Package: diagL1 (via r-universe)

August 27, 2024

Type Package

Title Routines for Fit, Inference and Diagnostics in Linear L1 and LAD Models

Version 1.0.0

Author Kevin Allan Sales Rodrigues [aut, cre] (<https://orcid.org/0000-0003-4925-5883>), Silvia Nagib Elian [ctb, ths]

Maintainer Kevin Allan Sales Rodrigues <kevin.asr@outlook.com>

Description Diagnostics for linear L1 regression (also known as LAD - Least Absolute Deviations), including: estimation, confidence intervals, tests of hypotheses, measures of leverage, methods of diagnostics for L1 regression, special diagnostics graphs and measures of leverage. The algorithms are based in Dielman (2005) <doi:10.1080/00949650420002236 80>, Elian et al. (2000) <doi:10.1080/03610920008832518> and Dodge (1997) <doi:10.1006/jmva.1997.1666>. This package builds on the 'quantreg' package, which is a well-established package for tuning quantile regression models. There are also tests to verify if the errors have a Laplace distribution based on the work of Puig and Stephens (2000) <doi:10.2307/1270952>.

License GPL (>= 2)

Depends R (>= 3.5), quantreg (>= 5.97), greekLetters (>= 1.0.2)

Encoding UTF-8

LazyData true

Imports stats, methods, conquer, lawstat, MatrixModels, Matrix, MASS, cubature, doParallel, foreach

RoxygenNote 7.2.3

NeedsCompilation no

Date/Publication 2024-01-30 00:30:02 UTC

Repository https://kevin444.r-universe.dev

RemoteUrl https://github.com/cran/diagL1

RemoteRef HEAD

RemoteSha 9371787c4d41ae500a696a011f5c7f702e96aa51

Contents

`bile` *Bile Data*

Description

A dataset containing lithogenic bile concentrations in 29 Pima Indians. For 29 females, the data give age and percent saturation of bile cholesterol.

Usage

```
data(bile)
```

Format

A data frame with 29 rows and 2 variables.

Details

- Age: age in years of each female indian
- Concentration: percent saturation of bile cholesterol

References

Goodall, C. (1983). Examining Residuals. In Hoaglin, D. C., Mosteller, F. and Tukey, J. W. (1983). Understanding Robust and Exploratory Data Analysis. *Wiley series in probability and mathematical statistics*.

---

boot.regL1 *Bootstrapping Linear L1 Models*

Description

This function can be used to construct standard errors, confidence intervals and tests of hypotheses regarding linear L1 regression models. The bootstrap method used compute xypairs bootstrap for linear L1 regression.

Usage

```
boot.regL1(x, y, R = 1000)
```

Arguments

x	the regression design matrix.
y	the regression response vector.
R	the number of bootstrap replications.

Value

A list consisting of four elements: a vector of lambda mle estimate for each bootstrap sample and another vector with empirical quantiles. A matrix B of dimension R by p is returned with the R resampled estimates of the vector of L1 linear regression parameters. A matrix U of sampled indices.

See Also

boot.rq Bootstrapping Quantile Regression from package quantreg.

Examples

```
data(stackloss)
bt1 = boot.regL1(stack.x, stack.loss, 1000)
plot(bt1$lambda_mle)
```

`boot_het.regL1` *Bootstrapping Linear L1 Models*

Description

This function can be used to construct standard errors, confidence intervals and tests of hypotheses regarding linear L1 regression models. The bootstrap method used compute xypairs bootstrap for linear L1 regression.

Usage

```
boot_het.regL1(x, y, groups, R = 1000)
```

Arguments

`x`	the regression design matrix.
`y`	the regression response vector.
`groups`	vector with the group index associated with the observation, observations with the same index belong to the same group.
`R`	the number of bootstrap replications.

Value

A list consisting of four elements: a vector of lambda mle estimate for each bootstrap sample and another vector with empirical quantiles. A matrix B of dimension R by p is returned with the R resampled estimates of the vector of L1 linear regression parameters. A matrix U of sampled indices.

See Also

`boot.rq` Bootstrapping Quantile Regression from package `quantreg`.

Examples

```
set.seed(123)
x1 = matrix(rnorm(20), ncol = 2)
y1 = x1[, 1] + x1[, 2] + rlaplace(10, 0, 5)
x2 = matrix(rnorm(20), ncol = 2)
y2 = x2[, 1] + x2[, 2] + rlaplace(10, 0, 10)
x3 = matrix(rnorm(20), ncol = 2)
y3 = x3[, 1] + x3[, 2] + rlaplace(10, 0, 15)
x4 = matrix(rnorm(20), ncol = 2)
y4 = x4[, 1] + x4[, 2] + rlaplace(10, 0, 20)
x5 = matrix(rnorm(20), ncol = 2)
y5 = x5[, 1] + x5[, 2] + rlaplace(10, 0, 30)

y = c(y1, y2, y3, y4, y5)
x = rbind(x1, x2, x3, x4, x5)
group_index = c(rep(1,10),rep(2,10),rep(3,10),rep(4,10),rep(5,10))

bt1 = boot_het.regL1(x, y, group_index, 1000)
bt1$lambda_mle_het
```

---

class_to_regL1 *Change object class from "rq" to "regL1"*

Description

Changing the object's class from "rq" to "regL1" allows you to use functions from the 'diagL1' package normally.

Usage

```
class_to_regL1(object)
```

Arguments

object	Object from "rq" class.

Value

Object with class "regL1".

See Also

regL1 for fitting linear L1 models. rq for fitting linear L1 models.

class_to_rq *Change object class from "regL1" to "rq"*

Description

Changing the object's class from "regL1" to "rq" allows you to use functions from the 'quantreg' package normally.

Usage

```
class_to_rq(object)
```

Arguments

object	Object from "regL1" class.

Value

Object with class "rq".

See Also

regL1 for fitting linear L1 models. rq for fitting linear L1 models.

CookDistance *Calculate Cook Distance*

Description

Calculate Cook Distance

Usage

```
CookDistance(model)
```

Arguments

model	Object returned from regL1 representing the fit of the L1 model.

Value

Cook Distance	A vector with Cook Distance for each observation.

.

References

Sun, R.-B. and Wei, B.-C. (2004). On influence assessment for lad regression. *Statistics & Probability Letters*, **67**, 97-110. doi:10.1016/j.spl.2003.08.018.

Examples

```
set.seed(123)
x = matrix(rnorm(100), ncol = 2)
y = x[, 1] + x[, 2] + rlaplace(50, 0, 5)

# Fits a linear regression L1 model
mod1 = regL1(y ~ x)
CookDistance(mod1)
```

`dlaplace` *Probability density function (PDF) of the Laplace distribution*

Description

Probability density function (PDF) of the Laplace distribution

Usage

```
dlaplace(x, location = 0, scale = 1)
```

Arguments

`x`	Values for which to calculate the probability density.
`location`	Location parameter of the Laplace distribution (default = 0).
`scale`	Scale parameter of the Laplace distribution (default = 1).

Value

Vector of calculated probability densities.

Examples

```
dlaplace(0, 0, 1)
```

FD *F Distance*

Description

F Distance

Usage

```
FD(model, norm = 2)
```

Arguments

model	Object returned from regL1 representing the fit of the L1 model.
norm	Type of norm, there are two types, 1 (norm L1) and 2 (norm L2). The default is norm L2.

Value

F Distance	A vector with F Distance for each observation.

.

References

Sun, R.-B. and Wei, B.-C. (2004). On influence assessment for lad regression. *Statistics & Probability Letters*, **67**, 97-110. doi:10.1016/j.spl.2003.08.018.

Examples

```
set.seed(123)
x = matrix(rnorm(100), ncol = 2)
y = x[, 1] + x[, 2] + rlaplace(50, 0, 5)

# Fits a linear regression L1 model
mod1 = regL1(y ~ x)
FD(mod1)
```

Fire *Fire Data*

Description

A dataset containing incidence of fire data for 47 residential areas in Chicago for the year 1975.

Usage

```
data(Fire)
```

Format

A data frame with 47 rows and 6 variables.

Details

- Fire: fires per 1000 housing units
- log_Fire: log of variable Fire
- Theft: thefts per 1000 residents
- Age: percent of housing units built in or before 1940
- Income: median family income as a multiple of $1000

References

Birkes, D.and Dodge, Y. (1993). Alternative Methods of Regression. *Jonh Wiley & Sons*.

forwardSearch_regL1 *Forward Search in Linear L1 Models*

Description

This function applies the forward search approach to robust analysis in linear L1 models. This function is based on function `fwdlm` of package 'forward'.

Usage

```
forwardSearch_regL1(
  formula,
  data,
  nsamp = "best",
  intercept = TRUE,
  trace = TRUE,
  subset,
  weights,
```

```
  na.action,
  method = "br",
  model = TRUE,
  contrasts = NULL,
  ...
)
```

Arguments

formula	a formula object, with the response on the left of a ~ operator, and the terms, separated by + operators, on the right.
data	a data.frame in which to interpret the variables named in the formula, or in the subset and the weights argument. If this is missing, then the variables in the formula should be on the search list. This may also be a single number to handle some special cases – see below for details.
nsamp	the initial subset for the forward search in linear regression is found by fitting the regression model with the R function lmsreg. This argument allows to control how many subsets are used in the Least Median of Squares regression. The choices are: the number of samples or "best" (the default) or "exact" or "sample". For details see lmsreg.
intercept	logical for the inclusion of the intercept (if no formula is provided).
trace	logical, if TRUE a message is printed for every ten iterations completed during the forward search.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	vector of observation weights; if supplied, the algorithm fits to minimize the sum of the weights multiplied into the absolute residuals. The length of weights must be the same as the number of observations. The weights must be nonnegative and it is strongly recommended that they be strictly positive, since zero weights are ambiguous.
na.action	a function to filter missing data. This is applied to the model.frame after any subset argument has been used. The default (with na.fail) is to create an error if any missing values are found. A possible alternative is na.omit, which deletes observations that contain one or more missing values.
method	the algorithmic method used to compute the fit. There are several options: "br", "fn", "pfn", "sfn", "fnc", "conquer", "pfnb", "qfnb", "ppro" and "lasso". See rq for more details.
model	if TRUE then the model frame is returned. This is essential if one wants to call summary subsequently.
contrasts	a list giving contrasts for some or all of the factors default = NULL appearing in the model formula. The elements of the list should have the same name as the variable and should be either a contrast matrix (specifically, any full-rank matrix with as many rows as there are levels in the factor), or else a function to compute such a matrix given the number of levels.
...	additional arguments for the fitting routines (see rq.fit.br and rq.fit.fnb, etc. and the functions they call).

Value

A fitted forward search in linear L1 regression model object.

References

Atkinson, A.C. and Riani, M. (2000). *Robust Diagnostic Regression Analysis*. New York: Springer.

Examples

```
# applies the forward search approach to robust analysis in a linear L1 model
mod = forwardSearch_regL1(Concentration ~ Age, data = bile)
```

`HV_test` *Homogeneity of Variance Tests for Linear L1 Models*

Description

Homogeneity of Variance Tests for Linear L1 Models

Usage

```
HV_test(y, x, groups, alpha = 0.05, tolerance = 0.001, max_iteration = 2000)
```

Arguments

`y`	A vector with response variables.
`x`	A matrix with a single explanatory variable.
`groups`	Vector containing the group index to which the observation belongs.
`alpha`	Significance level of the test, must be between 0 and 1.
`tolerance`	threshold that determines when the iterative algorithm should stop (for `regL1_het`).
`max_iteration`	maximum number of iterations (for `regL1_het`).

Details

The 3 statistics to test homogeneity of variance are discussed in Rodrigues (2024), for more details see this reference. In practice, use the HV_LRT statistic results. If possible, use the HV_LRT statistic with a critical value obtained via bootstrap simulation.

Value

A list with results from 3 homogeneity of variance tests

`alpha` alpha argument.

`asymptotic_critical_value` asymptotic alpha-based test critical value.

`HV_LRT` LF1 statistic value using MLE (Maximum Likelihood Estimator).

`HV_max_lambda_ratio` p-value of LF1 statistic using MLE.

`HV_max_log_lambda_ratio` LF1 statistic value using ROS (Residuals Order Statistics).

`number_of_groups` number of groups.

`N` overall sample size.

`lambda_heteroscedastic` heteroscedastic estimation of the scaling parameters.

`lambda_homocedastic` homoscedastic estimation of the scale parameter.

References

Rodrigues, K. A. S. (2024). **Analysis of the adjustment of the L1 regression model**. Phd dissertation, University of São Paulo, BR.

Examples

```
set.seed(123)
x1 = matrix(rnorm(20), ncol = 1)
y1 = x1 + rlaplace(20, 0, 1)
x2 = matrix(rnorm(20), ncol = 1)
y2 = x2 + rlaplace(20, 0, 1.5)
x3 = matrix(rnorm(20), ncol = 1)
y3 = x3 + rlaplace(20, 0, 2)
x4 = matrix(rnorm(20), ncol = 1)
y4 = x4 + rlaplace(20, 0, 2.5)
x5 = matrix(rnorm(20), ncol = 1)
y5 = x5 + rlaplace(20, 0, 3)

y = c(y1, y2, y3, y4, y5)
x = rbind(x1, x2, x3, x4, x5)
group_index = c(rep(1,20),rep(2,20),rep(3,20),rep(4,20),rep(5,20))

# Application of the homogeneity of variance test
test_result = HV_test(y, x, group_index)
test_result
```

lambda_mle *Function to estimate lambda via MLE*

Description

Function to estimate lambda via MLE

Usage

```
lambda_mle(model)
```

Arguments

model	A linear model L1 fitted with the rq function from the quantreg package.

Value

lambda MLE (Maximum Likelihood Estimator) estimator.

Examples

```
data(stackloss)
model_L1 = regL1(stack.loss ~ stack.x)
lambda_mle(model_L1)
```

lambda_ros *Function to calculate robust lambda estimator*

Description

Function to compute a robust lambda estimator, which is based on the residuals order statistics.

Usage

```
lambda_ros(model)
```

Arguments

model	A linear model L1 fitted with the rq function from the quantreg package.

Value

Robust lambda estimator, which is based on the residuals order statistics.

Examples

```
data(stackloss)
model_L1 = regL1(stack.loss ~ stack.x)
lambda_ros(model_L1)
```

`laplace.dist.test` *Wrapper for the 'laplace.test' function from the 'lawstat' package*

Description

This function is a wrapper for the 'laplace.test' function from the 'lawstat' package. The advantage of using this function instead of the 'laplace.test' function of the 'lawstat' package is that this function shows the tables with the critical values of the statistics provided by Puig and Stephens (2000). This makes interpretation of the results easier.

Usage

```
laplace.dist.test(y, print_tables = TRUE)
```

Arguments

`y`	A numeric vector containing the sample data.
`print_tables`	A boolean variable that indicates whether tables with critical values will be printed or not.

Value

The result of the laplace.test function.

References

Puig, P. and Stephens, M. A. (2000). Tests of fit for the Laplace distribution, with applications. *Technometrics*, **42**(4), 417-424. doi:10.2307/1270952.

See Also

`laplace.test` Goodness-of-fit Test Statistics for the Laplace Distribution from package `lawstat`.

Examples

```
normal_sample = rnorm(100, 0, 10)
laplace_sample = rlaplace(100, 0, 10)
laplace.dist.test(normal_sample)
laplace.dist.test(laplace_sample)
```

LF_test *Lack of Fit Tests for Linear L1 Models*

Description

Lack of Fit Tests for Linear L1 Models

Usage

```
LF_test(y, x, groups, alpha = 0.05)
```

Arguments

y	A vector with response variables.
x	A matrix with a single explanatory variable.
groups	Vector containing the group index to which the observation belongs.
alpha	Significance level of the test, must be between 0 and 1.

Details

The 3 statistics to test lack of fit are discussed in Rodrigues (2024), for more details see this reference. In practice, use the LF1_MLE statistic results. These tests were developed with just one explanatory variable in mind, which is why we include an error if there is more than one explanatory variable.

Value

A list with results from 3 lack of fit tests

alpha	alpha argument.
critical_value	alpha-based test critical value.
LF1_MLE	LF1 statistic value using MLE (Maximum Likelihood Estimator).
LF1_MLE	p-value of LF1 statistic using MLE.
LF1_ROS	LF1 statistic value using ROS (Residuals Order Statistics).
LF2	LF2 statistic value.
modelo_H0	model fitted under H0.
modelo_Ha	model fitted under Ha.
MLE	estimation of the scale parameter of the estimator model via MLE.
ROS	estimation of the scale parameter of the estimator model via ROS.
SAE_H0	SAE (Sum of Absolute Errors) of the adjusted model under H0.
SAE_Ha	SAE (Sum of Absolute Errors) of the adjusted model under Ha.
matrix_mean_x	average of the explanatory variable per group of observations.
number_of_groups	number of groups.

References

Rodrigues, K. A. S. (2024). **Analysis of the adjustment of the L1 regression model**. Phd dissertation, University of São Paulo, BR.

Examples

```
set.seed(123)
x1 = matrix(rnorm(20), ncol = 1)
y1 = x1 + rlaplace(20, 0, 5)
x2 = matrix(rnorm(20), ncol = 1)
y2 = x2 + rlaplace(20, 1, 5)
x3 = matrix(rnorm(20), ncol = 1)
y3 = x3 + rlaplace(20, 2, 5)
x4 = matrix(rnorm(20), ncol = 1)
y4 = x4 + rlaplace(20, 3, 5)
x5 = matrix(rnorm(20), ncol = 1)
y5 = x5 + rlaplace(20, 4, 5)

y = c(y1, y2, y3, y4, y5)
x = rbind(x1, x2, x3, x4, x5)
group_index = c(rep(1,20),rep(2,20),rep(3,20),rep(4,20),rep(5,20))

# Application of the lack of fit test
test_result = LF_test(y, x, group_index)
test_result
```

---

likelihoodCD *Calculate Conditional Likelihood Displacement*

Description

Calculate Conditional Likelihood Displacement

Usage

```
likelihoodCD(model)
```

Arguments

model	Object returned from regL1 representing the fit of the L1 model.

Value

Likelihood Displacement
: A vector with Likelihood Displacement for each observation.

.

References

Elian, S. N., André, C. D. S. and Narula, S. C. (2000). Influence Measure for the $L1$ regression. *Communications in Statistics - Theory and Methods*, **29**(4), 837-849. doi:10.1080/03610920008832518.

Examples

```
set.seed(123)
x = matrix(rnorm(100), ncol = 2)
y = x[, 1] + x[, 2] + rlaplace(50, 0, 5)

# Fits a linear regression L1 model
mod1 = regL1(y ~ x)
likelihoodCD(mod1)
```

---

likelihoodD *Calculate Likelihood Displacement*

Description

Calculate Likelihood Displacement

Usage

```
likelihoodD(model)
```

Arguments

model	Object returned from regL1 representing the fit of the L1 model.

Value

Likelihood Displacement
: A vector with Likelihood Displacement for each observation.

.

References

Elian, S. N., André, C. D. S. and Narula, S. C. (2000). Influence Measure for the $L1$ regression. *Communications in Statistics - Theory and Methods*, **29**(4), 837-849. doi:10.1080/03610920008832518.

Examples

```
set.seed(123)
x = matrix(rnorm(100), ncol = 2)
y = x[, 1] + x[, 2] + rlaplace(50, 0, 5)

# Fits a linear regression L1 model
mod1 = regL1(y ~ x)
likelihoodD(mod1)
```

PIF *Predictive Influence Function*

Description

Predictive Influence Function

Usage

```
PIF(y, x, w, num_cores = 2)
```

Arguments

y	A vector with response variables.
x	A matrix with a single explanatory variable.
w	Explanatory variables vector for z, usually explanatory variables averages.
num_cores	Number of cores you want to use for parallel processing (default = 2).

Details

Please, install and load the "foreach" package to use this function. For more details see Rodrigues (2024).

Value

A vector with predictive influence function value for each observation.

References

Rodrigues, K. A. S. (2024). **Analysis of the adjustment of the L1 regression model**. Phd dissertation, University of São Paulo, BR.

plaplace *Cumulative distribution function (CDF) of the Laplace distribution*

Description

Cumulative distribution function (CDF) of the Laplace distribution

Usage

```
plaplace(q, location = 0, scale = 1)
```

Arguments

q	Values for which to calculate the cumulative distribution.
location	Location parameter of the Laplace distribution (default = 0).
scale	Scale parameter of the Laplace distribution (default = 1).

Value

Vector of calculated cumulative distributions.

Examples

```
plaplace(0, 0, 1)
```

---

plot.forwardSearch_regL1

Forward Search in Linear L1 Models

Description

This function plots the results of a forward search in linear L1 models.

Usage

```
## S3 method for class 'forwardSearch_regL1'
plot(
  x,
  type.plot = 1:5,
  squared = FALSE,
  scaled = FALSE,
  ylim = NULL,
  xlim = NULL,
  th.Res = 2,
  th.Lev = 0.25,
  sig.Tst = 2.58,
  labels.in.plot = TRUE,
  ...
)
```

Arguments

x	a "forwardSearch_regL1" object.
type.plot	select which plots to draw, by default all. Each graph is addressed by an integer: 1. scaled residuals 2. minimum deletion residuals 3. coefficients

4. statistics
5. MAE (Mean Absolute Error) values

squared	logical, if TRUE plots squared residuals.
scaled	logical, if TRUE plots scaled coefficient estimates.
ylim	a two component vector for the min and max of the y axis.
xlim	a two component vector for the min and max of the x axis.
th.Res	numerical, a threshold for labelling the residuals.
th.Lev	numerical, a threshold for labelling the leverages.
sig.Tst	numerical, a value (on the scale of the t statistics) used to draw the confidence interval on the plot of the t statistics.
labels.in.plot	logical, if TRUE units are labelled in the plots when required.
...	additional arguments.

Value

No return value, just plots the results of a forward search in linear L1 models.

See Also

forwardSearch_regL1 for apply forward search in linear L1 regression model.

Examples

```
# applies the forward search approach to robust analysis in a linear L1 model
mod = forwardSearch_regL1(Concentration ~ Age, data = bile)
plot(mod, 1)
```

---

print.forwardSearch_regL1

Print an forwardSearch_regL1 object

Description

Print an object generated by forwardSearch_regL1

Usage

```
## S3 method for class 'forwardSearch_regL1'
print(x, ...)
```

Arguments

x	Object returned from forwardSearch_regL1.
...	Optional arguments.

Value

No return value, called for side effects.

See Also

forwardSearch_regL1 for apply forward search in linear L1 regression model.

Examples

```
# applies the forward search approach to robust analysis in a linear L1 model
mod = forwardSearch_regL1(Concentration ~ Age, data = bile)
mod # or print(mod)
```

---

print.regL1 *Print an regL1 object*

Description

Print an object generated by regL1

Usage

```
## S3 method for class 'regL1'
print(x, ...)
```

Arguments

x	Object returned from regL1 representing the fit of the L1 model.
...	Optional arguments.

Value

No return value, called for side effects.

See Also

regL1 for fitting linear L1 models.

print.summary.forwardSearch_regL1 *Print Forward Search in Linear L1 Model Summary Object*

Description

Print summary of forward search in linear L1 model object.

Usage

```
## S3 method for class 'summary.forwardSearch_regL1'
print(x, digits = 4, ...)
```

Arguments

x	This is an object of class "summary" produced by a call to summary.regL1().
digits	Significant digits reported in the printed table.
...	Optional arguments.

Value

No return value, called for side effects.

See Also

forwardSearch_regL1 for apply forward search in linear L1 regression model.

Examples

```
# applies the forward search approach to robust analysis in a linear L1 model
mod = forwardSearch_regL1(Concentration ~ Age, data = bile)
summary(mod)
```

print.summary.regL1 *Print Linear L1 Regression Summary Object*

Description

Print summary of linear L1 regression object.

Usage

```
## S3 method for class 'summary.regL1'
print(x, digits = max(5, .Options$digits - 2), ...)
```

Arguments

x	This is an object of class "summary.regL1" produced by a call to summary.regL1().
digits	Significant digits reported in the printed table.
...	Optional arguments.

Value

No return value, called for side effects.

---

qlaplace *Quantile function (inverse of the CDF) of the Laplace distribution*

Description

Quantile function (inverse of the CDF) of the Laplace distribution

Usage

```
qlaplace(p, location = 0, scale = 1)
```

Arguments

p	Values for which to calculate quantiles.
location	Location parameter of the Laplace distribution (default = 0).
scale	Scale parameter of the Laplace distribution (default = 1).

Value

Vector of calculated quantiles.

Examples

```
qlaplace(0.5, 0, 1)
```

regL1 *Fitting Linear L1 Models*

Description

This function fits an L1 regression model using the rq function from the 'quantreg' package. L1 regression allows dealing with outliers and non-normal distributions in the data.

Usage

```
regL1(
  formula,
  data,
  subset,
  weights,
  na.action,
  method = "br",
  model = TRUE,
  contrasts = NULL,
  ...
)
```

Arguments

formula	a formula object, with the response on the left of a ~ operator, and the terms, separated by + operators, on the right.
data	a data.frame in which to interpret the variables named in the formula, or in the subset and the weights argument. If this is missing, then the variables in the formula should be on the search list. This may also be a single number to handle some special cases – see below for details.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	vector of observation weights; if supplied, the algorithm fits to minimize the sum of the weights multiplied into the absolute residuals. The length of weights must be the same as the number of observations. The weights must be nonnegative and it is strongly recommended that they be strictly positive, since zero weights are ambiguous.
na.action	a function to filter missing data. This is applied to the model.frame after any subset argument has been used. The default (with na.fail) is to create an error if any missing values are found. A possible alternative is na.omit, which deletes observations that contain one or more missing values.
method	the algorithmic method used to compute the fit. There are several options: "br", "fn", "pfn", "sfn", "fnc", "conquer", "pfnb", "qfnb", "ppro" and "lasso". See rq for more details.
model	if TRUE then the model frame is returned. This is essential if one wants to call summary subsequently.

contrasts	a list giving contrasts for some or all of the factors default = NULL appearing in the model formula. The elements of the list should have the same name as the variable and should be either a contrast matrix (specifically, any full-rank matrix with as many rows as there are levels in the factor), or else a function to compute such a matrix given the number of levels.
...	additional arguments for the fitting routines (see rq.fit.br and rq.fit.fnb, etc. and the functions they call).

Details

L1 regression is an important particular case of quantile regression, so this function inherits from the "rq" class of the quantreg package.

Value

A fitted L1 linear regression model object.

Examples

```
set.seed(123)
x = matrix(rnorm(100), ncol = 2)
y = x[, 1] + x[, 2] + rlaplace(50, 0, 5)

# Fits a linear regression L1 model
mod1 = regL1(y ~ x)
```

---

regL1_het *Fitting Heteroscedastic Linear L1 Models*

Description

This function fits an groupwise heteroscedastic L1 regression model using the rq function from the 'quantreg' package.

Usage

```
regL1_het(
  x,
  y,
  groups,
  na.action = stats::na.omit,
  method = "br",
  model = TRUE,
  tolerance = 0.001,
  max_iteration = 2000,
  ...
)
```

Arguments

x	the regression design matrix.
y	the regression response vector.
groups	vector with the group index associated with the observation, observations with the same index belong to the same group.
na.action	a function to filter missing data. This is applied to the model.frame after any subset argument has been used. The default (with na.fail) is to create an error if any missing values are found. A possible alternative is na.omit, which deletes observations that contain one or more missing values.
method	the algorithmic method used to compute the fit. There are several options: "br", "fn", "pfn", "sfn", "fnc", "conquer", "pfnb", "qfnb", "ppro" and "lasso". See rq for more details.
model	if TRUE then the model frame is returned. This is essential if one wants to call summary subsequently.
tolerance	threshold that determines when the iterative algorithm should stop.
max_iteration	maximum number of iterations.
...	additional arguments for the fitting routines (see rq.fit.br and rq.fit.fnb, etc. and the functions they call).

Details

L1 regression is an important particular case of quantile regression, so this function inherits from the "rq" class of the quantreg package.

Value

A fitted heteroscedastic L1 linear regression model object.

Examples

```
set.seed(123)
x1 = matrix(rnorm(20), ncol = 2)
y1 = x1[, 1] + x1[, 2] + rlaplace(10, 0, 5)
x2 = matrix(rnorm(20), ncol = 2)
y2 = x2[, 1] + x2[, 2] + rlaplace(10, 0, 10)
x3 = matrix(rnorm(20), ncol = 2)
y3 = x3[, 1] + x3[, 2] + rlaplace(10, 0, 15)
x4 = matrix(rnorm(20), ncol = 2)
y4 = x4[, 1] + x4[, 2] + rlaplace(10, 0, 20)
x5 = matrix(rnorm(20), ncol = 2)
y5 = x5[, 1] + x5[, 2] + rlaplace(10, 0, 30)

y = c(y1, y2, y3, y4, y5)
x = rbind(x1, x2, x3, x4, x5)
group_index = c(rep(1,10),rep(2,10),rep(3,10),rep(4,10),rep(5,10))
# Fits a heteroscedastic linear regression L1 model
mod1 = regL1_het(x, y, group_index)
```

rlaplace *Function to generate random values from the Laplace distribution*

Description

Function to generate random values from the Laplace distribution

Usage

```
rlaplace(n, location = 0, scale = 1)
```

Arguments

n	Number of values to generate.
location	Location parameter of the Laplace distribution (default = 0).
scale	Scale parameter of the Laplace distribution (default = 1).

Value

Vector of generated random values.

Examples

```
rlaplace(10, 0, 1)
```

SAE *Function to calculate SAE*

Description

Function to calculate SAE (Sum of Absolute Errors) of the adjusted model.

Usage

```
SAE(model)
```

Arguments

model	A linear model L1 fitted with the rq function from the quantreg package.

Value

SAE (Sum of Absolute Errors).

Examples

```
data(stackloss)
model_L1 = regL1(stack.loss ~ stack.x)
SAE(model_L1)
```

`summary.forwardSearch_regL1` *Summarizing Fit of Forward Search in Linear L1 Regression*

Description

Returns a summary list for a forward search in linear L1 regression fit.

Usage

```
## S3 method for class 'forwardSearch_regL1'
summary(object, steps = "auto", ...)
```

Arguments

`object`	Object returned from forwardSearch_regL1.
`steps`	the number of forward steps to show.
`...`	Optional arguments.

Value

No return value, called for side effects.

References

Atkinson, A.C. and Riani, M. (2000). *Robust Diagnostic Regression Analysis*. New York: Springer.

See Also

`forwardSearch_regL1` for apply forward search in linear L1 regression model.

Examples

```
# applies the forward search approach to robust analysis in a linear L1 model
mod = forwardSearch_regL1(Concentration ~ Age, data = bile)
summary(mod)
```

summary.regL1 *Summary methods for L1 Regression*

Description

Returns a summary list for a L1 regression fit. A null value will be returned if printing is invoked.

Usage

```
## S3 method for class 'regL1'
summary(
  object,
  se = NULL,
  covariance = FALSE,
  hs = TRUE,
  U = NULL,
  gamma = 0.7,
  ...
)
```

Arguments

object	Object returned from regL1 representing the fit of the L1 model.
se	specifies the method used to compute standard standard errors. There are currently seven available methods: "rank", "iid", "nid", "ker", "boot", "BLB", "conquer" and "extreme".
covariance	logical flag to indicate whether the full covariance matrix of the estimated parameters should be returned.
hs	Use Hall Sheather bandwidth for sparsity estimation If false revert to Bofinger bandwidth.
U	Resampling indices or gradient evaluations used for bootstrap, see summary.rq and boot.rq for more details.
gamma	parameter controlling the effective sample size of the'bag of little bootstrap samples that will be b = n^gamma where n is the sample size of the original model.
...	Optional arguments. See summary.rq for more details.

Value

No return value, called for side effects.

See Also

regL1 for fitting linear L1 models. summary.rq summary methods for Quantile Regression.

Index