

# Package ‘csampling’

July 2, 2014

**Version** 1.2-2

**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

**URL** <http://www.r-project.org>, <http://statwww.epfl.ch/AA/>

**LazyLoad** yes

**LazyData** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-04-03 14:00:47

## R topics documented:

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

<b>Index</b>	<b>11</b>
--------------	-----------

---

csampling-package      *Functions for Conditional Simulation in Regression-Scale Models*

---

### 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)  
 URL: <http://www.r-project.org>, <http://statwww.epfl.ch/AA/>  
 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](mailto:alessandra.brazzale@unimore.it)>. R port by Alessandra R. Brazzale <[alessandra.brazzale@unimore.it](mailto:alessandra.brazzale@unimore.it)>.

Maintainer: Alessandra R. Brazzale <[alessandra.brazzale@unimore.it](mailto: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.

**Usage**

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

**Arguments**

<code>which</code>	the kind of marginal density that should be approximated. Possible choices are <code>c</code> (univariate: regression coefficient), <code>s</code> (univariate: scale parameter), <code>cc</code> (bivariate: two regression coefficients) and <code>cs</code> (bivariate: regression coefficient and scale parameter).
<code>data</code>	a special <i>conditional sampling</i> data object. This object must be a list with the following elements: <ul style="list-style-type: none"> <li><code>anc</code> 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.</li> <li><code>X</code> the model matrix. It may be obtained applying <code>model.matrix</code> to the fitted <code>rsm</code> 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 <code>coef</code> component.</li> <li><code>coef</code> the vector of <i>true</i> values of the regression coefficients, that is, the values used in the simulation study.</li> <li><code>disp</code> the <i>true</i> value of the scale parameter used in the simulation study.</li> <li><code>family</code> a <code>family.rsm</code> object characterizing the error distribution of the linear regression model. The following generator functions are available in the <code>marg</code> package of the R package bundle <code>hoa</code>: <code>student</code> (Student's <i>t</i>), <code>extreme</code> (Gumbel or extreme value), <code>logistic</code>, <code>logWeibull</code>, <code>logExponential</code>, <code>logRayleigh</code> and <code>Huber</code> (Huber's least favourable). The demonstration file '<code>margdemo.R</code>' that accompanies the <code>marg</code> package shows how to create a new generator function.</li> <li><code>fixed</code> a logical value. If <code>TRUE</code> the scale parameter is known.</li> </ul> <p>The <code>make.sample.data</code> function can be used to create this data object from a fitted <code>rsm</code> model.</p>
<code>val1</code>	sequence of values for the first MLE at which to calculate the density.
<code>idx1</code>	index of the first regression coefficient, that is, its position in the vector MLE.
<code>val2</code>	sequence of values for the second MLE at which to calculate the density.
<code>idx2</code>	index of the second regression coefficient, that is, its position in the vector MLE.
<code>log.scale</code>	logical value. If <code>TRUE</code> the approximation is calculated on the log scale. Highly recommended. The default is <code>TRUE</code> .

**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.sp1` 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

Brazzale, A. R. (2000) *Practical Small-Sample Parametric Inference*. Ph.D. Thesis N. 2230, Department of Mathematics, Swiss Federal Institute of Technology Lausanne.

DiCiccio, T. J., Field, C. A. and Fraser, D. A. S. (1990) Approximations of marginal tail probabilities and inference for scalar parameters. *Biometrika*, **77**, 77–95.

### See Also

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

---

<code>make.sample.data</code>	<i>Create a Conditional Sampling Data Object</i>
-------------------------------	--

---

### Description

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

### Usage

```
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:

<code>anc</code>	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.
<code>X</code>	the model matrix. It may be obtained applying <code>model.matrix</code> to the fitted <code>rsm</code> 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 <code>coef</code> component.
<code>coef</code>	the vector of <i>true</i> values of the regression coefficients, that is, the values used in the simulation study.
<code>disp</code>	the <i>true</i> value of the scale parameter used in the simulation study.
<code>family</code>	a <code>family.rsm</code> object characterizing the error distribution of the linear regression model. The following generator functions are available in the <code>marg</code> package of the R package bundle <code>hoa</code> : <code>student</code> (Student's <i>t</i> ), <code>extreme</code> (Gumbel or extreme value), <code>logistic</code> , <code>logWeibull</code> , <code>logExponential</code> , <code>logRayleigh</code> and <code>Huber</code> (Huber's least favourable). The demonstration file ' <code>margdemo.R</code> ' that accompanies the <code>marg</code> package shows how to create a new generator function.
<code>fixed</code>	a logical value. If <code>TRUE</code> 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**

Brazzale, A. R. (2000) *Practical Small-Sample Parametric Inference*. Ph.D. Thesis N. 2230, Department of Mathematics, Swiss Federal Institute of Technology Lausanne.

DiCiccio, T. J., Field, C. A. and Fraser, D. A. S. (1990) Approximations of marginal tail probabilities and inference for scalar parameters. *Biometrika*, **77**, 77–95.

**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**

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

Johnson, N. L. and Kotz, S. (1976) *Distributions in Statistics: Continuous Multivariate Distributions*. New York: Wiley.

**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.

**See Also**[Laplace](#)

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 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.

**R** the number of replicates.



<code>ran.gen</code>	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 <code>data</code> , and the second argument is <code>R</code> , the number of replicates required. Any other information needed may be passed through the <code>...</code> argument. The returned value should be a <code>R</code> times <code>k</code> matrix of simulated values. For the value of <code>k</code> see the details section below.
<code>trace</code>	a logical value; if <code>TRUE</code> , the iteration number is printed. Defaults to <code>TRUE</code> .
<code>step</code>	a numerical value defining after how many iterations to print the iteration number. Default is 100.
<code>...</code>	absorbs additional arguments to <code>ran.gen</code> . 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 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. 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 computation 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:

<code>sim</code>	a matrix with <code>R</code> rows each of which contains a sample from the conditional distribution of the MLEs.
<code>rho</code>	the acceptance probabilities at each Metropolis-Hastings step, that is, the probabilities with which the candidate values drawn from the candidate generation distribution are accepted.
<code>seed</code>	the value of <code>.Random.seed</code> when <code>rsm.sample</code> was called.
<code>data</code>	the data as passed to <code>rsm.sample</code> .
<code>R</code>	the value of <code>R</code> as passed to <code>rsm.sample</code> .
<code>call</code>	the original call to <code>rsm.sample</code> .

**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**

Brazzale, A. R. (2000) *Practical Small-Sample Parametric Inference*. Ph.D. Thesis N. 2230, Department of Mathematics, Swiss Federal Institute of Technology Lausanne.

DiCiccio, T. J., Field, C. A. and Fraser, D. A. S. (1990) Approximations of marginal tail probabilities and inference for scalar parameters. *Biometrika*, **77**, 77–95.

**See Also**

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

# Index

- \*Topic **datagen**
  - make.sample.data, 4
- \*Topic **distribution**
  - Laplace, 2
  - mt, 6
  - rsm.sample, 8
- \*Topic **hplot**
  - plot.Lapl, 7
- \*Topic **methods**
  - plot.Lapl, 7
- \*Topic **package**
  - csampling-package, 2
- \*Topic **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