

Package: L1pack (via r-universe)

August 21, 2024

Type Package

Title Routines for L1 Estimation

Version 0.41-245

Date 2024-04-23

Maintainer Felipe Osorio <felipe.osorios@usm.cl>

Description L1 estimation for linear regression using Barrodale and Roberts' method <doi:10.1145/355616.361024> and the EM algorithm <doi:10.1023/A:1020759012226>. Estimation of mean and covariance matrix using the multivariate Laplace distribution, density, distribution function, quantile function and random number generation for univariate and multivariate Laplace distribution <doi:10.1080/03610929808832115>.

Depends R(>= 3.5.0), fastmatrix

LinkingTo fastmatrix

Imports stats, grDevices, graphics

License GPL-3

URL <http://l1pack.mat.utfsm.cl/>

NeedsCompilation yes

LazyLoad yes

Repository <https://faosorios.r-universe.dev>

RemoteUrl <https://github.com/faosorios/l1pack>

RemoteRef HEAD

RemoteSha 60e126997bd33e942daab936c7106eccb5cabb8a

Contents

confint.lad	2
ereturns	3
l1fit	3
lad	5

lad.fit	6
lad.fit-methods	7
Laplace	8
LaplaceFit	9
mLaplace	11
simulate.lad	12
vcov.lad	13
WH.Laplace	14

Index	16
--------------	-----------

confint.lad	<i>Confidence intervals from lad models</i>
-------------	---

Description

Computes confidence intervals for one or more parameters in a fitted model associated to a lad object.

Usage

```
## S3 method for class 'lad'
confint(object, parm, level = 0.95, ...)
```

Arguments

object	a fitted model object.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
...	additional argument(s) for related methods.

Details

confint is a generic function. Confidence intervals associated to lad objects are asymptotic, and needs suitable `coef` and `vcov` methods to be available.

Value

A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as $(1-level)/2$ and $1 - (1-level)/2$ in % (by default 2.5% and 97.5%).

See Also

`confint.glm` and `confint.nls` in package **MASS**.

Examples

```
fm <- lad(stack.loss ~ ., data = stackloss, method = "BR")
confint(fm) # based on asymptotic normality
```

ereturns

*Excess returns for Martin Marietta and American Can companies***Description**

Data from the Martin Marietta and American Can companies collected over a period of 5 years on a monthly basis.

Usage

```
data(ereturns)
```

Format

A data frame with 60 observations on the following 4 variables.

Date the month in which the observations were collected.

am.can excess returns from the American Can company.

m.marietta excess returns from the Martin Marietta company.

CRSP an index for the excess rate returns for the New York stock exchange.

Source

Butler, R.J., McDonald, J.B., Nelson, R.D., and White, S.B. (1990). Robust and partially adaptive estimation of regression models. *The Review of Economics and Statistics* **72**, 321-327.

l1fit

*Minimum absolute residual (L1) regression***Description**

Performs an L1 regression on a matrix of explanatory variables and a vector of responses.

Usage

```
l1fit(x, y, intercept = TRUE, tolerance = 1e-07, print.it = TRUE)
```

Arguments

x	vector or matrix of explanatory variables. Each row corresponds to an observation and each column to a variable. The number of rows of x should equal the number of data values in y, and there should be fewer columns than rows. Missing values are not allowed.
y	numeric vector containing the response. Missing values are not allowed.
intercept	logical flag. If TRUE, an intercept term is included in the regression model.
tolerance	numerical value used to test for singularity in the regression.
print.it	logical flag. If TRUE, then warnings about non-unique solutions and rank deficiency are given.

Details

The Barrodale-Roberts algorithm, which is a specialized linear programming algorithm, is used.

Value

list defining the regression (compare with function `lsfit`).

coefficients	vector of coefficients.
residuals	residuals from the fit.
message	character strings stating whether a non-unique solution is possible, or if the x matrix was found to be rank deficient.

References

Barrodale, I., Roberts, F.D.K. (1973). An improved algorithm for discrete L1 linear approximations. *SIAM Journal of Numerical Analysis* **10**, 839-848.

Barrodale, I., Roberts, F.D.K. (1974). Solution of an overdetermined system of equations in the L1 norm. *Communications of the ACM* **17**, 319-320.

Bloomfield, P., Steiger, W.L. (1983). *Least Absolute Deviations: Theory, Applications, and Algorithms*. Birkhauser, Boston, Mass.

Examples

```
l1fit(stack.x, stack.loss)
```

lad *Least absolute deviations regression*

Description

This function is used to fit linear models considering Laplace errors.

Usage

```
lad(formula, data, subset, na.action, method = "BR", tol = 1e-7, maxiter = 200,  
    x = FALSE, y = FALSE, contrasts = NULL)
```

Arguments

formula	an object of class "formula": a symbolic description of the model to be fitted.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lad</code> is called.
subset	an optional expression indicating the subset of the rows of data that should be used in the fit.
na.action	a function that indicates what should happen when the data contain NAs.
method	character string specifying the fitting method to be used; the options are "BR" Barrodale and Roberts' method (the default) and "EM" for an EM algorithm using IRLS.
tol	the relative tolerance for the iterative algorithm. Default is <code>tol = 1e-7</code> .
maxiter	The maximum number of iterations for the EM method. Default to 200.
x, y	logicals. If TRUE the corresponding components of the fit (the model matrix, the response) are returned.
contrasts	an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .

Value

An object of class `lad` representing the linear model fit. Generic function `print`, show the results of the fit.

The functions `print` and `summary` are used to obtain and print a summary of the results. The generic accessor functions `coefficients`, `fitted.values` and `residuals` extract various useful features of the value returned by `lad`.

Author(s)

The design was inspired by the R function [lm](#).

References

Barrodale, I., Roberts, F.D.K. (1974). Solution of an overdetermined system of equations in the L1 norm. *Communications of the ACM* **17**, 319-320.

Phillips, R.F. (2002). Least absolute deviations estimation via the EM algorithm. *Statistics and Computing* **12**, 281-285.

Examples

```
fm <- lad(stack.loss ~ ., data = stackloss, method = "BR")
summary(fm)
```

lad.fit *Fitter functions for least absolute deviation (LAD) regression*

Description

This function is a *switcher* among various numerical fitting functions ([lad.fit.BR](#), and [lad.fit.EM](#)). The argument `method` does the switching: "BR" for [lad.fit.BR](#), etc. This should usually *not* be used directly unless by experienced users.

Usage

```
lad.fit(x, y, method = "BR", tol = 1e-7, maxiter = 200)
```

Arguments

<code>x</code>	design matrix of dimension $n \times p$.
<code>y</code>	vector of observations of length n .
<code>method</code>	currently, methods "BR" (default), and "EM" are supported.
<code>tol</code>	the relative tolerance for the iterative algorithm. Default is <code>tol = 1e-7</code> .
<code>maxiter</code>	The maximum number of iterations for the EM method. Default to 200.

Value

a [list](#) with components:

<code>coefficients</code>	a named vector of coefficients.
<code>scale</code>	final scale estimate of the random error.
<code>residuals</code>	the residuals, that is response minus fitted values.
<code>fitted.values</code>	the fitted values.
<code>SAD</code>	the sum of absolute deviations.
<code>weights</code>	estimated EM weights.
<code>basic</code>	basic observations, that is observations with zero residuals.
<code>logLik</code>	the log-likelihood at convergence.

See Also

[lad.fit.BR](#), [lad.fit.EM](#).

Examples

```
x <- cbind(1, stack.x)
fm <- lad.fit(x, stack.loss, method = "BR")
fm
```

lad.fit-methods *Fit a least absolute deviation (LAD) regression model*

Description

Fits a linear model using LAD methods, returning the bare minimum computations.

Usage

```
lad.fit.BR(x, y, tol = 1e-7)
lad.fit.EM(x, y, tol = 1e-7, maxiter = 200)
```

Arguments

<code>x, y</code>	numeric vectors or matrices for the predictors and the response in a linear model. Typically, but not necessarily, <code>x</code> will be constructed by one of the fitting functions.
<code>tol</code>	the relative tolerance for the iterative algorithm. Default is <code>tol = 1e-7</code> .
<code>maxiter</code>	The maximum number of iterations for the EM method. Default to 200.

Value

The bare bones of a `lad` object: the coefficients, residuals, fitted values, and some information used by `summary.lad`.

See Also

[lad](#), [lad.fit](#), [lm](#)

Examples

```
x <- cbind(1, stack.x)
z <- lad.fit.BR(x, stack.loss)
z
```

Laplace

*The symmetric Laplace distribution***Description**

Density, distribution function, quantile function and random generation for the Laplace distribution with location parameter μ and scale parameter ϕ .

Usage

```
dlaplace(x, location = 0, scale = 1, log = FALSE)
plaplace(q, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)
qlaplace(p, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)
rlaplace(n, location = 0, scale = 1)
```

Arguments

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations.
<code>location</code>	location parameter μ , which is the mean.
<code>scale</code>	scale parameter ϕ . Scale must be positive.
<code>log, log.p</code>	logical; if TRUE, probabilities p are given as $\log(p)$.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.

Details

If `location` or `scale` are not specified, they assume the default values of 0 and 1 respectively.

The Laplace distribution with location μ and scale ϕ has density

$$f(x) = \frac{1}{\sqrt{2}\phi} \exp(-\sqrt{2}|x - \mu|/\phi),$$

where $-\infty < x < \infty$, $-\infty < \mu < \infty$ and $\phi > 0$. The mean is μ and the variance is ϕ^2 .

The cumulative distribution function, assumes the form

$$F(x) = \begin{cases} \frac{1}{2} \exp(\sqrt{2}(x - \mu)/\phi) & x < \mu, \\ 1 - \frac{1}{2} \exp(-\sqrt{2}(x - \mu)/\phi) & x \geq \mu. \end{cases}$$

The quantile function, is given by

$$F^{-1}(p) = \begin{cases} \mu + \frac{\phi}{\sqrt{2}} \log(2p) & p < 0.5, \\ \mu - \frac{\phi}{\sqrt{2}} \log(2(1 - p)) & p \geq 0.5. \end{cases}$$

Value

dlaplace, plaplace, and qlaplace are respectively the density, distribution function and quantile function of the Laplace distribution. rlaplace generates random deviates drawn from the Laplace distribution, the length of the result is determined by n.

Author(s)

Felipe Osorio and Tymoteusz Wolodzko

References

Kotz, S., Kozubowski, T.J., Podgorski, K. (2001). *The Laplace Distributions and Generalizations*. Birkhauser, Boston.

Krishnamoorthy, K. (2006). *Handbook of Statistical Distributions with Applications*, 2nd Ed. Chapman & Hall, Boca Raton.

See Also

[Distributions](#) for other standard distributions and [rmLaplace](#) for the random generation from the multivariate Laplace distribution.

Examples

```
x <- rlaplace(1000)
## QQ-plot for Laplace data against true theoretical distribution:
qqplot(qlaplace(ppoints(1000)), x, main = "Laplace QQ-plot",
       xlab = "Theoretical quantiles", ylab = "Sample quantiles")
abline(c(0,1), col = "red", lwd = 2)
```

LaplaceFit

Estimation of mean and covariance using the multivariate Laplace distribution

Description

Estimates the mean vector and covariance matrix assuming the data came from a multivariate Laplace distribution.

Usage

```
LaplaceFit(x, data, subset, na.action, tol = 1e-6, maxiter = 200)
```

Arguments

x	a formula or a numeric matrix or an object that can be coerced to a numeric matrix.
data	an optional data frame (or similar: see model.frame), used only if x is a formula. By default the variables are taken from <code>environment(formula)</code> .
subset	an optional expression indicating the subset of the rows of data that should be used in the fitting process.
na.action	a function that indicates what should happen when the data contain NAs.
tol	the relative tolerance in the iterative algorithm.
maxiter	maximum number of iterations. The default is 200.

Value

A list with class 'LaplaceFit' containing the following components:

call	a list containing an image of the LaplaceFit call that produced the object.
center	final estimate of the location vector.
Scatter	final estimate of the scale matrix.
logLik	the log-likelihood at convergence.
numIter	the number of iterations used in the iterative algorithm.
weights	estimated weights corresponding to the Laplace distribution.
distances	estimated squared Mahalanobis distances.

Generic function `print` show the results of the fit.

References

Yavuz, F.G., Arslan, O. (2018). Linear mixed model with Laplace distribution (LLMM). *Statistical Papers* **59**, 271-289.

See Also

[cov](#)

Examples

```
fit <- LaplaceFit(stack.x)
fit

# covariance matrix
p <- fit$dim[2]
Sigma <- (4 * (p + 1)) * fit$Scatter
Sigma
```

mLaplace	<i>Multivariate Laplace distribution</i>
----------	--

Description

These functions provide the density and random number generation from the multivariate Laplace distribution.

Usage

```
dmLaplace(x, center = rep(0, nrow(Scatter)), Scatter = diag(length(center)), log = FALSE)
rmLaplace(n = 1, center = rep(0, nrow(Scatter)), Scatter = diag(length(center)))
```

Arguments

x	vector or matrix of data.
n	the number of samples requested.
center	a $k \times 1$ vector giving the locations.
Scatter	a $k \times k$ positive-definite dispersion matrix.
log	logical; if TRUE, the logarithm of the density function is returned.

Details

The multivariate Laplace distribution is a multidimensional extension of the univariate symmetric Laplace distribution. There are multiple forms of the multivariate Laplace distribution. Here, a particular case of the multivariate power exponential distribution introduced by Gomez et al. (1998) is considered.

The multivariate Laplace distribution with location $\boldsymbol{\mu} = \text{center}$ and $\boldsymbol{\Sigma} = \text{Scatter}$ has density

$$f(\mathbf{x}) = \frac{\Gamma(k/2)}{\pi^{k/2} \Gamma(k) 2^{k+1}} |\boldsymbol{\Sigma}|^{-1/2} \exp \left\{ -\frac{1}{2} [(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})]^{1/2} \right\}.$$

The function `rmLaplace` is an interface to C routines, which make calls to subroutines from LAPACK. The matrix decomposition is internally done using the Cholesky decomposition. If `Scatter` is not non-negative definite then there will be a warning message.

Value

If `x` is a matrix with n rows, then `dmLaplace` returns a $n \times 1$ vector considering each row of `x` as a copy from the multivariate Laplace.

If $n = 1$, then `rmLaplace` returns a vector of the same length as `center`, otherwise a matrix of n rows of random vectors.

References

Fang, K.T., Kotz, S., Ng, K.W. (1990). *Symmetric Multivariate and Related Distributions*. Chapman & Hall, London.

Gomez, E., Gomez-Villegas, M.A., Marin, J.M. (1998). A multivariate generalization of the power exponential family of distributions. *Communications in Statistics - Theory and Methods* **27**, 589-600.

Examples

```
# dispersion parameters
Scatter <- matrix(c(1,.5,.5,1), ncol = 2)
Scatter

# generate the sample
y <- rmLaplace(n = 2000, Scatter = Scatter)

# scatterplot of a random bivariate Laplace sample with center
# vector zero and scale matrix 'Scatter'
par(pty = "s")
plot(y, xlab = "", ylab = "")
title("bivariate Laplace sample", font.main = 1)
```

simulate.lad

Simulate responses from lad models

Description

Simulate one or more responses from the distribution corresponding to a fitted lad object.

Usage

```
## S3 method for class 'lad'
simulate(object, nsim = 1, seed = NULL, ...)
```

Arguments

object	an object representing a fitted model.
nsim	number of response vectors to simulate. Defaults to 1.
seed	an object specifying if and how the random number generator should be initialized ('seeded'). For the "lad" method, either NULL or an integer that will be used in a call to <code>set.seed</code> before simulating the response vectors. If set, the value is saved as the "seed" attribute of the returned value. The default, NULL will not change the random generator state, and return <code>.Random.seed</code> as the "seed" attribute, see 'Value'.
...	additional optional arguments.

Value

For the "lad" method, the result is a data frame with an attribute "seed". If argument seed is NULL, the attribute is the value of `.Random.seed` before the simulation was started.

Author(s)

Tymoteusz Wolodzko and Felipe Osorio

Examples

```
fm <- lad(stack.loss ~ ., data = stackloss)
sm <- simulate(fm, nsim = 4)
```

vcov.lad

Calculate variance-covariance matrix from lad models

Description

Returns the variance-covariance matrix of the main parameters of a fitted model for lad objects. The "main" parameters of model correspond to those returned by `coef`, and typically do not contain the nuisance scale parameter.

Usage

```
## S3 method for class 'lad'
vcov(object, ...)
```

Arguments

`object` an object representing a fitted model.
`...` additional arguments for method functions.

Value

A matrix of the estimated covariances between the parameter estimates in the linear regression model. This should have row and column names corresponding to the parameter names given by the `coef` method.

 WH.Laplace

 Wilson-Hilferty transformation

Description

Returns the Wilson-Hilferty transformation for multivariate Laplace deviates.

Usage

WH.Laplace(x, center, Scatter)

Arguments

x	object of class 'LaplaceFit' from which is extracted the estimated Mahalanobis distances of the fitted model. Also x can be a vector or matrix of data with, say, p columns.
center	mean vector of the distribution or data vector of length p . Not required if x have class 'LaplaceFit'.
Scatter	Scatter matrix (p by p) of the distribution. Not required if x have class 'LaplaceFit'.

Details

Let $T = D/(2p)$ be a Gamma distributed random variable, where D^2 denotes the squared Mahalanobis distance defined as

$$D^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}).$$

Thus, the Wilson-Hilferty transformation is given by

$$z = \frac{T^{1/3} - (1 - \frac{1}{9p})}{(\frac{1}{9p})^{1/2}}$$

and z is approximately distributed as a standard normal distribution. This is useful, for instance, in the construction of QQ-plots.

References

- Osorio, F., Galea, M., Henriquez, C., Arellano-Valle, R. (2023). Addressing non-normality in multivariate analysis using the t -distribution. *AStA Advances in Statistical Analysis* **107**, 785-813.
- Terrell, G.R. (2003). The Wilson-Hilferty transformation is locally saddlepoint. *Biometrika* **90**, 445-453.
- Wilson, E.B., Hilferty, M.M. (1931). The distribution of chi-square. *Proceedings of the National Academy of Sciences of the United States of America* **17**, 684-688.

Examples

```
Scatter <- matrix(c(1,.5,.5,1), ncol = 2)
Scatter

# generate the sample
y <- rmLaplace(n = 500, Scatter = Scatter)
fit <- LaplaceFit(y)
z <- WH.Laplace(fit)
par(pty = "s")
qqnorm(z, main = "Transformed distances QQ-plot")
abline(c(0,1), col = "red", lwd = 2)
```

Index

- * **datagen**
 - simulate.lad, 12
- * **datasets**
 - ereturns, 3
- * **distribution**
 - Laplace, 8
 - mLaplace, 11
- * **models**
 - confint.lad, 2
 - LaplaceFit, 9
 - simulate.lad, 12
 - vcov.lad, 13
- * **multivariate**
 - LaplaceFit, 9
 - mLaplace, 11
 - WH.Laplace, 14
- * **regression**
 - l1fit, 3
 - lad, 5
 - lad.fit, 6
 - lad.fit-methods, 7
 - .Random.seed, 12, 13
- coef, 2, 13
- confint.glm, 2
- confint.lad, 2
- confint.nls, 2
- cov, 10

- Distributions, 9
- dlaplace (Laplace), 8
- dmLaplace (mLaplace), 11

- ereturns, 3

- l1fit, 3
- lad, 5, 7
- lad.fit, 6, 7
- lad.fit-methods, 7
- lad.fit.BR, 6, 7
- lad.fit.BR (lad.fit-methods), 7
- lad.fit.EM, 6, 7
- lad.fit.EM (lad.fit-methods), 7
- Laplace, 8
- LaplaceFit, 9
- list, 6
- lm, 5, 7
- lsfit, 4

- mLaplace, 11
- model.frame, 10

- plaplace (Laplace), 8

- qlaplace (Laplace), 8

- rlaplace (Laplace), 8
- rmLaplace, 9
- rmLaplace (mLaplace), 11

- simulate.lad, 12

- vcov, 2
- vcov.lad, 13

- WH.Laplace, 14