# Package 'lmSubsets'

May 23, 2020

**Version** 0.5-1

**Date** 2020-05-23

**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) <10.18637/jss.v093.i03>.

**Depends** R (>= 3.4.0)

**SystemRequirements** C++11

**Imports** stats, graphics, utils

**License** GPL (>= 3)

**URL** https://github.com/marc-hofmann/lmSubsets.R

**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** 2020-05-23 16:50:06 UTC

## R topics documented:

---

`AirPollution` *Air Pollution and Mortality*

---

## Description

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

## Usage

```
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

## References

McDonald GC, Schwing RC (1973). Instabilities of Regression Estimates Relating Air Pollution to Mortality. *Technometrics*, **15**, 463–482.

Miller AJ (2002). *Subset Selection in Regression*. New York: Chapman and Hall.

## Examples

```
## 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)
```

---

`IbkTemperature` *Temperature Observations and Numerical Weather Predictions for Innsbruck*

---

## 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

```
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 ($J\ kg^{-1}$).

**ci** Convective inhibition ($J\ kg^{-1}$).

**sdlwrf** Surface downward long-wave radiation flux ($W\ m^{-2}$).

**sdswrf** Surface downward short-wave radiation flux ($W\ m^{-2}$).

**sulwrf** Surface upward long-wave radiation flux ($W\ m^{-2}$).

**suswrf** Surface upward short-wave radiation flux ($W\ m^{-2}$).

**ghf** Ground heat flux ($W\ m^{-2}$).

**slhnf** Surface latent heat net flux ($W\ m^{-2}$).

**sshnf** Surface sensible heat net flux ($W\ m^{-2}$).

**mslp** Mean sea level pressure ($Pa$).

**psfc** Surface pressure ($Pa$).

**pw** Precipitable water ($kg\ m^{-2}$).

**vsmc** Volumetric soil moisture content (fraction).

**sh2m** Specific humidity at 2 meters ($kg\ kg^{-1}$).

**tcc** Total cloud cover (percent).

**tcic** Total column-integrated condensate ($kg\ m^{-2}$).

**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 ($W\ m^{-2}$).

**wr** Water runoff ($kg\ m^{-2}$).

**we** Water equivalent of accumulated snow depth ($kg\ m^{-2}$).

**wp** Wind mixing energy ($J$).

**w850** Vertical velocity at 850 hPa surface ($Pa\ s^{-1}$).

**t2pvu** Temperature on 2 PVU surface ($K$).

**p2pvu** Pressure on 2 PVU surface ($Pa$).

**u2pvu** U-component of wind on 2 PVU surface ($m\ s^{-1}$).

**v2pvu** U-component of wind on 2 PVU surface ($m\ s^{-1}$).

**pv** Potential vorticity on 320 K isentrope ($K\ m^2\ kg^{-1}\ s^{-1}$).

**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

Observations: http://www.ogimet.com/synops.phtml.en

Reforecasts: http://www.esrl.noaa.gov/psd/forecasts/reforecast2/

## References

Hamill TM, Bates GT, Whitaker JS, Murray DR, Fiorino M, Galarneau Jr. TJ, Zhu Y, Lapenta W (2013). NOAA's Second-Generation Global Medium-Range Ensemble Reforecast Data Set. *Bulletin of the American Meteorological Society*, **94**(10), 1553–1565. doi: 10.1175/BAMSD12-00014.1

## Examples

```
## 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)
MOS0

## 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)
## -> Note that soil temperature and maximum temperature are selected
## in addition to the 2-meter temperature

## best subset selection of all variables
MOS2_all <- lmSubsets(temp ~ ., data = IbkTemperature)
plot(MOS2_all)
image(MOS2_all, size = 2:20)
## -> Note that 2-meter temperature is not selected into the best
## BIC model but soil-temperature (and maximum temperature) are used instead

## refit the best BIC subset selections
MOS1 <- refit(lmSelect(MOS1_all))
MOS2 <- refit(lmSelect(MOS2_all))

## compare BIC
BIC(CLIM, MOS0, MOS1, MOS2)

## compare RMSE
```

```
sqrt(sapply(list(CLIM, MOS0, MOS1, MOS2), deviance)/
  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 *Variable Selection Heatmaps*

---

## Description

Visualization of variable subsets.

## Usage

```
## 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` | An object of class `"lmSubsets"` or `"lmSelect"`. |
| `main, sub, xlab, ylab` | Main, sub- and axis titles. |
| `size, best` | Submodels to be plotted. |
| `which` | Regressors to be plotted. |
| `col, lab` | Color and label style. |
| `hilite, hilite_penalty` | Submodels to be highlighted. |
| `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. |
| `axes` | Plot axes. |
| `ann` | Annotate plot. |

## See Also

`lmSubsets`, `lmSelect`.

## 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-subset regression for ordinary linear models.

## Usage

```
lmSelect(formula, ...)

## Default S3 method:
lmSelect(formula, data, subset, weights, na.action, model = TRUE,
         x = FALSE, y = FALSE, contrasts = NULL, offset, ...)

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

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

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

## Arguments

| | |
|---|---|
| `formula`, `data`, `subset`, `weights`, `na.action`, `model`, `x`, `y`, `contrasts`, `offset` | Standard formula interface. |
| `intercept` | Include intercept. |
| `include`, `exclude` | Force regressors in or out. |
| `penalty` | Penalty per parameter. |
| `tolerance` | Approximation tolerance. |
| `nbest` | Number of best subsets. |
| `...` | Forwarded to `lmSelect_fit`. |
| `pradius` | Preordering radius. |

## Details

The lmSelect generic provides a convenient interface for best variable-subset selection in linear regression: The nbest best – according to an information criterion of the AIC family – subset models are returned.

The information criterion is specified with the penalty parameter. Accepted values are "AIC", "BIC", or a numeric value representing the penalty per model parameter (see AIC).

A custom selection criterion may be specified by passing an R function as the penalty argument. The expected signature is function(size,rss), where size is the number of predictors (including intercept, if any), and rss the residual sum of squares. The function must be non-decreasing in both parameters.

A low-level matrix interface is provided by lmSelect_fit.

See lmSubsets for further information.

## Value

An object of class "lmSelect", i.e., a list with the following components:

| | |
|---|---|
| nobs, nvar | Number of observations, of variables. |
| intercept | TRUE if model has intercept term; FALSE otherwise. |
| include, exclude | Included, excluded variables. |
| size | Subset sizes. |
| tolerance | Approximation tolerance. |
| nbest | Number of best subsets. |
| submodel | Submodel information. |
| subset | Selected variables. |

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

## References

Hofmann M, Gatu C, Kontoghiorghes EJ, Colubi A, Zeileis A (2020). lmSubsets: Exact Variable-Subset Selection in Linear Regression for R. *Journal of Statistical Software*. **93**, 1–21. doi:10.18637/jss.v093.i03.

## See Also

lmSubsets, summary, methods.

## Examples

```
## load data (with logs for relative potentials)
data("AirPollution", package = "lmSubsets")


###################
##  basic usage  ##
```

```
###################

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

## equivalent to:
## Not run:
lm_all <- lmSubsets(mortality ~ ., data = AirPollution, nbest = 20)
lm_best <- lmSelect(lm_all)

## End(Not run)

## 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
```

---

lmSubsets *All-Subsets Regression*

---

## Description

All-subsets regression for linear models estimated by ordinary least squares (OLS).

## Usage

```
lmSubsets(formula, ...)

## Default S3 method:
lmSubsets(formula, data, subset, weights, na.action, model = TRUE,
          x = FALSE, y = FALSE, contrasts = NULL, offset, ...)

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

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

## Arguments

| | |
|---|---|
| `formula, data, subset, weights, na.action, model, x, y, contrasts, offset` | Standard formula interface. |
| `intercept` | Include intercept. |
| `include, exclude` | Force regressors in or out. |
| `nmin, nmax` | Minimum and maximum number of regressors. |
| `tolerance` | Approximation tolerance (vector). |
| `nbest` | Number of best subsets. |
| `...` | Forwarded to `lmSubsets.default` and `lmSubsets_fit`. |
| `pradius` | Preordering radius. |

## 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 information can be extracted from an already fitted `"lm"` object. The regressor matrix and response variable can also be passed in directly (see Examples).

The call is forwarded to `lmSubsets_fit`, which provides a low-level matrix interface.

The `nbest` best subset models for every subset size are computed, where the "best" models are the models with the lowest residual sum of squares (RSS). The scope of the search can be limited to a range of subset sizes by setting `nmin` and `nmax`. A tolerance vector (expanded if necessary) may be specified to speed up the search, where `tolerance[j]` is the tolerance applied to subset models of size `j`.

By way of `include` and `exclude`, variables may be forced in to or out of the regression, respectively.

The extent to which variables are preordered is controlled with the `pradius` parameter.

A set of standard extractor functions for fitted model objects is available for objects of class `"lmSubsets"`. See `methods` for more details.

The `summary` method can be called to obtain summary statistics.

## Value

An object of class "lmSubsets", i.e., a list with the following components:

| | |
|---|---|
| nobs, nvar | Number of observations, of variables. |
| intercept | TRUE if model has intercept term; FALSE otherwise. |
| include, exclude | Included, excluded regressors. |
| size | Subset sizes. |
| tolerance | Approximation tolerance (vector). |
| nbest | Number of best subsets. |
| submodel | Submodel information. |
| subset | Selected variables. |

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

## References

Hofmann M, Gatu C, Kontoghiorghes EJ, Colubi A, Zeileis A (2020). lmSubsets: Exact Variable-Subset Selection in Linear Regression for R. *Journal of Statistical Software*. **93**, 1–21. doi:10.18637/jss.v093.i03.

Hofmann M, Gatu C, Kontoghiorghes EJ (2007). Efficient Algorithms for Computing the Best Subset Regression Models for Large-Scale Problems. *Computational Statistics \& Data Analysis*, **52**, 16–29. doi:10.1016/j.csda.2007.03.017.

Gatu C, Kontoghiorghes EJ (2006). Branch-and-Bound Algorithms for Computing the Best Subset Regression Models. *Journal of Computational and Graphical Statistics*, **15**, 139–156. doi:10.1198/106186006x100290.

## See Also

lmSelect, summary, methods.

## Examples

```
## load data (with logs for relative potentials)
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)


############################
##  forced in-/exclusion  ##
############################

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


########################
##  matrix interface  ##
########################

## same as above
x <- as.matrix(AirPollution)

lm_mat <- lmSubsets(x, y = "mortality")
lm_mat
```

---

`methods` *Methods for 'lmSubsets' and 'lmSelect' Objects*

---

## Description

Extractor methods for "`lmSubsets`" and "`lmSelect`" objects.

## Usage

```
## 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)

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

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

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

```
model.matrix(object, best, ...)

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

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

## 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)

## 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)

## 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)

## 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)

## 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)

## 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)

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

## S3 method for class 'lmSubsets'
```

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

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

## Arguments

| | |
|---|---|
| object, formula, data, x | An object of class "lmSubsets" or "lmSelect". |
| size | The submodel size. |
| best | The submodel ranking. |
| ... | Forwarded arguments. |
| k | AIC penalty. |
| drop | Control shape of return value. |
| na.rm | Remove missing submodels. |

## Details

The extractor methods work for "lmSubsets" and "lmSelect" objects, i.e., objects that have been generated using the formula interface.

If drop == FALSE, the extractor methods variable.names, deviance, sigma, logLik, AIC, BIC and coef return a data.frame object. If drop == TRUE, the return value is a logical matrix (variable.names), a numeric matrix (coef), or a numeric vector. If the drop parameter is not set explicitly when calling variable.names or coef, one-dimensional values are represented in a compact form.

If a desired extractor function is not available, refit can be called explicitly to obtain the corresponding "lm" object.

## See Also

lmSubsets, lmSelect, refit.

## Examples

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

## fit subsets (5 best subsets per size)
lm_all <- lmSubsets(mortality ~ ., data = AirPollution, nbest = 5)

## extract information (for best submodel of size 3)
coef(lm_all, size = 3)
vcov(lm_all, size = 3)
residuals(lm_all, size = 3)
```

```
fitted(lm_all, size = 3)
model.matrix(lm_all, size = 3)

## select best (BIC) submodels
lm_best <- lmSelect(lm_all)

## extract information
deviance(lm_best)
logLik(lm_best)
AIC(lm_best)
BIC(lm_best, best = 1:5)

## refit model
lm5 <- refit(lm_all, size = 5)
summary(lm5)
## (Note that the p-values are not valid due to model selection.)
```

---

`refit` *Refitting Models*

---

## Description

Generic function for refitting a model on various subsets or reweighted data sets.

## Usage

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

## Arguments

| | |
|---|---|
| `object` | An object. |
| `...` | 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.

## See Also

`methods`

# Index