

Package ‘ADVICE’

July 21, 2025

Type Package

Title Automatic Direct Variable Selection via Interrupted Coefficient Estimation

Version 1.0

Author L. Tazik [aut, cre],
W.J. Braun [aut]

Maintainer L. Tazik <ladan.tazik@ubc.ca>

Description

Accurate point and interval estimation methods for multiple linear regression coefficients, under classical normal and independent error assumptions, taking into account variable selection.

LazyData true

LazyLoad true

Depends R (>= 2.0.1)

ZipData no

License Unlimited

NeedsCompilation no

RoxygenNote 7.1.1

Repository CRAN

Date/Publication 2021-04-16 08:50:06 UTC

Contents

confint.QRS	2
hello	3
ices	3
MSDecision	5
plot.QRS	6
QRS	7
rmultreg	8
summary.QRS	9

Index	11
--------------	-----------

confint.QRS	<i>Confidence Interval Function</i>
-------------	-------------------------------------

Description

Computes confidence intervals for one or more parameters in a fitted model. There is a default and a method for objects inheriting from class "qrs".

Usage

```
## S3 method for class 'QRS'
confint(object, parm, level, ...)
```

Arguments

object	a fitted model object from the QRS class.
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	a numeric value specifying the required confidence level.
...	additional argument(s) for the methods.

Details

This function computes t-based confidence intervals using $n-p$ degrees of freedom, where n is the number of observations and p is the number of regression coefficients in the full model.

Value

A 2-column matrix giving lower and upper confidence limits (corresponding to the given level) for each parameter. These will be labelled as $(1-level)/2$ and $1 - (1-level)/2$ in

Author(s)

Ladan Tazik, W.J. Braun

See Also

ices.R

Examples

```
myRegressionData <- rmultreg(100, k=20, p=.1, sdnoise = 1)
pairs(myRegressionData$data)
out <- ices(y ~ ., data = myRegressionData$data) # fit model to simulated data
confint(out) # calculate 95% confidence intervals for all coefficients
myRegressionData$coefficients # compare with true coefficients
```

hello	<i>Hello, World!</i>
-------	----------------------

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

ices	<i>Interrupted Coefficient Estimation Selection</i>
------	---

Description

This function provides an alternative multiple regression fitting procedure which simultaneously estimates and selects variables. The resulting coefficient estimates will tend to be slightly biased, but in a sparse setting, they can be quite accurate. A full regression model is specified by the user, and the function usually returns coefficient estimates for a reduced model, i.e., a model for which some of the coefficient estimates are exactly 0.

Usage

```
ices(formula, data, model = TRUE, x = FALSE, y = FALSE, qr = TRUE)
```

Arguments

formula	a formula object specifying the full regression model.
data	a data frame containing observations on the response variable and the predictor variables.
model, x, y, qr	logicals. If TRUE the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.

Value

a QRS class object

<code>coefficients</code>	a named numeric vector of coefficients
<code>residuals</code>	a numeric vector containing the response minus the fitted values.
<code>effects</code>	a numeric vector of containing the projections of the response variable under the orthogonal Q matrix coming from the QR decomposition of the model matrix.
<code>rank</code>	the numeric rank of the fitted linear model.
<code>fitted.values</code>	the estimated response values according to the fitted interrupted coefficient estimation selection regression model.
<code>sigma2</code>	the estimated noise variance based on the n-p residual effects, where p is the size of the full model.
<code>std_error</code>	a numeric vector of standard errors.
<code>df.residual</code>	residual degrees of freedom.
<code>x</code>	a numeric matrix containing the model matrix.
<code>y</code>	a numeric vector containing the response variable values.
<code>qr</code>	the QR decomposition object coming from the model matrix (after re-ordering columns).
<code>coefOrder</code>	permutation of the sequence 1:p which gives the ascending order of the coefficients of the linear model object, as a result of the pre-screening.
<code>call</code>	the matched call.
<code>terms</code>	the terms object used.
<code>names</code>	a character vector containing the column names of the model matrix.
<code>model</code>	if requested (the default), the model frame used in the case of the full regression model.

Author(s)

Ladan Tazik, W.J. Braun

See Also

`lm.R`, `QRS.R`

Examples

```
myRegressionData <- rmultreg(50, k=10, p=.25, sdnoise = .5)
pairs(myRegressionData$data)
out <- ices(y ~ ., data = myRegressionData$data) # fit model to simulated data
confint(out) # calculate 95 % confidence intervals for all coefficients
myRegressionData$coefficients # compare with true coefficients
```

MSDecision

Multiple Sclerosis Decision Delay

Description

This data frame contains the time (in weeks) between the initial symptoms (onset symptoms) and the decision time to visit a doctor in the case of 54 patients who eventually were diagnosed with multiple sclerosis. Interest centers on whether there are any factors which tend to be related to the delay time.

Usage

```
data(MSDecision)
```

Format

A data frame with 54 observations on the following 16 variables.

Delay numeric, time in weeks

ClinicalDiseaseCourse factor, 2 levels

CodedGender factor, 2 levels, 1 = Male, 2 = Female

AgeAtOnset numeric, age in years

OnsetSymptom1 factor, 4 levels

OnsetSymptom2 factor, 5 levels

OnsetSymptomSeverity factor, 2 levels, 0 = Low, 1 = High

TriggerSymptom1 factor, 4 levels

TriggerSymptom2 factor, 4 levels

TriggerSymptomSeverity factor, 2 levels, 0 = Low, 1 = High

FamilyHistory factor, 2 levels, yes = there is MS in the family history

FearOfWorseningSymptoms factor, 2 levels

MoreThanOneSymptom factor, 2 levels

EffectonResponsibilities factor, 2 levels, yes = the symptoms are having an effect on the individual

UncertainResponse logical, TRUE = recorded delay time is not accurate

Details

The levels of the Clinical Disease Course variable are: Clinically Isolated Syndrome and Relapse-Remitting.

Examples

```
xy <- MSDecision
xy$sensoryOnset1 <- factor(xy$OnsetSymptom1=="SENSORY")
xy$brainstemOnset2 <- factor(xy$OnsetSymptom2=="BRAINSTEM")
xy$sensoryTrigger1 <- factor(xy$TriggerSymptom1=="SENSORY")
xy$brainstemTrigger2 <- factor(xy$TriggerSymptom2=="BRAINSTEM")
xy <- xy[, -c(5, 6, 8, 9, 15)]
xy[,1]<-log(xy[,1])
names(xy)[1] <- "y"
out <- ices(y ~ ., data = xy)
summary(out)
plot(out)
plot(out, normqq=TRUE)
plot(out, scaleloc=TRUE)
```

plot.QRS

plot.QRS

Description

By default, this function plots residuals from the interrupted coefficient estimation selection model versus the corresponding fitted values. Alternatively, options to obtain a normal QQ plot or a scale-location plot of the residuals are also available.

Usage

```
## S3 method for class 'QRS'
plot(x, normqq = FALSE, scaleloc = FALSE, ...)
```

Arguments

<code>x</code>	an object of QRS class
<code>normqq</code>	a logical value, if TRUE, a normal QQ plot of the residuals will be plotted.
<code>scaleloc</code>	a logical value, if TRUE, a scale-location plot of the residuals will be plotted.
<code>...</code>	arguments to be passed to plot methods, such as graphical parameters (see par).

Value

No return value

Author(s)

Ladan Tazik, W.J. Braun

See Also

`plot.lm`

Description

Given a design matrix and a response variable, create a list which has the fitted model, estimated regression coefficients and standard error based on interrupted coefficient estimation selection.

Usage

```
QRS(x, y, Nsims)
```

Arguments

x	a numeric matrix; usually the model matrix for a multiple regression model.
y	a numeric vector; usually the values of the response variable in the regression model.
Nsims	number of simulation runs required for estimating the regression coefficient standard errors.

Details

The interrupted coefficient estimation selection procedure begins with consideration of a full model whereby a regression model with p terms is fit to n observations on a response and $p-1$ predictor variables. The variables are pre-screened by application of `lm` in order to cast the columns of the model matrix in increasing order of the p -values observed for the corresponding regression coefficients. The estimation then proceeds by the usual QR decomposition of the model matrix but is interrupted at the effects stage. The effects are classified as "different from 0" or "not different from 0", according to what is essentially a control chart procedure. The effects that are "not different from 0" are replaced with true 0's and the nonzero effects are left alone. The estimation is completed by backward-substitution solution of the zero and nonzero effects using the upper triangular matrix from the QR decomposition. The result is a set of coefficient estimates that will tend to be more accurate in a mean-squared-error sense than the original `lm` coefficient estimates, especially when some or all of the regression coefficients are 0. Coefficient standard error estimates are obtained by a parametric bootstrap procedure. This method is not recommended for strongly non-normal data, or where there is substantial multicollinearity.

Value

a QRS class object	
coefficients	a named numeric vector of coefficients
residuals	a numeric vector containing the response minus the fitted values.
effects	a numeric vector of containing the projections of the response variable under the orthogonal Q matrix coming from the QR decomposition of the model matrix.
rank	the numeric rank of the fitted linear model.

fitted.values	the estimated response values according to the fitted interrupted coefficient estimation selection regression model.
sigma2	the estimated noise variance based on the n-p residual effects, where p is the size of the full model.
std_error	a numeric vector of standard errors.
qr	the QR decomposition object coming from the model matrix (after re-ordering columns).
df.residual	the residual degrees of freedom.
model	if requested (the default), the model frame used.
x	a numeric matrix containing the model matrix.
y	a numeric vector containing the response variable values.
coefOrder	A permutation of the sequence 1:p which gives the ascending order of the coefficients of the linear model object, as a result of the pre-screening.
names	a character vector containing the column names of the model matrix.

Author(s)

Ladan Tazik, W.J. Braun

See Also

ices.R, lm.R

 rmultreg

Multiple Regression Data Generator

Description

Values of any number of predictor variables and a single response variable are simulated according to a model with randomly generated coefficients. Values of each predictor are simulated independently from standard normal distributions. The regression coefficients are generated independently from a uniform distribution on the interval (minimum, maximum), and each coefficient is multiplied by a Bernoulli (p) variate, independent of the other coefficients. This results in some of the coefficients being zeroed out. Noise is added to the regression response according to independent t variates with degrees of freedom equal to dfnoise.

Usage

```
rmultreg(n, k = 1, minimum = 0, maximum = 1, p = 0.5, dfnoise = 100, sdnoise = 1)
```


Arguments

n	number of observations.
k	number of predictor variables in addition to the intercept.
minimum	minimum possible value for the regression coefficients, apart, possibly, from some zeroes.
maximum	maximum possible value for the regression coefficients, apart, possibly, from some zeroes.
p	probability that a given regression coefficient remains nonzero.
dfnoise	degrees of freedom for t-distributed additive noise.
sdnoise	standard deviation of the noise term.

Value

a list containing	
data	a dataframe containing n observations on k predictor variables and a response y.
coefficients	a numeric vector containing the true regression coefficients.

Author(s)

W.J. Braun

Examples

```
myRegressionData <- rmultreg(50, k=3, p=.5, sdnoise = .25)
pairs(myRegressionData$data)
out <- ices(y ~ ., data = myRegressionData$data) # fit model to simulated data
confint(out) # calculate 95% confidence intervals for all coefficients
myRegressionData$coefficients # compare with true coefficients
```

summary.QRS

summary.QRS

Description

summary method for class "qrs"

Usage

```
## S3 method for class 'QRS'
summary(object, ...)
```

Arguments

object	an object of class "qrs"
...	additional arguments affecting the summary produced.

Value

The function computes and returns a list of summary statistics of the fitted linear model given in the QRS object.

Residuals	the weighted residuals, the usual residuals rescaled by the square root of the weights specified in the call to qrs
Coefficients	a p x 4 matrix with columns for the estimated coefficient, its standard error, z-score and corresponding (two-sided) probabilities
df	degrees of freedom
residualStandardError	Residual standard error

Author(s)

Ladan Tazik, W.J.Braun

See Also

QRS.R

Examples

```
myRegressionData <- rmultreg(25, k=5, p=.15, sdnoise = .25)
pairs(myRegressionData$data)
out <- ices(y ~ ., data = myRegressionData$data) # fit model to simulated data
summary(out) # estimates and standard errors for all coefficients
myRegressionData$coefficients # compare with true coefficients
```

Index

- * **datagen**
 - rmultreg, [8](#)
- * **datasets**
 - MSDecision, [5](#)
- * **graphics**
 - plot.QRS, [6](#)
- * **models**
 - confint.QRS, [2](#)
 - ices, [3](#)
 - QRS, [7](#)
 - summary.QRS, [9](#)
- confint.QRS, [2](#)
- hello, [3](#)
- ices, [3](#)
- MSDecision, [5](#)
- plot.QRS, [6](#)
- QRS, [7](#)
- rmultreg, [8](#)
- summary.QRS, [9](#)