Package 'L1pack'

October 19, 2022

Type Package

Title Routines for L1 Estimation

Version 0.41-2

Date 2022-10-18

Author Felipe Osorio [aut, cre] (<https://orcid.org/0000-0002-4675-5201>), Tymoteusz Wolodzko [aut]

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>, density, distribution function, quantile function and random number generation for univariate and multivariate Laplace distribution.

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 CRAN

Date/Publication 2022-10-19 07:17:52 UTC

R topics documented:

`confint.lad` *Confidence intervals from* `lad` *models*

Description

Computes confidence intervals for one or more parameters in a fitted model corresponding 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 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
```

`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`	vector of one or two character strings stating whether a non-unique solution is possible, or if the `x` matrix was found to be rank deficient.

References

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

Barrodale, I., and 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., and 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 = 500,
  model = TRUE, 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 `environment(formula)`, typically the environment from which `lad` 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 `tol = 1e-7`.
`maxiter`	The maximum number of iterations for the `EM` method. Default to 500.
`model, x, y`	logicals. If `TRUE` the corresponding components of the fit (the model frame, the model matrix, the response) are returned.
`contrasts`	an optional list. See the `contrasts.arg` of `model.matrix.default`.

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., and 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 = 500)
```

Arguments

`x`	design matrix of dimension $n \times p$.
`y`	vector of observations of length n.
`method`	currently, methods `"BR"` (default), and `"EM"` are supported.
`tol`	the relative tolerance for the iterative algorithm. Default is `tol = 1e-7`.
`maxiter`	The maximum number of iterations for the `EM` method. Default to 500.

Value

a `list` with components:

`coefficients`	a named vector of coefficients.
`scale`	final scale estimate of the random error.
`residuals`	the residuals, that is response minus fitted values.
`fitted.values`	the fitted values.
`SAD`	the sum of absolute deviations.
`weights`	estimated `EM` weights.
`basic`	basic observations, that is observations with zero residuals.
`logLik`	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 = 500)
```

Arguments

x, y	numeric vectors or matrices for the predictors and the response in a linear model. Typically, but not necessarily, x will be constructed by one of the fitting functions.
tol	the relative tolerance for the iterative algorithm. Default is tol = 1e-7.
maxiter	The maximum number of iterations for the EM method. Default to 500.

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 Laplace distribution*

Description

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

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

`x, q`	vector of quantiles.
`location, scale`	location and scale parameters. Scale must be positive.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

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

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

Value

`dlaplace`, `plaplace`, and `qlaplace` are respectively the density, distribution function and quantile function of the Laplace distribution. `rlaplace` generates random deviates from the Laplace.

The length of the result is determined by `n` for `rlaplace`, and is the maximum of the lengths of the numerical parameters for the other functions.

Author(s)

Felipe Osorio and Tymoteusz Wolodzko

References

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

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

See Also

Distributions for other standard distributions and `rmLaplace` for the random generation from the multivariate Laplace distribution.

Examples

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

`rmLaplace` *Multivariate Laplace random deviates*

Description

Random number generation from the multivariate Laplace distribution.

Usage

```
rmLaplace(n = 1, center = rep(0, nrow(Scatter)), Scatter = diag(length(center)))
```

Arguments

`n`	the number of samples requested
`center`	a vector giving the locations of each variable
`Scatter`	a positive-definite dispersion matrix

Details

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 `n = 1` a vector of the same length as `center`, otherwise a matrix of `n` rows of random vectors.

References

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

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

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 `set.seed` 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 `.Random.seed` 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 a 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.

Index