Package 'BAQM'

May 6, 2026

Type Package

Title Babson Analytics and Quantitative Methods Tools

Version 0.1.4

Copyright Copyright Peter Lert, PhD, CFA

Description Instructor-developed tools for Analytics and Quantitative Methods (AQM) courses at Babson College. Included are compact descriptive statistics for data frames and lists, expanded reporting and graphics for linear regressions, and formatted reports for best subsets analyses.

License GPL (>= 2)

Depends R (>= 3.5),

Suggests leaps, shiny, testthat (>= 3.0.0), utf8, vdiffr, waldo, withr

Config/testthat/edition 3

Encoding UTF-8

Imports cowplot, ggplot2, ggrepel, lmtest, stats, utils

URL https://github.com/CPA-wrk/BAQM, https://cpa-wrk.github.io/BAQM/

BugReports https://github.com/CPA-wrk/BAQM/issues

RoxygenNote 7.3.3

NeedsCompilation no

Author Peter Lert [aut, cre, cph]

Maintainer Peter Lert <plert@babson.edu>

Repository CRAN

Date/Publication 2026-01-17 19:50:17 UTC

Contents

`lm_plot.4way` *Create a Four-Panel Regression Assumptions Plot*

Description

Generates a 4-panel diagnostic plot for a multiple regression model, including: 1) Fitted values vs. observed values (check for non-linearity), 2) Quantile–Quantile plot of residuals (check for non-normality), 3) Residuals vs. fitted values (check for heteroskedasticity), 4) Autocorrelation for time series otherwise influence plot (leverage also available).

Usage

```
lm_plot.4way(
  mdl,
  ...,
  is.ts = FALSE,
  pred.intvl = TRUE,
  pval.SW = FALSE,
  pval.BP = FALSE,
  pval.DW = FALSE,
  cook.loess = FALSE,
  rtn.all = FALSE,
  plt.nms = c("fit", "var", "qq", ifelse(is.ts, "ac", "infl")),
  parms = list()
)
```

Arguments

`mdl`	A fitted model object (typically from `lm`).
`...`	Additional arguments (not currently used).
`is.ts`	Logical; `TRUE` if data are time series, `FALSE` otherwise.

pred.intvl	Logical; plot prediction interval on fitted vs observed.
pval.SW	Logical; include Shapiro–Wilk p-value in qq plot.
pval.BP	Logical; include Breusch–Pagan p-value in var plot.
pval.DW	Logical; include Durbin–Watson p-value in ac plot.
cook.loess	Logical; overlay Cook's distance loess curve in infl plot.
rtn.all	Logical; return all plots and parameters (vs. 4-way plot only).
plt.nms	Character vector of which panels to plot. Defaults to fit, var, qq, and ac/infl depending on is.ts (Order: start upper left, continue clockwise)
parms	List of overrides to plot formatting parameters (see lm_plot.parms).

Details

This function is a high-level wrapper that calls internal plotting functions (lm_plot.fit, lm_plot.var, lm_plot.qq, and either lm_plot.ac or lm_plot.infl) and assembles their outputs into a combined plot_grid.

Value

A ggplot object representing the 4-way diagnostic panel. Optionally invisibly returns a list containing:

- p_4way – the combined plot,
- other elements passed through from the individual plot functions.

Examples

```
fit <- lm(mpg ~ wt + hp, data = mtcars)
lm_plot.4way(fit, is.ts = FALSE, pval.SW = TRUE)
fit <- lm(Employed ~ ., data = longley)
lm_plot.4way(fit, is.ts = TRUE, pval.DW = TRUE)
```

---

lm_plot.ac *Plot Residuals vs. Observation Order (Autocorrelation Check)*

Description

Creates a plot of residuals against the sequence/order of observations to visually assess independence and detect autocorrelation. Outliers are labeled by default. Optionally includes p-value result from the Durbin–Watson test for autocorrelation.

Usage

```
lm_plot.ac(
  mdl,
  ...,
  pval.DW = FALSE,
  parms = lm_plot.parms(mdl),
  df = lm_plot.df(mdl, parms = parms)
)
```

Arguments

mdl	A fitted model object (typically from lm).
...	Additional arguments (not currently used).
pval.DW	(logical, default = FALSE) Option to show Durbin-Watson p-value on the plot.
parms	A list of plotting parameters, usually from lm_plot.parms().
df	Data frame with augmented model data. Defaults to lm_plot.df(mdl).

Details

Points are colored and shaped according to whether they are residual outliers (as determined by Tukey's boxplot rule). The function can label points using **ggrepel** if parmptsid$outl or parm$ptsidreg are set to TRUE.

Value

A ggplot object representing the residuals vs. order plot. Included as an attribute "parm" is a list containing:

- lim Plotted limits on x and y axes,
- pval.DW Option to show Durbin-Watson p-value,
- DW The htest object with Durbin-Watson test results.

See Also

dwtest, lm_plot.df, lm_plot.parms

Examples

```
fit <- lm(res ~ ., data = data.frame(time = time(austres), res = austres))
lm_plot.ac(fit, pval.DW = TRUE)
```

`lm_plot.df` *Augment Model Data for Diagnostic Plots*

Description

Generates an augmented data frame from a linear model object, including fitted values, residuals, leverage, Cook's distance, prediction intervals, and outlier/influence flags. This function prepares model diagnostics for plotting.

Usage

```
lm_plot.df(mdl, parms = lm_plot.parms(mdl))
```

Arguments

`mdl`	An object of class `lm`, representing the fitted linear model.
`parms`	List of plotting parameters, usually from `lm_plot.parms()`.

Details

The returned data frame contains key statistics for each observation:

- `.id`: Observation identifier
- `.sequence`: Sequence index
- `.obs`: Observed values
- `.fits`: Fitted/predicted values
- `.resid`: Residuals
- `.se.fit`: Estimated standard error of fitted (mean) value
- `.lower.pi`: Lower bound on 95% prediction interval of fitted value
- `.upper.pi`: Upper bound on 95% prediction interval of fitted value
- `.std.resid`: Standardized residuals
- `.stud.resid`: Studentized residuals
- `.cooksd`: Cook's distance
- `.hat`: Leverage (diagonal of the hat matrix)
- `.quantile`: Theoretical normal quantile of residuals
- `.is.outl`: Flag for residual outlier ("outl" or "reg")
- `.is.infl`: Flag for influential point ("infl" or "outl" or "reg")
- `.is.lev`: Flag for high leverage point ("lev" or "infl" or "reg")

Value

A data frame with augmented diagnostic variables, one row per observation.

See Also

influence, influence.measures, rstandard, rstudent, outlier

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width, data = iris)
df <- lm_plot.df(mdl)
head(df)
```

`lm_plot.fit` *Plot Observed vs. Fitted Values to Check Linearity*

Description

Generates a scatter plot of fitted values versus observed values from a linear model, with identification of outlier points and optional prediction interval. The plot includes a reference line `y = x` for assessing linearity.

Usage

```
lm_plot.fit(
  mdl,
  ...,
  pred.intvl = TRUE,
  parms = lm_plot.parms(mdl),
  df = lm_plot.df(mdl, parms = parms)
)
```

Arguments

`mdl`	A fitted model object (typically from `lm`).
`...`	Additional arguments (not currently used).
`pred.intvl`	(logical, default = TRUE) Option to show prediction interval bounds of fitted values.
`parms`	List of plotting parameters, usually from `lm_plot.parms()`.
`df`	Data frame with augmented model data. Defaults to `lm_plot.df(mdl)`.

Details

The plot visualizes fitted versus observed values, includes a diagonal reference line, marks outliers, and can optionally display prediction intervals. Outlier and regular points can be labeled. This plot is useful for visually assessing linearity and model fit quality.

Value

A ggplot object representing the fitted vs. observed plot. Included as an attribute "parm" is a list containing:

- lim Plotted limits on x and y axes,
- pred.intvl Option to show prediction interval,

See Also

lm_plot.df, lm_plot.parms

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width, data = iris)
lm_plot.fit(mdl)
```

---

lm_plot.infl *Plot Leverage vs. Fitted Values to Visualize Inflential Observations*

Description

Creates a plot of leverage values versus the linear fitted values, including an identification of points with a large Cook's distance, to visualize high-leverage and influential observations.

Usage

```
lm_plot.infl(
  mdl,
  ...,
  parms = lm_plot.parms(mdl),
  df = lm_plot.df(mdl, parms = parms)
)
```

Arguments

mdl	A fitted model object (typically from lm).
...	Additional arguments (not currently used).
parms	List of plotting parameters, usually from lm_plot.parms().
df	Data frame with augmented model data. Defaults to lm_plot.df(mdl).

Details

The plot visualizes the calculated leverage of individual data points, defined as the diagonal element of the 'hat' matrix, as a function of the fitted values and implicitly relative to their location in the field of predictor variables, and the threshold value of high leverage is indicated. In addition, Cook's distance can be used to label influential points, along with outlier and regular points.

Value

A ggplot object representing the comb plot of residuals vs sequence, indicating influential points. Included as an attribute "parms" is a list containing:

- lim Plotted limits on x and y axes.

See Also

lm_plot.df, lm_plot.parms, outlier

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width, data = iris)
result <- lm_plot.infl(mdl)
```

---

lm_plot.lev *Plot Standard Residuals vs. Leverage with Cook's Distance Contours*

Description

Creates a plot of standard residuals versus leverage values, including Cook's distance contours to visualize influential observations.

Usage

```
lm_plot.lev(
  mdl,
  ...,
  cook.loess = FALSE,
  parms = lm_plot.parms(mdl),
  df = lm_plot.df(mdl, parms = parms)
)
```

Arguments

mdl	A fitted model object (typically from lm).
...	Additional arguments (not currently used).
cook.loess	Option (logical, default = FALSE) indicates whether to show loess curve for Cook's distances on the plot.
parms	List of plotting parameters, usually from lm_plot.parms().
df	Data frame with augmented model data. Defaults to lm_plot.df(mdl).

Details

The plot displays standardized residuals against leverage, overlays Cook's distance contours, and marks outliers based on residuals and Cook's distance. Outlier and influential points can be labeled, and a loess fit line is optionally added.

Value

A ggplot object representing the standardized residuals vs leverage plot. Included as an attribute "parms" is a list containing:

- lim Plotted limits on x and y axes,
- cook.loess Option to show loess curve for Cook's distances.

See Also

lm_plot.df, lm_plot.parms

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width, data = iris)
lm_plot.lev(mdl)
```

---

lm_plot.parms *Set or Retrieve Default Plot Parameters for Model Diagnostic Plots*

Description

Initializes or updates a list of plot element parameters for use in model diagnostic plotting functions. If any required parameter is missing or invalid, a default value will be supplied.

Usage

```
lm_plot.parms(mdl, parms = list())
```

Arguments

mdl	An object of class lm.
parms	A list of plot parameters. Any missing or invalid entries are replaced with defaults.

Details

The returned list contains parameters for points (size, color, shape), lines (type, color, size), options for plot features, Cook's distance/Influence contours, and seed values for sampling functions. These are used by other lm_plot.* functions to control plot appearance and annotation. Key defaults include:

- **pts**: Properties for points (size, color, shape, outlier flags)
- **lins**: Properties for lines (type, color, size)
- **cook**: Cook's distance contour settings (points, levels, line type)
- **infl**: Influence line settings (line type)

See function code for full list of available settings and default values.

Value

A list of plot element parameters with defaults filled in for any missing or invalid entries.

See Also

lm_plot.fit, lm_plot.lev, lm_plot.infl outlier

Examples

```
# Retrieve default parameters
parms <- lm_plot.parms(mdl = lm(Sepal.Length ~ Sepal.Width, data = iris))
# Set custom color for regular points
parms <- lm_plot.parms(mdl = lm(Sepal.Length ~ Sepal.Width, data = iris),
                       parms = list(pts = list(colr = list(reg = "blue"))))
```

---

lm_plot.qq *Q-Q Plot of Residuals to Assess Normality*

Description

Produces a Q-Q plot of residuals from a linear model to test for normality, optionally annotating outlier points and adding the Shapiro-Wilk test p-value.

Usage

```
lm_plot.qq(
  mdl,
  ...,
  pval.SW = FALSE,
  parms = lm_plot.parms(mdl),
  df = lm_plot.df(mdl, parms = parms)
)
```

Arguments

mdl	A fitted model object (typically from lm).
...	Additional arguments (not currently used).
pval.SW	(logical, default = FALSE) indicates whether to include Shapiro-Wilk p-value on the plot.
parms	List of plotting parameters, usually from lm_plot.parms().
df	Data frame with augmented model data. Defaults to lm_plot.df(mdl).

Details

The plot visualizes the distribution of residuals against theoretical normal quantiles. Points are colored and shaped by outlier status, and a reference Q-Q line is added. Optionally, outlier and regular points can be labeled. If enabled, the Shapiro-Wilk normality test p-value is displayed.

Value

A ggplot object representing the quantile-quantile plot. Included as an attribute "parm" is a list containing:

- lim Plotted limits on x and y axes,
- pval.SW Option to show Shapiro-Wilk p-value,
- DW The htest object with Shapiro-Wilk test results.

See Also

lm_plot.df, lm_plot.parms, shapiro.test

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width, data = iris)
lm_plot.qq(mdl)
```

---

lm_plot.var *Plot Residuals vs. Fitted Values to Assess Homoskedasticity*

Description

Produces a scatter plot of residuals against fitted values from a linear model, highlighting outlier points and optionally displaying the Breusch-Pagan test p-value for heteroskedasticity.

Usage

```
lm_plot.var(
  mdl,
  ...,
  pval.BP = FALSE,
  parms = lm_plot.parms(mdl),
  df = lm_plot.df(mdl, parms = parms)
)
```

Arguments

mdl	A fitted model object (typically from lm).
...	Additional arguments (not currently used).
pval.BP	(logical, default = FALSE) option to include Breusch-Pagan p-value on the plot.
parms	List of plotting parameters, usually from lm_plot.parms().
df	Data frame with augmented model data. Defaults to lm_plot.df(mdl).

Details

The plot visualizes residuals versus fitted values to assess homoskedasticity (constant variance). Points are colored and shaped by outlier status, and outlier/regular points can be labeled. The Breusch-Pagan test for heteroskedasticity is run and, if enabled, its p-value annotates the plot.

Value

A `ggplot` object representing the residuals versus fitted values plot. Included as an attribute `"parms"` is a list containing:

- `lim` Plotted limits on `x` and `y` axes,
- `pval.BP` Option to show Breusch-Pagan p-value,
- `BP` The `htest` object with Breusch-Pagan test results.

See Also

`lm_plot.df`, `lm_plot.parms`, `bptest`

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width, data = iris)
lm_plot.var(mdl, pval.BP = TRUE)
```

---

outlier *Identify Outliers Using Boxplot Heuristic*

Description

Detects outliers in a numeric vector using the boxplot heuristic (default, outside 1.5 times the IQR from the 1st and 3rd quartiles). Handles missing values (NA) automatically.

Usage

```
outlier(x, coef = 1.5, rpt = FALSE)
```

Arguments

`x`	Numeric vector for which outliers are to be detected.
`coef`	Numeric value for boxplot heuristic coefficient (default is 1.5). Higher values result in fewer points being classified as outliers.
`rpt`	Logical. If `FALSE` (default), returns a logical vector indicating which elements of `x` are outliers. If `TRUE`, returns the calculated lower and upper limits for outlier detection, identified by `x < rpt[1] | x > rpt[2]`.

Value

If rpt = FALSE, returns a logical vector indicating outliers. If rpt = TRUE, returns a numeric vector of length 2 giving lower and upper limits for outlier detection

Examples

```
set.seed(1)
vals <- c(rnorm(100), 10, -10)
outlier(vals)
outlier(vals, rpt = TRUE)
```

---

print.sumry.lm *Print a* sumry *Summarization for Linear Model Objects*

Description

Prints a comprehensive summary for objects of class summary.lm or lm, including model statistics, ANOVA table, coefficients, and optional tables (correlations, covariance, fits), followed by a five-number summary of residuals and the model call.

Usage

```
## S3 method for class 'sumry.lm'
print(
  x,
  ...,
  digits = max(5, getOption("digits") - 2),
  symbolic.cor = NULL,
  signif.stars = getOption("show.signif.stars"),
  options = NULL,
  na.print = "",
  eps = .Machine$double.eps
)
```

Arguments

x	An object of class sumry.lm or lm.
...	Additional arguments (not currently used).
digits	Minimal number of significant digits. Defaults to max(5, getOption("digits") - 2).
symbolic.cor	Not implemented. Defaults to NULL.
signif.stars	Logical; whether to show significance stars in the coefficients table. Defaults to getOption("show.signif.stars").
options	A character vector of optional summary tables to print (e.g., "v.correlation", "cov.unscaled", "correlation", "fits"). Printed in the given order if present.

na.print	String to use for NA values in the tables.
eps	Smallest positive floating-point value, used for formatting near-zero residuals. Defaults to .Machine$double.eps.

Details

The function prints summary statistics, ANOVA, and coefficients tables for a linear model in order, along with specified optional tables if provided. It concludes with a five-number-plus-mean summary of residuals and the model call.

Value

Invisibly return the unmodified object included in the call.

See Also

sumry.lm, print.table.sumry.lm

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width, data = iris)
sumry(mdl)
sumry(mdl, options = c("correlation", "fits"))
```

---

print.sumry.regsubsets

*Print Method for Best Subset Selection (*regsubsets*) Objects*

Description

Prints a summary for objects of class summary.regsubsets or regsubsets (from the leaps package), showing model selection statistics for best subsets, including R-squared, adjusted R-squared, standard error of estimate, Mallows' Cp, and AIC.

Usage

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

Arguments

x	An object of class summary.regsubsets or regsubsets.
...	Additional arguments (not currently used).

Details

The function prints the model call and a table summarizing the best models selected, including the number of predictors, R-squared, adjusted R-squared, standard error of estimate (SEE), Mallows' Cp, and included variables. If the input is a `regsubsets` object, it is converted with `sumry()`. If not, the object is returned unmodified.

Value

Invisibly return the object printed.

See Also

`regsubsets`, `sumry.regsubsets`

Examples

```
fit <- leaps::regsubsets(Fertility ~ ., data = swiss, nbest = 3)
print(sumry(fit))
```

---

print.table.sumry.lm *Print a Table from Linear Model Summary*

Description

Prints a formatted table from a sumry of a linear model, including coefficients, ANOVA, correlation matrices, or summary statistics. Significance stars and legends are added as appropriate.

Usage

```
## S3 method for class 'table.sumry.lm'
print(
  x,
  ...,
  digits = max(4, getOption("digits") - 2),
  quote = FALSE,
  na.print = "",
  zero.print = "0",
  right = TRUE,
  justify = "right",
  signif.stars = getOption("show.signif.stars"),
  eps = .Machine$double.eps,
  nsmall = 4,
  prnt.lgnd = c("coefficients"),
  dig.test = max(1, min(5, digits - 2))
)
```

Arguments

x	A table object from a linear model sumry (e.g., coefficients, ANOVA, statistics, correlation matrices).
...	Additional arguments (not currently used).
digits	Number of significant digits to print. Defaults to `max(4, getOption("digits") - 2)`.
quote	Logical; whether to print with quotes (default: FALSE).
na.print	String to use for NA values (default: '').
zero.print	String to use for 0 values (default: 0).
right	Logical, indicating whether or not strings should be right aligned (default: TRUE).
justify	Justification for columns ("right" or "left"; default: "right").
signif.stars	Logical; whether to show significance stars for p-values (default: `getOption("show.signif.stars")`).
eps	Smallest positive floating-point value for formatting near-zero p-values (default: `.Machine$double.eps`).
nsmall	Minimum number of digits to the right of the decimal point (default: 4).
prnt.lgnd	Character vector naming tables to print significance legends for (default: `"coefficients"`).
dig.test	Digits for hypothesis test statistics (default: `max(1, min(5, digits - 2))`).

Details

This method handles tabular output from linear model sumry.lm, including coefficients, ANOVA, statistics, and correlation/covariance matrices. It formats p-values, adds significance stars, and prints appropriate legends for hypothesis tests.

Value

Invisibly returns the input table.

See Also

print.sumry.lm

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width, data = iris)
sum_mdl <- sumry(mdl)
print(sum_mdl$coefficients)
```

sumry *Summary Descriptive Statistics for BAQM*

Description

sumry is a generic function for the BAQM package used to produce summaries of the results of certain model fitting functions.

Usage

```
sumry(x, ...)
```

Arguments

x	An object to summarize; methods are available for data frames, lists, vectors, linear regression models (lm), and best subsets models (regsubsets from the leaps package).
...	Additional arguments passed to specific methods.

Value

Invisibly returns a list of summary tables for each variable.

Examples

```
sumry(penguins)
sumry(data.frame(a = rnorm(100),
          b = c(NA, 1:98, NA),
          c = sample(letters[4:6], 100, TRUE)),
          transpose = TRUE, pad = 1)
sumry(lm(Sepal.Length ~ ., data = iris))
sumry(leaps::regsubsets(mpg ~ ., data = mtcars, nbest = 2))
```

sumry.default *Summary Descriptive Statistics for List or Data Frame*

Description

Computes and prints summary descriptive statistics for each variable in a data frame, list, or vector including counts, numeric summaries (min, quartiles, mean, max, standard deviation), and factor summaries (levels and frequencies).

Usage

```
## Default S3 method:
sumry(
  x,
  ...,
  transpose = FALSE,
  pad = 2,
  maxsum = 10,
  opts = list(digits = 4, scipen = 2)
)
```

Arguments

x	A data frame, list, or vector containing variables to summarize. A vector is treated as a single variable data frame. Unnamed variables receive generic names like V1, V2, etc.
...	Additional arguments (not currently used).
transpose	A logical controlling report format. By default the table printed is organized to show variables in columns and their statistic values in rows. Setting transpose = TRUE prints the transposed table with variables in rows and statistics in columns.
pad	A positive integer for the number of spaces between output columns.
maxsum	A positive integer limiting the number of levels for factor reports.
opts	A key=value type list, optional input for options values on output. Existing values are restored on exit.

Details

For each variable in x, the function computes the count of non-missing and missing values. Numeric variables are summarized by minimum, first quartile, median, mean, third quartile, maximum, and standard deviation. Factor, logical, and character variables are summarized by level frequencies, where level names may be abbreviated for readability. Results are formatted in a table and printed. The function returns a list containing numeric or factor summaries for each variable.

Value

Invisibly returns a list containing a summary data frame for each variable with no rounding. For example, smry <- sumry.df(df[c("A", "B", "C")]); smry$A$std.dev gives the standard deviation of variable A in data frame df

Examples

```
sumry(penguins)
sumry(data.frame(a = rnorm(100),
                 b = c(NA, 1:98, NA),
                 c = sample(letters[4:6], 100, TRUE)),
                 transpose = TRUE, pad = 1)
```

`sumry.lm` *Method* `sumry` *to Summarize Linear Model (*`lm`*) Objects*

Description

Computes a comprehensive summary for an object of class `lm`, including performance statistics, ANOVA, coefficients with VIFs, and correlation/covariance tables. Handles factor variable recoding and collinearity/singularity warnings.

Usage

```
## S3 method for class 'lm'
sumry(x, ...)
```

Arguments

`x`	An object of class `lm`.
`...`	Additional arguments (currently unused).

Details

The returned `sumry` object includes:

- **stats**: Performance statistics (F-statistic, R-squared, RMSE, etc.)
- **anova**: Simplified ANOVA table (Sum of squares, mean squares, F-statistic, p-value)
- **coefficients**: Table of regression coefficients with standard errors, t-stats, p-values, and VIFs
- **cov.unscaled**, **correlation**: Covariance and correlation matrices for coefficients
- **v.correlation**: Variable correlation matrix (for models with interaction terms)
- **fits**: Observed, fitted, and residual values
- **aliased**: Logical vector indicating aliased coefficients
- **df**: Degrees of freedom
- **sigma**: Estimated standard deviation of residuals
- **r.squared**, **adj.r.squared**: R-squared and adjusted R-squared
- **fstatistic**, **f.pval**: F-statistic and p-value for overall regression
- **notes**: Warnings, singularity, and collinearity notes (attached as attribute)

Factor variable names are recoded for clarity, and coefficients for aliased or singular variables are omitted with notes produced as attributes.

Value

An object of class `sumry.lm` containing tables and statistics described above.

See Also

print.sumry.lm, lm

Examples

```
mdl <- lm(Sepal.Length ~ Sepal.Width + Petal.Length, data = iris)
sumry(mdl)
```

---

sumry.regsubsets *Summary for Subset Selection (regsubsets) Objects*

Description

Generates a BAQM summary for objects of class `regsubsets` (from the `leaps` package), showing model selection statistics for best subsets, including R-squared, adjusted R-squared, standard error of estimate, Mallows' Cp, and AIC.

Usage

```
## S3 method for class 'regsubsets'
sumry(x, ...)
```

Arguments

`x`	An object of class `regsubsets`.
`...`	Additional arguments (not currently used).

Details

The function formats a table summarizing the best models selected, including the number of predictors, R-squared, adjusted R-squared, standard error of estimate (SEE), Mallows' Cp, and included variables. It is first converted to a summary.regsubsets with `summary()`. If not, the object is returned unmodified.

Value

Returns a matrix containing a summary table for the best subsets analysis. Each row summarizes a model showing: the number of predictors, `k`, used; "which best" that model is for that `k`; performance statistics (see above); and a series of columns with asterisks indicating the specific predictors included in the model.

See Also

regsubsets, print.sumry.regsubsets

Examples

```
fit <- leaps::regsubsets(Fertility ~ ., data = swiss, nbest = 3)
sumry(fit)
```

Index