leone

Usage

data(RiskyTransport)

Format

A dataframe containing:

id individual id
choice 1 for the chosen mode
mode one of Helicopter, WaterTaxi, Ferry and Hovercraft
cost the generalised cost of the transport mode
risk the fatality rate, numbers of death per 100,000 trips
weight weights
seats
noise
crowdness
convloc
clientele
chid choice situation id
african yes if born in Africa, no otherwise
lifeExp declared life expectancy
dwage declared hourly wage
iwage imputed hourly wage
educ level of education, one of low and high
fatalism self-ranking of the degree of fatalism
gender gender, one of female and male
age age
haveChildren yes if the traveler has children, no otherwise
swim yes if the traveler knows how to swim, no otherwise

Source

References

rpar random parameter objects

Description
rpar objects contain the relevant information about estimated random parameters. The homonymous function extract on rpar object from a mlogit object.

Usage
rpar(x, par = NULL, norm = NULL, ...)

Arguments
x a mlogit object,
par the name or the index of the parameters to be extracted; if NULL, all the parameters are selected,
norm the coefficient used for normalization if any,
... further arguments.

Details
mlogit objects contain an element called rpar which contain a list of rpar objects, one for each estimated random parameter. The print method prints the name of the distribution and the parameter, the summary behave like the one for numeric vectors.
The three tests for mlogit models

Description

Three tests for mlogit models: specific methods for the Wald test and the likelihood ration test and a new function for the score test

Usage

scoretest(object, ...)  
## S3 method for class 'mlogit'
scoretest(object, ...)  
## S3 method for class 'mlogit'
waldtest(object, ...)  
## S3 method for class 'mlogit'
lrtest(object, ...)

Arguments

object           an object of class mlogit or a formula,
...              two kinds of arguments can be used. If "mlogit" arguments are introduced, initial model is updated using these arguments. If "formula" or other "mlogit" models are introduced, the standard behavior of "waldtest" and "lrtest" is followed.
Details

The "scoretest" function and "mlogit" method for "waldtest" and "lrtest" from the "lmtest" package provides the infrastructure to compute the three tests of hypothesis for "mlogit" objects.

The first argument must be a "mlogit" object. If the second one is a fitted model or a formula, the behaviour of the three functions is the one of the default methods of "waldtest" and "lrtest": the two models provided should be nested and the hypothesis tested is that the constrained model is the 'right' model.

If no second model is provided and if the model provided is the constrained model, some specific arguments of "mlogit" should be provided to describe how the initial model should be updated. If the first model is the unconstrained model, it is tested versus the ‘natural’ constrained model; for example, if the model is a heteroscedastic logit model, the constrained one is the multinomial logit model.

Value

an object of class "htest".

Author(s)

Yves Croissant

Examples

library("mlogit")
data("TravelMode", package = "AER")
ml <- mlogit(choice ~ wait + travel + vcost, TravelMode,
shape = "long", chid.var = "individual", alt.var = "mode")

hl <- mlogit(choice ~ wait + travel + vcost, TravelMode,
shape = "long", chid.var = "individual", alt.var = "mode",
method = "bfgs", heterosc = TRUE)

lrtest(ml, hl)
waldtest(hl)
scoretest(ml, heterosc = TRUE)

Telephone

Choice among residential telephone service options for local calling

Description

a cross-section from 1984

number of observations : 434
observation : households
country : United-States

Usage

data(Train)
**Format**

A dataframe containing:

- **choice** a logical which indicates if the alternative has been chosen
- **service** a telephone service option
- **household** the household index
- **cost** the logarithm of the cost

**Source**


**References**


<table>
<thead>
<tr>
<th>TollRoad</th>
<th>Stated Preferences survey for a toll road</th>
</tr>
</thead>
</table>

**Description**

- a panel from 1999-2000

  - number of observations: 1448 observations for 548 individuals
  - observation: individuals
  - country: United-States

**Usage**

`data(TollRoad)`

**Format**

A dataframe containing:

- **id** the individual id
- **src** the source of the data, one of br: revealed preference survey conducted by the Brookings Institution, bs: stated preferences survey conducted by the Brookings Institution and cal: revealed preferences survey conducted by the Californian Polytechnic State University
- **route** the route chosen, one of express (the toll-road) or freeway (the free road)
- **toll.alt** the monetary cost of the road (a for the free road)
time.alt the median time of the trip on both highways for the given schedule
reliability.alt the reliability of the trip length on both highways for the given schedule, measured by the difference between the 80th and the 50th percentile of the trip length
occupance the number of people in the car
size the household size
sex one of male or female
flexibility does the respondent declare having a flexible arrival time, a factor with levels yes or no
distance trip distance in miles
commute a long-commute for trips longer than 45 minutes (a factor with levels yes or no)
age3050 a factor with levels yes if the respondent is between 30 and 50 years old, no otherwise
income a factor with levels low, medium and high

Source

References
Econometrica data archive

---

**Train**

**Stated Preferences for Train Traveling**

**Description**

a cross-section from 1987

*number of observations*: 2929

*observation*: individuals

*country*: Netherland

**Usage**

data(Train)

**Format**

A dataframe containing:

*id* individual identifient

*choiceid* choice identifient

*choice* one of ’A’ or ’B’

*price\_z* price of proposition z (z = ’A’, ’B’) in cents of guilders

*time\_z* travel time of proposition z (z = ’A’, ’B’) in minutes

*comfort\_z* comfort of proposition z (z = ’A’, ’B’), 0, 1 or 2 in decreasing comfort order

*change\_z* number of changes for proposition z (z = ’A’, ’B’)
Source


References


---

**Tuna**

*Choice of Brand for Tuna*

**Description**

- a cross-section
- number of observations: 13705
- observation: individuals
- country: United States

**Usage**

`data(Tuna)`

**Format**

A dataframe containing:

- `hid`: individuals identifiers
- `id`: purchase identifiers
- `choice`: one of skw (Starkist water), cosw (Chicken of the sea water), pw (store-specific private label water), sko (Starkist oil), coso (Chicken of the sea oil)
- `price.z`: price of brand z

**Source**


**References**

Description

The `vcov` method for `mlogit` objects extract the covariance matrix of the coefficients, the errors or the random parameters.

Usage

```r
## S3 method for class 'mlogit'
vcov(object, what = c('coefficient', 'errors', 'rpar'),
      type = c('cov', 'cor', 'sd'), reflevel = NULL, ...)
```

Arguments

- `object`: a `mlogit` object,
- `what`: indicates which covariance matrix has to be extracted: the default value is coefficients, in this case, `vcov` behaves as usual. If `what` equals `errors` the covariance matrix of the errors of the model is returned. Finally, if `what` equals `rpar`, the covariance matrix of the random parameters are extracted.
- `type`: with this argument, the covariance matrix may be returned (the default); the correlation matrix and the standard deviation vector may also be extracted.
- `reflevel`: relevent for the extraction of the errors of a multinomial probit model; in this case the covariance matrix is of error differences is returned and, with this argument, the alternative used for differentiation is indicated.
- `...`: further arguments.

Details

This new interface replaces the `cor.mlogit` and `cov.mlogit` functions which are deprecated.

Author(s)

Yves Croissant

See Also

- `mlogit` for the estimation of multinomial logit models.
**Yogurt**  

*Choice of Brand for Yogurts*

---

**Description**

a cross-section

*number of observations*: 2412  
*observation*: individuals  
*country*: United States

**Usage**

data(Yogurt)

**Format**

A dataframe containing:

- **id**: individuals identifiers
- **choice**: one of yoplait, dannon, hiland, weight (weight watcher)
- **feat.z**: is there a newspaper feature advertisement for brand z?
- **price.z**: price of brand z

**Source**


**References**

Index

*Topic **attribute**
mlogit.data, 26

*Topic **datasets**
Beef, 2
Car, 3
Catsup, 4
Cracker, 6
Electricity, 10
Examples, 11
Fishing, 11
Game, 12
HC, 13
Heating, 13
JapaneseFDI, 16
Ketchup, 17
MobilePhones, 30
Mode, 31
ModeCanada, 32
NOx, 33
RiskyTransport, 35
Telephone, 38
TollRoad, 39
Train, 40
Tuna, 41
Yogurt, 43

*Topic **htest**
hmftest, 14
scoretest, 37

*Topic **models**
mFormula, 19

*Topic **regression**
correlation, 5
distribution, 7
effects.mlogit, 8
logsum, 17
mlogit, 20
mlogit.optim, 29
plot.mlogit, 34
rpar, 36

dcov.mlogit, 42
[[<-.mlogit.data(mlogit.data), 26
$<-.mlogit.data(mlogit.data), 26

Beef, 2
Car, 3
Catsup, 4
ccoef.mlogit(mlogit), 20
ccoef.summary.mlogit(mlogit), 20
cor.mlogit(correlation), 5
correlation, 5
cov.mlogit(correlation), 5
Cracker, 6
df.residual.mlogit(mlogit), 20
distribution, 7, 35
drpar(distribution), 7
effects.mlogit, 8
Elec.mx1(Examples), 11
Elec.mx12(Examples), 11
Elec.mx13(Examples), 11
Elec.mx14(Examples), 11
Elec.mx15(Examples), 11
Electricity, 10
Examples, 11
Fish.mprobit(Examples), 11
Fishing, 11
fitted.mlogit(mlogit), 20
formula.mlogit.data(mlogit.data), 26

Game, 12
Game2(Game), 12

HC, 13
Heating, 13
hmftest, 14

index(mlogit.data), 26
is.mFormula (mFormula), 19
JapaneseFDI, 16
Ketchup, 17
logLik.mlogit (mlogit), 20
logsum, 17
lrtest (scoretest), 37
mean.mlogit (distribution), 7
mean.mlogit.data (mlogit.data), 26
mean.rpar (distribution), 7
med (distribution), 7
mFormula, 19, 23
mlogit, 8, 9, 18, 20, 35, 37, 42
mlogit.data, 23, 24, 26
mlogit.optim, 23, 24, 28
MobilePhones, 30
Mode, 31
ModeCanada, 32
model.frame.mFormula (mFormula), 19
model.matrix.mFormula (mFormula), 19
model.matrix.mlogit (mlogit), 20
multinom, 24

NOx, 33
p1 (Examples), 11
p2 (Examples), 11
plot.mlogit, 34
plot.rpar (plot.mlogit), 34
predict.mlogit (mlogit), 20
print.mlogit (mlogit), 20
print.pseries (mlogit.data), 26
print.rpar (rpar), 36
print.summary.mlogit (mlogit), 20
prpar (distribution), 7
qrpar (distribution), 7
residuals.mlogit (mlogit), 20
rg (distribution), 7
RiskyTransport, 35
rpar, 8, 35, 36
scoretest, 37
stdev (distribution), 7
suml (mlogit), 20
summary.mlogit (mlogit), 20
summary.rpar (rpar), 36
Telephone, 38
terms.mlogit (mlogit), 20
TollRoad, 39
Train, 40
Train.ml (Examples), 11
Train.mxl (Examples), 11
Train.mxl (Examples), 11
Tuna, 41
update.mlogit (mlogit), 20
vcov.mlogit, 42
waldtest (scoretest), 37
Yogurt, 43