Package ‘csampling’

October 12, 2022

Version 1.2-2.1
Date 2014-03-31
Title Functions for Conditional Simulation in Regression-Scale Models
Author S original by Alessandra R. Brazzale <alessandra.brazzale@unipd.it>,
R port by Alessandra R. Brazzale <alessandra.brazzale@unipd.it>.
Maintainer Alessandra R. Brazzale <alessandra.brazzale@unipd.it>
Depends R (>= 3.0.0), marg, statmod, survival
Description Monte Carlo conditional inference for the parameters of a
linear nonnormal regression model.
License GPL (>= 2) | file LICENCE
LazyLoad yes
LazyData yes
NeedsCompilation no
Repository CRAN
Date/Publication 2018-04-16 15:01:44 UTC

R topics documented:

  csampling-package .................................................. 2
  Laplace .................................................................. 2
  make.sample.data .................................................... 4
  mt ........................................................................ 6
  plot.Lapl ................................................................. 7
  rsm.sample ............................................................... 8

Index 11
Laplace

Description

Monte Carlo conditional inference for the parameters of a linear nonnormal regression model

Details

Package: csampling
Version: 1.2-0
Date: 2009-10-03
Depends: R (>= 2.6.0), marg, statmod, survival
License: GPL (>= 2)
LazyLoad: yes
LazyData: yes

Index:

Functions:

Laplace Calculate Laplace's Marginal Density Approximation
dmt Multivariate Student t Distribution
make.sample.data Create a Conditional Sampling Data Object
plot.Lapl.spl Plot uni- and bivariate approximate marginal densities
rsm.sample Conditional Sampler for Regression-Scale Models

Author(s)

S original by Alessandra R. Brazzale <alessandra.brazzale@unimore.it>. R port by Alessandra R. Brazzale <alessandra.brazzale@unimore.it>.
Maintainer: Alessandra R. Brazzale <alessandra.brazzale@unimore.it>

Laplace

Calculate Laplace's Marginal Density Approximation

Description

Calculates the Laplace approximation to the uni- and bivariate marginal densities of components of the MLE in a regression-scale model. The reference distribution is the conditional distribution given the ancillary.
Laplace

Usage

Laplace(which = stop("no choice made"), data = stop("data are missing"),
val1, idx1, val2, idx2, log.scale = TRUE)

Arguments

which
the kind of marginal density that should be approximated. Possible choices
are c (univariate: regression coefficient), s (univariate: scale parameter), cc
(bivariate: two regression coefficients) and cs (bivariate: regression coefficient
and scale parameter).

data
a special conditional sampling data object. This object must be a list with the
following elements:

anc the vector containing the values of the ancillary; usually the Pearson resid-
uals. It has to be of the same length than the number of observations in the
linear regression model.

X the model matrix. It may be obtained applying model.matrix to the fitted rsm
object of interest. The number of observations has to be the same than the
dimension of the ancillary, and the number of covariates must correspond
to the number of regression coefficients defined in the coef component.

coeff the vector of true values of the regression coefficients, that is, the values
used in the simulation study.

disp the true value of the scale parameter used in the simulation study.

family a family.rsm object characterizing the error distribution of the linear
regression model. The following generator functions are available in the
marg package of the R package bundle hoa: student (Student's t), extreme
(Gumbel or extreme value), logistic, logWeibull, logExponential, logRayleigh
and Huber (Huber's least favourable). The demonstration file 'margdemo.R'
that accompanies the marg package shows how to create a new generator
function.

fixed a logical value. If TRUE the scale parameter is known.

The make.sample.data function can be used to create this data object from a
fitted rsm model.

val1 sequence of values for the first MLE at which to calculate the density.

idx1 index of the first regression coefficient, that is, its position in the vector MLE.

val2 sequence of values for the second MLE at which to calculate the density.

idx2 index of the second regression coefficient, that is, its position in the vector MLE.

log.scale logical value. If TRUE the approximation is calculated on the log scale. Highly
recommended. The default is TRUE.

Details

Laplace's integral approximation method is used in order to avoid multi-dimensional numerical
integration. The uni- and bivariate approximations to the marginal distributions give insight into
how the multivariate conditional distribution of the MLE vector is structured. Methods are available
to plot them. They help in choosing a suitable candidate generation density to be used in the `rsm.sample` function.

All information is supplied through the data argument. Note that the user has to keep to the structure described above. If a conditional simulation is to be performed for a fitted `rsm` object, the `make.sample.data` function can be used to generate this special object. The logical switch `fixed` in the conditional sampling data object must be specified.

Value

Returns a `Lapl.spl` or `Lapl.cont` object with the approximate uni- or bivariate conditional distribution of one or two components of the MLE.

Demonstration

The file `csamplingdemo.R` contains code that can be used to run a conditional simulation study similar to the one described in Brazzale (2000, Section 7.3) using the data given in Example 3 of DiCiccio, Field and Fraser (1990).

References


See Also

`make.sample.data`, `rsm.sample`, `family.rsm.object`.

---

### Description

Uses a fitted `rsm` model to create the data object used by the conditional sampler `rsm.sample`.

### Usage

```r
make.sample.data(rsmObject)
```

### Arguments

- `rsmObject` a fitted `rsm` object.
**Value**

Returns a *conditional sampling* data object such as needed by the `rsm.sample` function. This object is a list with the following elements:

- `anc`: the vector containing the values of the ancillary; usually the Pearson residuals. It has to be of the same length than the number of observations in the linear regression model.
- `X`: the model matrix. It may be obtained applying `model.matrix` to the fitted `rsm` object of interest. The number of observations has to be the same than the dimension of the ancillary, and the number of covariates must correspond to the number of regression coefficients defined in the `coef` component.
- `coef`: the vector of *true* values of the regression coefficients, that is, the values used in the simulation study.
- `disp`: the *true* value of the scale parameter used in the simulation study.
- `family`: a `family.rsm` object characterizing the error distribution of the linear regression model. The following generator functions are available in the `marg` package of the R package bundle `hoa`: `student` (Student’s t), `extreme` (Gumbel or extreme value), `logistic`, `logWeibull`, `logExponential`, `logRayleigh` and `Huber` (Huber’s least favourable). The demonstration file ‘`margdemo.R`’ that accompanies the `marg` package shows how to create a new generator function.
- `fixed`: a logical value. If TRUE the scale parameter is known.

The `make.sample.data` function can be used to create this data object from a fitted `rsm` model.

**Demonstration**

The file ‘`csamplingdemo.R`’ contains code that can be used to run a conditional simulation study similar to the one described in Brazzale (2000, Section 7.3) using the data given in Example 3 of DiCiccio, Field and Fraser (1990).

**References**


**See Also**

`rsm.object`, `rsm.sample`
Description

Density and random number generation for the multivariate Student t distribution.

Usage

dmt(x, df=stop("'df' argument is missing, with no default"),
    mm=rep(0, length(x)), cov=diag(rep(1, length(x))))

rmt(n, df=stop("'df' argument is missing, with no default"),
    mm=rep(0, mult), cov=diag(rep(1, mult)), mult, is.chol=FALSE)

Arguments

- **x**: a single multivariate observation. Missing values (NAs) are allowed.
- **n**: the sample size. If length(n) is larger than 1, then length(n) random vectors are returned, bound together in a length(n) times mult matrix, where mult is the dimension of the multivariate variable.
- **df**: the degrees of freedom. In rmt this is replicated to be of the same length than the number of deviates generated by rmt.
- **mult**: the dimension of the multivariate Student t variate.
- **mm**: a vector location parameter. The default is a vector of 0’s.
- **cov**: a square scale matrix. The default is the identity matrix.
- **is.chol**: logical flag. If TRUE, the argument cov is the result from the Choleski decomposition of the original scale matrix.

Value

Returns the density (dmt) of or a random sample (rmt) from the multivariate Student t distribution on df degrees of freedom.

Side Effects

The function rmt causes creation of the dataset .Random.seed if it does not already exist, otherwise its value is updated.

Background

The multivariate Student t distribution is a real valued symmetric distribution centered at mm. It is defined as the ratio of a centred multivariate normal distribution with covariance matrix cov, and the square root of an independent $\chi^2$ distribution with df degrees of freedom subsequently translated by mm. (See Johnson and Kotz, 1976, par. 37.3, pg. 134ff.) The multivariate t distribution approaches the multivariate Gaussian (Normal) distribution as the degrees of freedom go to infinity.
Note

Elements of x that are missing will cause the corresponding elements of the result to be missing.

References


See Also

TDist, Normal, Random.

Examples

dmt(c(0.1, -0.4), df = 4, mm = c(1, -1))
## density of a bivariate t distribution with 4 degrees of freedom
## and centered at (1,-1)

rmt(n = 100, df = 5, mult = 4)
## generates 100 replicates of a standard four-variate t distribution
## with 5 degrees of freedom

plot.Lapl

Plot uni- and bivariate approximate marginal densities

Description

Plots the uni- and bivariate approximations to the marginal densities of components of the MLE obtained by Laplace’s method.

Usage

## S3 method for class 'Lapl.spl'
plot(x, ...)

## S3 method for class 'Lapl.cont'
plot(x, ...)

Arguments

x an object of class Lapl.spl or Lapl.cont such as generated by the Laplace function.

... additional graphics parameters.

Details

This is a method for the function plot() for objects inheriting from class Lapl.spl and Lapl.cont generated by the Laplace() routine.
rsm.sample  Conditional Sampler for Regression-Scale Models

Description

Generates replicates of the MLEs of the parameters occurring in a regression-scale model using as reference distribution the conditional distribution of the MLEs given the value of the ancillary.

Usage

rsm.sample(data = stop("no data given"), R = 10000,
        ran.gen = stop("candidate distribution is missing, with no default"),
        trace = TRUE, step = 100, ...)

Arguments

data  A special conditional sampling data object. This object must be a list with the following elements:
anc  the vector containing the values of the ancillary; usually the Pearson residuals. It has to be of the same length than the number of observations in the linear regression model.
X  the model matrix. It may be obtained applying model.matrix to the fitted rsm object of interest. The number of observations has to be the same than the dimension of the ancillary, and the number of covariates must correspond to the number of regression coefficients defined in the coef component.
coef  the vector of true values of the regression coefficients, that is, the values used in the simulation study.
disp  the true value of the scale parameter used in the simulation study.
family  a family.rsm object characterizing the error distribution of the linear regression model. The following generator functions are available in the marg package of the R package bundlehoa: student (Student’s t), extreme (Gumbel or extreme value), logistic, logWeibull, logExponential, logRayleigh and Huber (Huber’s least favourable). The demonstration file ‘margdemo.R’ that accompanies the marg package shows how to create a new generator function.
fixed  a logical value. If TRUE the scale parameter is known.

The make.sample.data function can be used to create this data object from a fitted rsm model.

R  the number of replicates.

See Also

Laplace
ran.gen  a function which describes how the candidate values used in the Metropolis-
Hastings algorithm should be generated. It must be a function of at least two
arguments. The first one is the data object data, and the second argument is R,
the number of replicates required. Any other information needed may be passed
through the ... argument. The returned value should be a R times k matrix of
simulated values. For the value of k see the details section below.

trace  a logical value; if TRUE, the iteration number is printed. Defaults to TRUE.

step  a numerical value defining after how many iterations to print the iteration num-
ber. Default is 100.

...  absorbs additional arguments to ran.gen. These are passed unchanged each
time this function is called.

Details

The rsm.sample function uses the Metropolis-Hastings algorithm to generate an ergodic chain
with equilibrium distribution equal to the conditional distribution of the MLEs given the ancillary.
Because of the broad applicability of this algorithm the candidate generation density was not built
in, but has to be supplied by the user through the ran.gen argument. The output of this function
must be a R times k matrix, where k = p + 1 or k = p + 2 depending on whether the scale parameter
is fixed or not. The first p columns contain the MLEs of the regression coefficients, the following
the MLEs of the scale parameter if unknown, and the last column contains the probabilities of the
candidate values drawn from the candidate generation distribution. Note that these probabilities
need only be calculated up to a normalizing constant.

All information is supplied through the data argument. The user has to keep to the structure de-
scribed above. If a conditional simulation is to be performed for a fitted rsm object, the make.sample.data
function can be used to generate this special object. It is advisable to specify the logical switch
fixed in the conditional sampling object, although it needs not (in which case the scale parameter
is supposed to be unknown).

The conditional simulation (cs) object generated by rsm.sample contains all information necessary
for further investigation, such as the derivation of the conditional distribution of test statistics, the
calculation of conditional coverage levels of confidence intervals and many more. As the computa-
tion is somewhat tricky, an example is given in the demonstration file `csamplingdemo.R`.

Value

The returned value is an object of class cs containing the following components:

sim  a matrix with R rows each of which contains a sample from the conditional
distribution of the MLEs.

rho  the acceptance probabilities at each Metropolis-Hastings step, that is, the prob-
abilities with which the candidate values drawn from the candidate generation
distribution are accepted.

seed  the value of .Random.seed when rsm.sample was called.

data  the data as passed to rsm.sample.

R  the value of R as passed to rsm.sample.

call  the original call to rsm.sample.
Side Effects

The function `rsm.sample` causes creation of the dataset `.Random.seed` if it does not already exist, otherwise its value is updated.

Demonstration

The file `csamplingdemo.R` contains code that can be used to run a conditional simulation study similar to the one described in Brazzale (2000, Section 7.3) using the data given in Example 3 of DiCiccio, Field and Fraser (1990).

References


See Also

`make.sample.data`, `rsm.object`, `family.rsm.object`, `rsm`
Index

* datagen
  make.sample.data, 4

* distribution
  Laplace, 2
  mt, 6
  rsm.sample, 8

* hplot
  plot.Lapl, 7

* methods
  plot.Lapl, 7

* package
  csampling-package, 2

* regression
  make.sample.data, 4
  rsm.sample, 8
  .Random.seed, 6, 10

csampling (csampling-package), 2
csampling-package, 2
dmt (mt), 6

family.rsm, 3, 5, 8
family.rsm.object, 4, 10

Laplace, 2, 7, 8

make.sample.data, 3, 4, 4, 5, 8–10
model.matrix, 3, 5, 8
mt, 6

Normal, 6, 7

plot.Lapl, 7

Random, 7
rmt (mt), 6
rsm, 10
rsm.object, 5, 10
rsm.sample, 4, 5, 8

TDist, 7