Package ‘lmSubsets’

February 7, 2021

Version 0.5-2
Date 2021-02-03
Title Exact Variable-Subset Selection in Linear Regression
Description Exact and approximation algorithms for variable-subset selection in ordinary linear regression models. Either compute all submodels with the lowest residual sum of squares, or determine the single-best submodel according to a pre-determined statistical criterion. Hofmann et al. (2020) <doi:10.18637/jss.v093.i03>.
Depends R (>= 3.5.0)
SystemRequirements C++11
Imports stats, graphics, utils
License GPL (>= 3)
NeedsCompilation yes
Author Marc Hofmann [aut, cre],
          Cristian Gatu [aut],
          Erricos J. Kontoghiorghes [aut],
          Ana Colubi [aut],
          Achim Zeileis [aut] (https://orcid.org/0000-0003-0918-3766),
          Martin Moene [cph] (for the GSL Lite library),
          Microsoft Corporation [cph] (for the GSL Lite library),
          Free Software Foundation, Inc. [cph] (for snippets from the GNU ISO C++ Library)
Maintainer Marc Hofmann <marc.hofmann@gmail.com>
Repository CRAN
Date/Publication 2021-02-07 20:10:13 UTC

R topics documented:

   lmSubsets-package .................................................. 2
   AIC.lmSubsets ...................................................... 3
lmSubsets-package

Package lmSubsets

Description

Variable-subset selection in ordinary linear regression.

Author(s)

- Marc Hofmann (<marc.hofmann@gmail.com>)
- Cristian Gatu (<cgatu@info.uaic.ro>)
- Erricos J. Kontoghiorghes (<erricos@dcs.bbk.ac.uk>)
- Ana Colubi (<ana.colubi@gmail.com>)
- Achim Zeileis (<Achim.Zeileis@R-project.org>)
AIC.lmSubsets

References


See Also

Home page: [https://github.com/marc-hofmann/lmSubsets.R](https://github.com/marc-hofmann/lmSubsets.R)

AIC.lmSubsets Extract AIC values from a subset regression

Description

Evaluate Akaike's information criterion (AIC) for the specified submodels.

Usage

## S3 method for class 'lmSubsets'
AIC(object, size, best = 1, ..., k = 2, na.rm = TRUE, drop = TRUE)

## S3 method for class 'lmSelect'
AIC(object, best = 1, ..., k = 2, na.rm = TRUE, drop = TRUE)

Arguments

- object: "lmSubsets","lmSelect"—a subset regression
- size: integer[]—the submodel sizes
- best: integer[]—the submodel positions
- ...: ignored
- k: double—the penalty per model parameter
- na.rm: logical—if TRUE, remove NA entries
- drop: logical—if TRUE, simplify structure

Value

double[]—the AIC values
See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `AIC()` for the S3 generic

---

### Description

Data relating air pollution and mortality, frequently used for illustrations in ridge regression and related tasks.

### Usage

```r
data(AirPollution)
```

### Format

A data frame containing 60 observations on 16 variables.

- **precipitation**: average annual precipitation in inches
- **temperature1**: average January temperature in degrees Fahrenheit
- **temperature7**: average July temperature in degrees Fahrenheit
- **age**: percentage of 1960 SMSA population aged 65 or older
- **household**: average household size
- **education**: median school years completed by those over 22
- **housing**: percentage of housing units which are sound and with all facilities
- **population**: population per square mile in urbanized areas, 1960
- **noncauc**: percentage of non-Caucasian population in urbanized areas, 1960
- **whitecollar**: percentage employed in white collar occupations
- **income**: percentage of families with income < USD 3000
- **hydrocarbon**: relative hydrocarbon pollution potential
- **nox**: relative nitric oxides potential
- **so2**: relative sulphur dioxide potential
- **humidity**: annual average percentage of relative humidity at 13:00
- **mortality**: total age-adjusted mortality rate per 100,000

### Source

http://lib.stat.cmu.edu/datasets/pollution
BIC.lmSubsets

References


Examples

```r
## load data (with logs for relative potentials)
data("AirPollution", package = "lmSubsets")
for (i in 12:14) AirPollution[[i]] <- log(AirPollution[[i]])

## fit subsets
lm_all <- lmSubsets(mortality ~ ., data = AirPollution)
plot(lm_all)

## refit best model
lm6 <- refit(lm_all, size = 6)
summary(lm6)
```

BIC.lmSubsets

*Extract BIC values from a subset regression*

Description

Evaluate the Bayesian information criterion (BIC) for the specified submodels.

Usage

```r
## S3 method for class 'lmSubsets'
BIC(object, size, best = 1, ..., na.rm = TRUE, drop = TRUE)

## S3 method for class 'lmSelect'
BIC(object, best = 1, ..., na.rm = TRUE, drop = TRUE)
```

Arguments

- `object` "lmSubsets", "lmSelect"—a subset regression
- `size` integer[]—the submodel sizes
- `best` integer[]—the submodel positions
- `...` ignored
- `na.rm` logical—if TRUE, remove NA entries
- `drop` logical—if TRUE, simplify structure

Value

double[]—the BIC values
See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `BIC()` for the S3 generic

---

**coef.lmSubsets**

*Extract the coefficients from a subset regression*

### Description

Return the coefficients for the specified submodels.

### Usage

```r
## S3 method for class 'lmSubsets'
coef(object, size, best = 1, ..., na.rm = TRUE, drop = TRUE)

## S3 method for class 'lmSelect'
coef(object, best = 1, ..., na.rm = TRUE, drop = TRUE)
```

### Arguments

- `object` 
  "lmSubsets","lmSelect"—a subset regression
- `size` 
  integer[]—the submodel sizes
- `best` 
  integer[]—the submodel positions
- `...` 
  ignored
- `na.rm` 
  logical—if TRUE, remove NA entries
- `drop` 
  logical—if TRUE, simplify structure

### Value

`double[,]","data.frame"`—the submodel coefficients

### See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `coef()` for the S3 generic
deviance.lmSubsets

Extract the deviance from a subset regression

Description

Return the deviance for the specified submodels.

Usage

## S3 method for class 'lmSubsets'
deviance(object, size, best = 1, ..., na.rm = TRUE, drop = TRUE)

## S3 method for class 'lmSelect'
deviance(object, best = 1, ..., na.rm = TRUE, drop = TRUE)

Arguments

object "lmSubsets","lmSelect"—a subset regression
size integer[]—the submodel sizes
best integer[]—the submodel positions
... ignored
na.rm logical—if TRUE, remove NA entries
drop logical—if TRUE, simplify structure

Value

double[], "data.frame"—the submodel deviances

See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `deviance()` for the S3 generic
fitted.lmSubsets  Extract the fitted values from a subset regression

Description

Return the fitted values for the specified submodel.

Usage

```r
## S3 method for class 'lmSubsets'
fitted(object, size, best = 1, ...)

## S3 method for class 'lmSelect'
fitted(object, best = 1, ...)
```

Arguments

- `object` "lmSubsets","lmSelect"—a subset regression
- `size` integer—the submodel size
- `best` integer—the submodel position
- `...` ignored

Value

double[]—the fitted values

See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `fitted()` for the S3 generic

formula.lmSubsets  Extract a formula from a subset regression

Description

Return the formula for the specified submodel.

Usage

```r
## S3 method for class 'lmSubsets'
formula(x, size, best = 1, ...)

## S3 method for class 'lmSelect'
formula(x, best, ...)
```
**IbkTemperature**

**Arguments**
- `x` "lmSubsets", "lmSelect"—a subset regression
- `size` integer—the submodel size
- `best` integer—the submodel position
- ... ignored

**Value**
- "formula"—the submodel formula

**See Also**
- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `formula()` for the S3 generic

---

**Description**

00UTC temperature observations and corresponding 24-hour reforecast ensemble means from the Global Ensemble Forecast System (GEFS, Hamill et al. 2013) for SYNOP station Innsbruck Airport (11120; 47.260, 11.357) from 2011-01-01 to 2015-12-31.

**Usage**

```r
data(IbkTemperature)
```

**Format**

A data frame containing 1824 daily observations/forecasts for 42 variables. The first column (`temp`) contains temperature observations at 00UTC (coordinated universal time), columns 2–37 are 24-hour lead time GEFS reforecast ensemble means for different variables (see below). Columns 38–42 are deterministic time trend/season patterns.

- **temp** observed temperature at Innsbruck Airport (deg C)
- **tp** total accumulated precipitation (kg m\(^{-2}\))
- **t2m** temperature at 2 meters (K)
- **u10m** U-component of wind at 10 meters (m s\(^{-1}\))
- **v10m** V-component of wind at 10 meters (m s\(^{-1}\))
- **u80m** U-component of wind at 80 meters (m s\(^{-1}\))
- **v80m** U-component of wind at 80 meters (m s\(^{-1}\))
cape  convective available potential energy ($J \text{ kg}^{-1}$)
ci  convective inhibition ($J \text{ kg}^{-1}$)
sdlwrf  surface downward long-wave radiation flux ($W \text{ m}^{-2}$)
sdswwrf  surface downward short-wave radiation flux ($W \text{ m}^{-2}$)
sulwrf  surface upward long-wave radiation flux ($W \text{ m}^{-2}$)
suswwrf  surface upward short-wave radiation flux ($W \text{ m}^{-2}$)
ghf  ground heat flux ($W \text{ m}^{-2}$)
slhnf  surface latent heat net flux ($W \text{ m}^{-2}$)
sshnf  surface sensible heat net flux ($W \text{ m}^{-2}$)
mslp  mean sea level pressure ($Pa$)
psfc  surface pressure ($Pa$)
pw  precipitable water ($kg \text{ m}^{-2}$)
vsmc  volumetric soil moisture content (fraction)
sh2m  specific humidity at 2 meters ($kg \text{ kg}^{-1}$)
tcc  total cloud cover (percent)
tcie  total column-integrated condensate ($kg \text{ m}^{-2}$)
tsfc  skin temperature ($K$)
tmax2m  maximum temperature ($K$)
tmin2m  minimum temperature ($K$)
st  soil temperature (0–10 cm below surface) ($K$)
ulwrf  upward long-wave radiation flux ($W \text{ m}^{-2}$)
wr  water runoff ($kg \text{ m}^{-2}$)
we  water equivalent of accumulated snow depth ($kg \text{ m}^{-2}$)
wp  wind mixing energy ($J$)
w850  vertical velocity at 850 hPa surface ($Pa \text{ s}^{-1}$)
t2pvu  temperature on 2 PVU surface ($K$)
p2pvu  pressure on 2 PVU surface ($Pa$)
u2pvu  U-component of wind on 2 PVU surface ($m \text{ s}^{-1}$)
v2pvu  U-component of wind on 2 PVU surface ($m \text{ s}^{-1}$)
pv  Potential vorticity on 320 K isentrope ($K \text{ m}^2 \text{ kg}^{-1} \text{ s}^{-1}$)
time  time in years
sin, cos  sine and cosine component of annual harmonic pattern
sin2, cos2  sine and cosine component of bi-annual harmonic pattern

Source

References


Examples

```r
## load data and omit missing values
data("IbkTemperature", package = "lmSubsets")
IbkTemperature <- na.omit(IbkTemperature)

## fit a simple climatological model for the temperature
## with a linear trend and annual/bi-annual harmonic seasonal pattern
CLIM <- lm(temp ~ time + sin + cos + sin2 + cos2, data = IbkTemperature)

## fit a simple MOS with 2-meter temperature forecast in addition
## to the climatological model
MOS0 <- lm(temp ~ t2m + time + sin + cos + sin2 + cos2, data = IbkTemperature)

## graphical comparison and MOS summary
plot(temp ~ time, data = IbkTemperature, type = "l", col = "darkgray")
lines(fitted(MOS0) ~ time, data = IbkTemperature, col = "darkred")
lines(fitted(CLIM) ~ time, data = IbkTemperature, lwd = 2)

## best subset selection of remaining variables for the MOS
## (i.e., forcing the regressors of m1 into the model)
MOS1_all <- lmSubsets(temp ~ ., data = IbkTemperature, include = c("t2m", "time", "sin", "cos", "sin2", "cos2"))
plot(MOS1_all)
image(MOS1_all, size = 8:20)

## compare BIC
BIC(CLIM, MOS0, MOS1, MOS2)
```
```r
nrow(IbkTemperature))

## compare coefficients
cf0 <- coef(CLIM)
cf1 <- coef(MOS0)
cf2 <- coef(MOS1)
cf3 <- coef(MOS2)
names(cf2) <- gsub("^x", "", names(coef(MOS1)))
names(cf3) <- gsub("^x", "", names(coef(MOS2)))
nam <- unique(c(names(cf0), names(cf1), names(cf2), names(cf3)))
cf <- matrix(NA, nrow = length(nam), ncol = 4,
             dimnames = list(nam, c("CLIM", "MOS0", "MOS1", "MOS2")))
cf[names(cf0), 1] <- cf0
cf[names(cf1), 2] <- cf1
cf[names(cf2), 3] <- cf2
cf[names(cf3), 4] <- cf3
print(round(cf, digits = 3), na.print = "")
```

### image.lmSubsets

**Heatmap of a subset regression**

**Description**

Plot a heatmap of the specified submodels.

**Usage**

```r
## S3 method for class 'lmSubsets'
image(x, size = NULL, best = 1, which = NULL, hilite, hilite_penalty,
      main, sub, xlab = NULL, ylab, ann = par("ann"), axes = TRUE,
      col = c("gray40", "gray90"), lab = "lab",
      col_hilite = cbind("red", "pink"), lab_hilite = "lab",
      pad_size = 3, pad_best = 1, pad_which = 3, axis_pos = -4,
      axis_tck = -4, axis_lab = -10, ...)

## S3 method for class 'lmSelect'
image(x, best = NULL, which = NULL, hilite, hilite_penalty,
      main, sub = NULL, xlab = NULL, ylab, ann = par("ann"),
      axes = TRUE, col = c("gray40", "gray90"), lab = "lab",
      col_hilite = cbind("red", "pink"), lab_hilite = "lab",
      pad_best = 2, pad_which = 2, axis_pos = -4, axis_tck = -4,
      axis_lab = -10, ...)
```

**Arguments**

- `x` — "lmSubsets", "lmSelect"—a subset regression
- `size, best` — submodels to be plotted
- `which` — regressors to be plotted
image.lmSubsets

hilite, hilite_penalty
submodels to be highlighted
main, sub, xlab, ylab
main, sub-, and axis titles
ann
annotate plot
axes
plot axes
col, lab
color and label style
col_hilite, lab_hilite
highlighting style
pad_size, pad_best, pad_which
padding
axis_pos, axis_tck, axis_lab
position of axes, tick length, and position of labels
...
ignored

Value
invisible(x)

See Also

• lmSubsets() for all-subsets regression
• lmSelect() for best-subset regression

Examples

## data
data("AirPollution", package = "lmSubsets")

###############
## lmSubsets ##
###############

lm_all <- lmSubsets(mortality ~ ., data = AirPollution, nbest = 20)

## heatmap
image(lm_all, best = 1:3)

## highlight 5 best (BIC)
image(lm_all, best = 1:3, hilite = 1:5, hilite_penalty = "BIC")

###############
## lmSelect ##
###############

## default criterion: BIC
lm_best <- lmSelect(lm_all)
## highlight 5 best (AIC)
image(lm_best, hilite = 1:5, hilite_penalty = "AIC")

## axis labels
image(lm_best, lab = c("bold(lab)", "lab"), hilite = 1,
    lab_hilite = "underline(lab")

---

**lmSelect**

*Best-subset regression*

**Description**

Best-variable-subset selection in ordinary linear regression.

**Usage**

```r
lmSelect(formula, ...)  
```

**Arguments**

- `formula`, `data`, `subset`, `weights`, `na.action`, `model`, `x`, `y`, `contrasts`, `offset`
  - standard formula interface
  - `...` forwarded to `lmSelect_fit()`

**Details**

The `lmSelect()` generic provides various methods to conveniently specify the regressor and response variables. The standard formula interface (see `lm()`) can be used, or the model information can be extracted from an already fitted "lm" object. The model matrix and response can also be passed in directly.

After processing the arguments, the call is forwarded to `lmSelect_fit()`.

**Value**

"lmSelect"—a list containing the components returned by `lmSelect_fit()`

Further components include `call`, `na.action`, `weights`, `offset`, `contrasts`, `xlevels`, `terms`, `mf`, `x`, and `y`. See `lm()` for more information.
See Also

- `lmSelect.matrix()` for the matrix interface
- `lmSelect.lmSubsets()` for coercing an all-subsets regression
- `lmSelect_fit()` for the low-level interface
- `lmSubsets()` for all-subsets regression

Examples

```r
## load data
data("AirPollution", package = "lmSubsets")

# basic usage
# fit 20 best subsets (BIC)
lm_best <- lmSelect(mortality ~ ., data = AirPollution, nbest = 20)
lm_best

# summary statistics
summary(lm_best)

# visualize
plot(lm_best)

# custom criterion
# the same as above, but with a custom criterion:
M <- nrow(AirPollution)
ll <- function (rss) {
  -M/2 * (log(2 * pi) - log(M) + log(rss) + 1)
}
aic <- function (size, rss, k = 2) {
  -2 * ll(rss) + k * (size + 1)
}
bic <- function (size, rss) {
  aic(size, rss, k = log(M))
}
lm_cust <- lmSelect(mortality ~ ., data = AirPollution,
                   penalty = bic, nbest = 20)
lm_cust
```
lmSelect.lmSubsets

Best-subset regression

Description

Coerce an all-subsets regression.

Usage

## S3 method for class 'lmSubsets'
lmSelect(formula, penalty = "BIC", ...)

Arguments

formula "lmSubsets"—an all-subsets regression
penalty double, character, "function"—penalty per model parameter
... ignored

Details

Computes a best-subset regression from an all-subsets regression.

Value

"lmSelect"—a best-subset regression

See Also

- `lmSelect()` for the S3 generic
- `lmSubsets()` for all-subsets regression

Examples

data("AirPollution", package = "lmSubsets")

lm_all <- lmSubsets(mortality ~ ., data = AirPollution, nbest = 20)

lm_best <- lmSelect(lm_all)

lm_best
**lmSelect.matrix**  

**Best-subset regression**

**Description**

Matrix interface to best-variable-subset selection in ordinary linear regression.

**Usage**

```r
# S3 method for class 'matrix'
lmSelect(formula, y, intercept = TRUE, ...)
```

**Arguments**

- `formula` "matrix"—the model matrix
- `y` double[]—the model response
- `intercept` logical[]—if FALSE, remove intercept term
- `...` forwarded to `lmSelect.default()`

**Details**

This is a utility interface. Use the standard formula interface wherever possible.

**Value**

"lmSelect"—a best-subset regression

**See Also**

- `lmSelect()` for the S3 generic
- `lmSelect.default()` for the standard formula interface

---

**lmSelect_fit**  

**Best-subset regression**

**Description**

Low-level interface to best-variable-subset selection in ordinary linear regression.

**Usage**

```r
lmSelect_fit(x, y, weights = NULL, offset = NULL, include = NULL, exclude = NULL, penalty = "BIC", tolerance = 0, nbest = 1, ..., pradius = NULL)
```
Arguments

- **x** double[,]—the model matrix
- **y** double[]—the model response
- **weights** double[]—the model weights
- **offset** double[]—the model offset
- **include** logical[], integer[], character[]—the regressors to force in
- **exclude** logical[], integer[], character[]—the regressors to force out
- **penalty** double, character, "function"—the penalty per model parameter
- **tolerance** double—the approximation tolerance
- **nbest** integer—the number of best subsets
- **pradius** integer—the preordering radius

Details

The best variable-subset model is determined, where the "best" model is the one with the lowest information criterion value. The information criterion belongs to the AIC family.

The regression data is specified with the **x**, **y**, **weights**, and **offset** parameters. See `lm.fit()` for further details.

To force regressors into or out of the regression, a list of regressors can be passed as an argument to the **include** or **exclude** parameters, respectively.

The information criterion is specified with the **penalty** parameter. Accepted values are "AIC", "BIC", or a "numeric" value representing the penalty-per-model-parameter. A custom selection criterion may be specified by passing an R function as an argument. The expected signature is function (size, rss), where size is the number of predictors (including the intercept, if any), and rss is the residual sum of squares. The function must be non-decreasing in both parameters.

An approximation **tolerance** can be specified to speed up the search.

The number of returned submodels is determined by the **nbest** parameter.

The preordering radius is given with the **pradius** parameter.

Value

A list with the following components:

- **NOBs** integer—number of observations in model (before weights processing)
- **nobs** integer—number of observations in model (after weights processing)
- **nvar** integer—number of regressors in model
- **weights** double[]—model weights
- **intercept** logical—is TRUE if model contains an intercept term, FALSE otherwise
- **include** logical[]—regressors forced into the regression
- **exclude** logical[]—regressors forced out of the regression
lmSubsets

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td>integer[]—subset sizes</td>
</tr>
<tr>
<td>ic</td>
<td>information criterion</td>
</tr>
<tr>
<td>tolerance</td>
<td>double—approximation tolerance</td>
</tr>
<tr>
<td>nbest</td>
<td>integer—number of best subsets</td>
</tr>
<tr>
<td>submodel</td>
<td>“data.frame”—submodel information</td>
</tr>
<tr>
<td>subset</td>
<td>“data.frame”—selected subsets</td>
</tr>
</tbody>
</table>

References


See Also

• lmSelect() for the high-level interface
• lmSubsets_fit() for all-subsets regression

Examples

```r
data("AirPollution", package = "lmSubsets")

x <- as.matrix(AirPollution[, names(AirPollution) != "mortality"])
y <- AirPollution[, names(AirPollution) == "mortality"]

f <- lmSelect_fit(x, y)
f
```

Description

All-variable-subsets selection in ordinary linear regression.

Usage

```r
lmSubsets(formula, ...) # Default S3 method:
```

```r
lmSubsets(formula, data, subset, weights, na.action,
model = TRUE, x = FALSE, y = FALSE, contrasts = NULL,
offset, ...)```

```r
```
Arguments

formula, data, subset, weights, na.action, model, x, y, contrasts, offset

standard formula interface

... forwarded to lmSubsets_fit()

Details

The lmSubsets() generic provides various methods to conveniently specify the regressor and response variables. The standard formula interface (see lm()) can be used, or the model information can be extracted from an already fitted "lm" object. The model matrix and response can also be passed in directly.

After processing of the arguments, the call is forwarded to lmSubsets_fit().

Value

"lmSubsets"—a list containing the components returned by lmSubsets_fit()

Further components include call, na.action, weights, offset, contrasts, xlevels, terms, mf, x, and y. See lm() for more information.

See Also

- lmSubsets.matrix() for the "matrix" interface
- lmSubsets_fit() for the low-level interface
- lmSelect() for best-subset regression

Examples

```r
## load data
data("AirPollution", package = "lmSubsets")

# basic usage
#
# canonical example: fit all subsets
lm_all <- lmSubsets(mortality ~ ., data = AirPollution, nbest = 5)
lm_all

# plot RSS and BIC
plot(lm_all)

# summary statistics
summary(lm_all)
```

```r
# forced in-/exclusion
```

```r
```
lmSubsets.matrix 21

lm_force <- lmSubsets(lm_all, include = c("nox", "so2"),
                   exclude = "whitecollar")
lm_force

lmSubsets.matrix  All-subsets regression

Description
Matrix interface to all-variable-subsets selection in ordinary linear regression.

Usage

## S3 method for class 'matrix'
lmSubsets(formula, y, intercept = TRUE, ...)

Arguments

formula  "matrix"—the model matrix
y        double[]—the model response
intercept logical—if FALSE, remove intercept term
...      forwarded to lmSubsets.default()

Details
This is a utility interface. Use the standard formula interface wherever possible.

Value
"lmSubsets"—an all-subsets regression

See Also
• lmSubsets() for the S3 generic
  • lmSubsets.default() for the standard formula interface

Examples

data("AirPollution", package = "lmSubsets")
x <- as.matrix(AirPollution)

lm_mat <- lmSubsets(x, y = "mortality")
lm_mat
Description
Low-level interface to all-variable-subsets selection in ordinary linear regression.

Usage
lmSubsets_fit(x, y, weights = NULL, offset = NULL, include = NULL, exclude = NULL, nmin = NULL, nmax = NULL, tolerance = 0, nbest = 1, ..., pradius = NULL)

Arguments
- x: double[, ]—the model matrix
- y: double[]—the model response
- weights: double[]—the model weights
- offset: double[]—the model offset
- include: logical[], integer[], character[]—the regressors to force in
- exclude: logical[], integer[], character[]—the regressors to force out
- nmin: integer—the minimum number of regressors
- nmax: integer—the maximum number of regressors
- tolerance: double[]—the approximation tolerances
- nbest: integer—the number of best subsets
- ...: ignored
- pradius: integer—the preordering radius

Details
The best variable-subset model for every subset size is determined, where the "best" model is the one with the lowest residual sum of squares (RSS).

The regression data is specified with the x, y, weights, and offset parameters. See lm.fit() for further details.

To force regressors into or out of the regression, a list of regressors can be passed as an argument to the include or exclude parameters, respectively.

The scope of the search can be limited to a range of subset sizes by setting nmin and nmax, the minimum and maximum number of regressors allowed in the regression, respectively.

A tolerance vector can be specified to speed up the search, where tolerance[j] is the approximation tolerance applied to subset models of size j.

The number of submodels returned for each subset size is determined by the nbest parameter.

The preordering radius is given with the pradius parameter.
Value

A list with the following components:

- **NOBS** integer—number of observations in model (before weights processing)
- **nobs** integer—number of observations in model (after weights processing)
- **nvar** integer—number of regressors in model
- **weights** double[]—model weights
- **intercept** logical—is TRUE if model contains an intercept term, FALSE otherwise
- **include** logical[]—regressors forced into the regression
- **exclude** logical[]—regressors forced out of the regression
- **size** integer[]—subset sizes
- **tolerance** double[]—approximation tolerances
- **nbest** integer—number of best subsets
- **submodel** "data.frame"—submodel information
- **subset** "data.frame"—variable subsets

References


See Also

- `lmSubsets()` for the high-level interface
- `lmSelect_fit()` for best-subset regression

Examples

data("AirPollution", package = "lmSubsets")

x <- as.matrix(AirPollution[, names(AirPollution) != "mortality"])
y <- AirPollution[, names(AirPollution) == "mortality"]

f <- lmSubsets_fit(x, y)

f
**logLik.lmSubsets**

Extract the log-likelihood from a subset regression

**Description**

Return the log-likelihood of the specified submodels.

**Usage**

```r
## S3 method for class 'lmSubsets'
logLik(object, size, best = 1, ..., na.rm = TRUE, drop = TRUE)

## S3 method for class 'lmSelect'
logLik(object, best = 1, ..., na.rm = TRUE, drop = TRUE)
```

**Arguments**

- `object` "lmSubsets","lmSelect"—a subset regression
- `size` integer[]—the submodel sizes
- `best` integer[]—the submodel positions
- `...` ignored
- `na.rm` logical—if TRUE, remove NA entries
- `drop` logical—if TRUE, simplify structure

**Value**

double[]—the log-likelihoods

**See Also**

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `logLik()` for the S3 generic
model.frame.lmSubsets  

Extract the model frame from a subset regression

Description

Return the model frame.

Usage

```r
## S3 method for class 'lmSubsets'
model.frame(formula, ...)

## S3 method for class 'lmSelect'
model.frame(formula, ...)
```

Arguments

- `formula`—a subset regression
- `...`—forwarded to `model.frame()`

Value

"data.frame"—the model frame

See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `model.frame()` for the S3 generic

model.matrix.lmSubsets

Extract a model matrix from a subset regression

Description

Returns the model matrix for the specified submodel.

Usage

```r
## S3 method for class 'lmSubsets'
model.matrix(object, size, best = 1, ...)

## S3 method for class 'lmSelect'
model.matrix(object, best, ...)
```
model_response

Arguments
- object: "lmSubsets", "lmSelect"—a subset regression
- size: integer—the submodel size
- best: integer—the submodel position
- ... forwarded to `model.frame()`

Value
- `double[,]`—the model matrix

See Also
- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `model.matrix()` for the S3 generic

Description
Extract the model response.

Usage
```r
model_response(data, ...)  
```

## Default S3 method:
```r
model_response(data, type = "any", ...)  
```

Arguments
- data: an object
- type: character—the return type
- ... further arguments

Details
The default method simply forwards the call to `model.response()`.

Value
- `double[,]`—the model response

See Also
- `model.response()` for the default implementation
model_response.lmSubsets

Extract the model response from a subset regression

Description

Return the model response.

Usage

## S3 method for class 'lmSubsets'
model_response(data, ...)

## S3 method for class 'lmSelect'
model_response(data, ...)

Arguments

data        "lmSubsets","lmSelect"—a subset regression
...          ignored

Value

double[]—the model response

See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `model_response()` for the S3 generic

plot.lmSubsets

Plot a subset regression

Description

Plot the deviance of the selected submodels, as well as a specified information criterion.
Usage

## S3 method for class 'lmSubsets'
plot(x, penalty = "BIC", xlim, ylim_rss, ylim_ic, type_rss = "o", type_ic = "o", main, sub, xlab, ylab_rss, ylab_ic, legend_rss, legend_ic, ann = par("ann"), axes = TRUE, lty_rss = c(1, 3), lty_ic = c(1, 3), pch_rss = c(16, 21), col_rss = "black", bg_rss = "white", pch_ic = c(16, 21), col_ic = "red", bg_ic = "white", ...)

## S3 method for class 'lmSelect'
plot(x, xlim, ylim, type = "o", main, sub, xlab, ylab, legend, ann = par("ann"), axes = TRUE, lty = 1, pch = 16, col = "red", bg = "white", ...)

Arguments

x "lmSubsets", "lmSelect"—a subset regression

penalty the information criterion

xlim, ylim, ylim_rss, ylim_ic x and y limits

type, type_rss, type_ic type of plot

main, sub main and sub-title

xlab, ylab, ylab_rss, ylab_ic axis titles

legend, legend_rss, legend_ic plot legend

ann annotate plot

axes plot axes

lty, lty_rss, lty_ic line type

pch, pch_rss, pch_ic plotting character

col, col_rss, col_ic color

bg, bg_rss, bg_ic background color

... further graphical parameters

Value

invisible(x)
See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `plot()` for the S3 generic

Examples

```r
## load data
data("AirPollution", package = "lmSubsets")

################################
## lmSubsets ##
################################
lm_all <- lmSubsets(mortality ~ ., data = AirPollution, nbest = 5)
plot(lm_all)

################################
## lmSelect ##
################################
lm_best <- lmSelect(mortality ~ ., data = AirPollution, nbest = 20)
plot(lm_best)
```

refit  
---

Refitting models

Description

Generic function for refitting a model on a subset or reweighted data set.

Usage

```r
refit(object, ...)
```

Arguments

- `object`  
an object to be refitted
- `...`  
forwarded arguments

Details

The `refit` generic is a new function for refitting a certain model object on multiple versions of a data set (and is hence different from `update`). Applications refit models after some kind of model selection, e.g., variable subset selection, partitioning, reweighting, etc.

The generic is similar to the one provided in `modeltools` and `fxregime` (and should fulfill the same purpose). To avoid dependencies, it is also provided here.
Value

"lm"—the refitted model

---

refit.lmSubsets  Refit a subset regression

Description

Fit the specified submodel and return the obtained "lm" object.

Usage

```r
## S3 method for class 'lmSubsets'
refit(object, size, best = 1, ...)

## S3 method for class 'lmSelect'
refit(object, best = 1, ...)
```

Arguments

- `object` "lmSubsets", "lmSelect"—a subset regression
- `size` integer—the submodel size
- `best` integer—the submodel position
- `...` ignored

Value

"lm"—the fitted model

See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `refit()` for the S3 generic

Examples

```r
## load data
data("AirPollution", package = "lmSubsets")

## fit subsets
lm_all <- lmSubsets(mortality ~ ., data = AirPollution)

## refit best model
lm5 <- refit(lm_all, size = 5)
summary(lm5)
```
residuals.lmSubsets  

Extract the residuals from all-subsets regression

Description

Return the residuals for the specified submodel.

Usage

## S3 method for class 'lmSubsets'
residuals(object, size, best = 1, ...)

## S3 method for class 'lmSelect'
residuals(object, best = 1, ...)

Arguments

- object: "lmSubsets", "lmSelect"—a subset regression
- size: integer—the submodel size
- best: integer—the submodel position
- ...: ignored

Value

double[]—the residuals

See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `residuals()` for the S3 generic

sigma.lmSubsets  

Extract the residual standard deviation from a subset regression

Description

Return the residual standard deviation for the specified submodels.

Usage

## S3 method for class 'lmSubsets'
sigma(object, size, best = 1, ..., na.rm = TRUE, drop = TRUE)

## S3 method for class 'lmSelect'
sigma(object, best = 1, ..., na.rm = TRUE, drop = TRUE)
Arguments

object "lmSubsets","lmSelect"—a subset regression
size integer[]—the submodel sizes
best integer[]—the submodel positions
... ignored
na.rm logical—if TRUE, remove NA entries
drop logical—if TRUE, simplify structure

Value
double[]—the residual standard deviations

See Also

• `lmSubsets()` for all-subsets regression
• `lmSelect()` for best-subset regression
• `sigma()` for the S3 generic

summary.lmSubsets

Summarize a subset regression

Description

Evaluate summary statistics for the selected submodels.

Usage

## S3 method for class 'lmSubsets'
summary(object, ..., na.rm = TRUE)

## S3 method for class 'lmSelect'
summary(object, ..., na.rm = TRUE)

Arguments

object "lmSubsets","lmSelect"—a subset regression
... ignored
na.rm if TRUE, remove NA values

Value

"summary.lmSubsets","summary.lmSelect"—a subset regression summary

See Also

• `lmSubsets()` for all-subsets regression
• `lmSelect()` for best-subset regression
variable.names.lmSubsets

Extract variable names from a subset regression

Description

Return the variable names for the specified submodels.

Usage

```r
## S3 method for class 'lmSubsets'
variable.names(object, size, best = 1, ..., na.rm = TRUE, drop = TRUE)

## S3 method for class 'lmSelect'
variable.names(object, best = 1, ..., na.rm = TRUE, drop = TRUE)
```

Arguments

- `object` "lmSubsets","lmSelect"—a subset regression
- `size` integer[]—the submodel sizes
- `best` integer[]—the submodel positions
- `...` ignored
- `na.rm` logical—if TRUE, remove NA entries
- `drop` logical—if TRUE, simplify structure

Value

logical[,],"data.frame"—the variable names

See Also

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `variable.names()` for the S3 generic
vcov.lmSubsets

**Extract the variance-covariance matrix from a subset regression**

**Description**

Return the variance-covariance matrix for the specified submodel.

**Usage**

```r
## S3 method for class 'lmSubsets'
vcov(object, size, best = 1, ...)

## S3 method for class 'lmSelect'
vcov(object, best = 1, ...)
```

**Arguments**

- `object` "lmSubsets","lmSelect"—a subset regression
- `size` integer—the submodel size
- `best` integer—the submodel position
- `...` ignored

**Value**

double[,]—the variance-covariance matrix

**See Also**

- `lmSubsets()` for all-subsets regression
- `lmSelect()` for best-subset regression
- `vcov()` for the S3 generic
Index

* datasets
  AirPollution, 4
  IbkTemperature, 9

* regression
  image.lmSelect, 12
  lmSelect, 14
  lmSelect.lmSubsets, 16
  lmSelect.matrix, 17
  lmSelect_fit, 17
  lmSubsets, 19
  lmSubsets.matrix, 21
  lmSubsets_fit, 22
  plot.lmSubsets, 27
  AIC, 18
  AIC(), 4
  AIC.lmSelect (AIC.lmSubsets), 3
  AIC.lmSubsets, 3
  AirPollution, 4
  BIC(), 6
  BIC.lmSelect (BIC.lmSubsets), 5
  BIC.lmSubsets, 5
  coef(), 6
  coef.lmSelect (coef.lmSubsets), 6
  coef.lmSubsets, 6
  deviance(), 7
  deviance.lmSelect (deviance.lmSubsets), 7
  deviance.lmSubsets, 7
  fitted(), 8
  fitted.lmSelect (fitted.lmSubsets), 8
  fitted.lmSubsets, 8
  formula(), 9
  formula.lmSelect (formula.lmSubsets), 8
  formula.lmSubsets, 8
  IbkTemperature, 9
  image.lmSelect (image.lmSubsets), 12
  image.lmSubsets, 12
  lm(), 14, 20
  lm.fit(), 18, 22
  lmSelect, 14
  lmSelect(), 4, 6–9, 13, 16, 17, 19, 20, 24–27, 29–34
  lmSelect.default(), 17
  lmSelect.lmSubsets, 16
  lmSelect.lmSubsets(), 15
  lmSelect.matrix, 17
  lmSelect.matrix(), 15
  lmSelect_fit, 17
  lmSelect_fit(), 14, 15, 23
  lmSubsets, 19
  lmSubsets(), 4, 6–9, 13, 15, 16, 21, 23–27, 29–34
  lmSubsets-package, 2
  lmSubsets.default(), 21
  lmSubsets.matrix, 21
  lmSubsets.matrix(), 20
  lmSubsets_fit, 22
  lmSubsets_fit(), 19, 20
  logLik(), 24
  logLik.lmSelect (logLik.lmSubsets), 24
  logLik.lmSubsets, 24
  model.frame(), 25, 26
  model.frame.lmSelect
    (model.frame.lmSubsets), 25
  model.frame.lmSubsets, 25
  model.matrix(), 26
  model.matrix.lmSelect
    (model.matrix.lmSubsets), 25
  model.matrix.lmSubsets, 25
  model.response(), 25
  model_response, 26
  model_response(), 27

35
model_response.lmSelect
  (model_response.lmSubsets), 27
model_response.lmSubsets, 27

plot(), 29
plot.lmSelect (plot.lmSubsets), 27
plot.lmSubsets, 27
print.lmSelect (lmSelect), 14
print.lmSubsets (lmSubsets), 19
print.summary.lmSelect
  (summary.lmSubsets), 32
print.summary.lmSubsets
  (summary.lmSubsets), 32

refit, 29
refit(), 30
refit.lmSelect (refit.lmSubsets), 30
refit.lmSubsets, 30
residuals(), 31
residuals.lmSelect
  (residuals.lmSubsets), 31
residuals.lmSubsets, 31

sigma(), 32
sigma.lmSelect (sigma.lmSubsets), 31
sigma.lmSubsets, 31
summary.lmSelect (summary.lmSubsets), 32
summary.lmSubsets, 32

variable.names(), 33
variable.names.lmSelect
  (variable.names.lmSubsets), 33
variable.names.lmSubsets, 33
vcov(), 34
vcov.lmSelect (vcov.lmSubsets), 34
vcov.lmSubsets, 34