Package ‘olsrr’

August 31, 2017

Type Package

Title Tools for Teaching and Learning OLS Regression

Version 0.3.0

Description Tools for teaching and learning ordinary least squares regression. Includes comprehensive regression output, heteroskedasticity tests, collinearity diagnostics, residual diagnostics, measures of influence, model fit assessment and variable selection procedures.

Depends R(>= 3.2.4)

Imports ggplot2, gridExtra, graphics, dplyr, magrittr, purrr, tibble, tidyr, nortest, goftest, Rcpp

Suggests testthat, covr, knitr, rmarkdown

License MIT + file LICENSE

URL https://rsquaredacademy.github.io/olsrr/, https://github.com/rsquaredacademy/olsrr

BugReports https://github.com/rsquaredacademy/olsrr/issues

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 6.0.1

LinkingTo Rcpp

NeedsCompilation yes

Author Aravind Hebbali [aut, cre]

Maintainer Aravind Hebbali <hebbali.aravind@gmail.com>

Repository CRAN

Date/Publication 2017-08-31 08:00:33 UTC

R topics documented:

`cement` *Test Data Set*

Description

Test Data Set

Usage

```
cement
```

Format

An object of class `data.frame` with 13 rows and 6 columns.

`fitness` *Test Data Set*

Description

Test Data Set

Usage

```
fitness
```

Format

An object of class `data.frame` with 31 rows and 7 columns.

hsb *Test Data Set*

Description

Test Data Set

Usage

```
hsb
```

Format

An object of class `data.frame` with 200 rows and 15 columns.

olsrr `olsrr` *package*

Description

Tools for teaching and learning OLS regression

Details

See the README on GitHub

ols_aic *AIC*

Description

Akaike Information Criterion

Usage

```
ols_aic(model, method = c("R", "STATA", "SAS"))
```

Arguments

`model`	an object of class `lm`
`method`	a character vector; specify the method to compute AIC. Valid options include R, STATA and SAS

Details

AIC provides a means for model selection. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. R and STATA use loglikelihood to compute AIC. SAS uses residual sum of squares. Below is the formula in each case:

R & STATA

$$AIC = -2(loglikelihood) + 2p$$

SAS

$$AIC = n * ln(SSE/n) + 2p$$

where *n* is the sample size and *p* is the number of model parameters including intercept.

Value

Akaike Information Criterion

References

Akaike, H. (1969). "Fitting Autoregressive Models for Prediction." Annals of the Institute of Statistical Mathematics 21:243–247.

Judge, G. G., Griffiths, W. E., Hill, R. C., and Lee, T.-C. (1980). The Theory and Practice of Econometrics. New York: John Wiley & Sons.

Examples

```
# using R computation method
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_aic(model)

# using STATA computation method
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_aic(model, method = 'STATA')

# using SAS computation method
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_aic(model, method = 'SAS')
```

`ols_all_subset` *All Possible Regression*

Description

Fits all regressions involving one regressor, two regressors, three regressors, and so on. It tests all possible subsets of the set of potential independent variables.

Usage

```
ols_all_subset(model, ...)

## S3 method for class 'ols_all_subset'
plot(x, model = NA, ...)
```

Arguments

`model`	an object of class `lm`
`...`	other arguments
`x`	an object of class `ols_best_subset`

Value

`ols_all_subset` returns an object of class `"ols_all_subset"`. An object of class `"ols_all_subset"` is a data frame containing the following components:

`n`	model number
`predictors`	predictors in the model
`rsquare`	rsquare of the model
`adjr`	adjusted rsquare of the model
`predrsq`	predicted rsquare of the model
`cp`	mallow's Cp
`aic`	akaike information criteria
`sbic`	sawa bayesian information criteria
`sbc`	schwarz bayes information criteria
`gmsep`	estimated MSE of prediction, assuming multivariate normality
`jp`	final prediction error
`pc`	amemiya prediction criteria
`sp`	hocking's Sp

References

Mendenhall William and Sinsich Terry, 2012, A Second Course in Statistics Regression Analysis (7th edition). Prentice Hall

Examples

```
model <- lm(mpg ~ disp + hp, data = mtcars)
k <- ols_all_subset(model)
k

# plot
plot(k)
```

ols_apc *Amemiya's Prediction Criterion*

Description

Amemiya's prediction error

Usage

```
ols_apc(model)
```

Arguments

model	an object of class lm

Details

Amemiya's Prediction Criterion penalizes R-squared more heavily than does adjusted R-squared for each addition degree of freedom used on the right-hand-side of the equation. The higher the better for this criterion.

$$((n+p)/(n-p))(1-(R^2))$$

where *n* is the sample size, *p* is the number of predictors including the intercept and *R^2* is the coefficient of determination.

Value

Amemiya's prediction error

References

Amemiya, T. (1976). Selection of Regressors. Technical Report 225, Stanford University, Stanford, CA.

Judge, G. G., Griffiths, W. E., Hill, R. C., and Lee, T.-C. (1980). The Theory and Practice of Econometrics. New York: John Wiley & Sons.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_apc(model)
```

ols_avplots *Added Variable Plot*

Description

Added variable plot provides information about the marginal importance of a predictor variable, given the other predictor variables already in the model. It shows the marginal importance of the variable in reducing the residual variability.

Usage

```
ols_avplots(model)
```

Arguments

model	an object of class lm

Details

The added variable plot was introduced by Mosteller and Tukey (1977). It enables us to visualize the regression coefficient of a new variable being considered to be included in a model. The plot can be constructed for each predictor variable.

Let us assume we want to test the effect of adding/removing variable X from a model. Let the response variable of the model be Y

Steps to construct an added variable plot:

- Regress Y on all variables other than X and store the residuals (Y residuals).
- Regress X on all the other variables included in the model (X residuals).
- Construct a scatter plot of Y residuals and X residuals.

What do the Y and X residuals represent? The Y residuals represent the part of **Y** not explained by all the variables other than X. The X residuals represent the part of **X** not explained by other variables. The slope of the line fitted to the points in the added variable plot is equal to the regression coefficient when **Y** is regressed on all variables including **X**.

A strong linear relationship in the added variable plot indicates the increased importance of the contribution of **X** to the model already containing the other predictors.

References

Chatterjee, Samprit and Hadi, Ali. Regression Analysis by Example. 5th ed. N.p.: John Wiley & Sons, 2012. Print.

Kutner, MH, Nachtscheim CJ, Neter J and Li W., 2004, Applied Linear Statistical Models (5th edition). Chicago, IL., McGraw Hill/Irwin.

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_avplots(model)
```

ols_bartlett_test *Bartlett Test*

Description

Test if k samples are from populations with equal variances.

Usage

```
ols_bartlett_test(variable, ...)

## Default S3 method:
ols_bartlett_test(variable, ..., group_var = NA)

## S3 method for class 'lm'
ols_bartlett_test(variable, ...)

## S3 method for class 'formula'
ols_bartlett_test(variable, data, ...)
```

Arguments

`variable`	a numeric vector/an object of class `formula` or `lm`
`...`	numeric vectors
`group_var`	grouping variable
`data`	a data frame

Details

Bartlett's test is used to test if variances across samples is equal. It is sensitive to departures from normality. The Levene test is an alternative test that is less sensitive to departures from normality.

Value

`ols_bartlett_test` returns an object of class `"ols_bartlett_test"`. An object of class `"ols_bartlett_test"` is a list containing the following components:

`fstat`	f statistic
`pval`	p-value of `fstat`
`df`	degrees of freedom
`var_c`	name(s) of `variable`
`g_var`	name of `group_var`

References

Snedecor, George W. and Cochran, William G. (1989), Statistical Methods, Eighth Edition, Iowa State University Press.

Examples

```
# using grouping variable
model <- lm(mpg ~ disp + hp, data = mtcars)
resid <- residuals(model)
cyl <- as.factor(mtcars$cyl)
ols_bartlett_test(resid, group_var = cyl)

# using variables
ols_bartlett_test(hsb$read, hsb$write)

# using formula
mt <- mtcars
mt$cyl <- as.factor(mt$cyl)
ols_bartlett_test(mpg ~ cyl, data = mt)

# using model
model <- lm(mpg ~ cyl, data = mt)
ols_bartlett_test(model)
```

---

ols_best_subset *Best Subsets Regression*

Description

Select the subset of predictors that do the best at meeting some well-defined objective criterion, such as having the largest R2 value or the smallest MSE, Mallow's Cp or AIC.

Usage

```
ols_best_subset(model, ...)

## S3 method for class 'ols_best_subset'
plot(x, model = NA, ...)
```

Arguments

model	an object of class lm
...	other inputs
x	an object of class ols_best_subset

Value

ols_best_subset returns an object of class "ols_best_subset". An object of class "ols_best_subset" is a data frame containing the following components:

n	model number
predictors	predictors in the model

rsquare	rsquare of the model
adjr	adjusted rsquare of the model
predrsq	predicted rsquare of the model
cp	mallow's Cp
aic	akaike information criteria
sbic	sawa bayesian information criteria
sbc	schwarz bayes information criteria
gmsep	estimated MSE of prediction, assuming multivariate normality
jp	final prediction error
pc	amemiya prediction criteria
sp	hocking's Sp

References

Kutner, MH, Nachtscheim CJ, Neter J and Li W., 2004, Applied Linear Statistical Models (5th edition). Chicago, IL., McGraw Hill/Irwin.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_best_subset(model)

# plot
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
k <- ols_best_subset(model)
plot(k)
```

---

ols_bp_test *Breusch Pagan Test*

Description

Test for constant variance. It assumes that the error terms are normally distributed.

Usage

```
ols_bp_test(model, fitted.values = TRUE, rhs = FALSE, multiple = FALSE,
  p.adj = c("none", "bonferroni", "sidak", "holm"), vars = NA)
```

Arguments

`model`	an object of class `lm`
`fitted.values`	logical; if TRUE, use fitted values of regression model
`rhs`	logical; if TRUE, specifies that tests for heteroskedasticity be performed for the right-hand-side (explanatory) variables of the fitted regression model
`multiple`	logical; if TRUE, specifies that multiple testing be performed
`p.adj`	p value adjustment, following options are available: bonferroni, holm, sidak and none
`vars`	variables to be used for for heteroskedasticity test

Details

Breusch Pagan Test was introduced by Trevor Breusch and Adrian Pagan in 1979. It is used to test for heteroskedasticity in a linear regression model. It test whether variance of errors from a regression is dependent on the values of a independent variable.

- Null Hypothesis: Equal/constant variances
- Alternative Hypothesis: Unequal/non-constant variances

Computation

- Fit a regression model
- Regress the squared residuals from the above model on the independent variables
- Compute nR^2. It follows a chi square distribution with p -1 degrees of freedom, where p is the number of independent variables, n is the sample size and R^2 is the coefficient of determination from the regression in step 2.

Value

`ols_bp_test` returns an object of class `"ols_bp_test"`. An object of class `"ols_bp_test"` is a list containing the following components:

`bp`	breusch pagan statistic
`p`	p-value of `bp`
`fv`	fitted values of the regression model
`rhs`	names of explanatory variables of fitted regression model
`multiple`	logical value indicating if multiple tests should be performed
`padj`	adjusted p values
`vars`	variables to be used for heteroskedasticity test
`resp`	response variable
`preds`	predictors

References

T.S. Breusch & A.R. Pagan (1979), A Simple Test for Heteroscedasticity and Random Coefficient Variation. Econometrica 47, 1287–1294

Cook, R. D.; Weisberg, S. (1983). "Diagnostics for Heteroskedasticity in Regression". Biometrika. 70 (1): 1–10.

Examples

```
# model
model <- lm(mpg ~ disp + hp + wt + drat, data = mtcars)

# Use fitted values of the model
ols_bp_test(model)

# Use independent variables of the model
ols_bp_test(model, rhs = TRUE)

# Use independent variables of the model and perform multiple tests
ols_bp_test(model, rhs = TRUE, multiple = TRUE)

# Bonferroni p value Adjustment
ols_bp_test(model, rhs = TRUE, multiple = TRUE, p.adj = 'bonferroni')

# Sidak p value Adjustment
ols_bp_test(model, rhs = TRUE, multiple = TRUE, p.adj = 'sidak')

# Holm's p value Adjustment
ols_bp_test(model, rhs = TRUE, multiple = TRUE, p.adj = 'holm')
```

`ols_coll_diag` *Collinearity Diagnostics*

Description

Variance inflation factor, tolerance, eigenvalues and condition indices.

Usage

```
ols_coll_diag(model)

ols_vif_tol(model)

ols_eigen_cindex(model)
```

Arguments

`model`	an object of class `lm`

Details

Collinearity implies two variables are near perfect linear combinations of one another. Multicollinearity involves more than two variables. In the presence of multicollinearity, regression estimates are unstable and have high standard errors.

Tolerance

Percent of variance in the predictor that cannot be accounted for by other predictors.

Steps to calculate tolerance:

- Regress the kth predictor on rest of the predictors in the model.
- Compute R^2 - the coefficient of determination from the regression in the above step.
- $Tolerance = 1 - R^2$

Variance Inflation Factor

Variance inflation factors measure the inflation in the variances of the parameter estimates due to collinearities that exist among the predictors. It is a measure of how much the variance of the estimated regression coefficient β_k is inflated by the existence of correlation among the predictor variables in the model. A VIF of 1 means that there is no correlation among the kth predictor and the remaining predictor variables, and hence the variance of β_k is not inflated at all. The general rule of thumb is that VIFs exceeding 4 warrant further investigation, while VIFs exceeding 10 are signs of serious multicollinearity requiring correction.

Steps to calculate VIF:

- Regress the kth predictor on rest of the predictors in the model.
- Compute R^2 - the coefficient of determination from the regression in the above step.
- $Tolerance = 1/1 - R^2 = 1/Tolerance$

Condition Index

Most multivariate statistical approaches involve decomposing a correlation matrix into linear combinations of variables. The linear combinations are chosen so that the first combination has the largest possible variance (subject to some restrictions), the second combination has the next largest variance, subject to being uncorrelated with the first, the third has the largest possible variance, subject to being uncorrelated with the first and second, and so forth. The variance of each of these linear combinations is called an eigenvalue. Collinearity is spotted by finding 2 or more variables that have large proportions of variance (.50 or more) that correspond to large condition indices. A rule of thumb is to label as large those condition indices in the range of 30 or larger.

Value

`ols_coll_diag` returns an object of class `"ols_coll_diag"`. An object of class `"ols_coll_diag"` is a list containing the following components:

`vif_t`	tolerance and variance inflation factors
`eig_cindex`	eigen values and condition index

References

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980). Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. New York: John Wiley & Sons.

Examples

```
# model
model <- lm(mpg ~ disp + hp + wt + drat, data = mtcars)

# vif and tolerance
ols_vif_tol(model)

# eigenvalues and condition indices
ols_eigen_cindex(model)

# collinearity diagnostics
ols_coll_diag(model)
```

---

ols_cooksd_barplot *Cooks' D Bar Plot*

Description

Bar Plot of cook's distance to detect observations that strongly influence fitted values of the model.

Usage

```
ols_cooksd_barplot(model)
```

Arguments

model	an object of class lm

Details

Cook's distance was introduced by American statistician R Dennis Cook in 1977. It is used to identify influential data points. It depends on both the residual and leverage i.e it takes it account both the *x* value and *y* value of the observation.

Steps to compute Cook's distance:

- Delete observations one at a time.
- Refit the regression model on remaining $n - 1$ observations
- examine how much all of the fitted values change when the ith observation is deleted.

A data point having a large cook's d indicates that the data point strongly influences the fitted values.

Value

`ols_cooksd_barplot` returns a list containing the following components:

`outliers`	a tibble with observation number and `cooks distance` that exceed `threshold`
`threshold`	`threshold` for classifying an observation as an outlier

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_cooksd_barplot(model)
```

`ols_cooksd_chart` *Cooks' D Chart*

Description

Chart of cook's distance to detect observations that strongly influence fitted values of the model.

Usage

```
ols_cooksd_chart(model)
```

Arguments

`model`	an object of class `lm`

Details

Cook's distance was introduced by American statistician R Dennis Cook in 1977. It is used to identify influential data points. It depends on both the residual and leverage i.e it takes it account both the *x* value and *y* value of the observation.

Steps to compute Cook's distance:

- Delete observations one at a time.
- Refit the regression model on remaining $n - 1$ observations
- exmine how much all of the fitted values change when the ith observation is deleted.

A data point having a large cook's d indicates that the data point strongly influences the fitted values.

Value

`ols_cooksd_chart` returns a list containing the following components:

`outliers`	a tibble with observation number and `cooks distance` that exceed `threshold`
`threshold`	`threshold` for classifying an observation as an outlier

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_cooksd_chart(model)
```

ols_correlations *Part and Partial Correlations*

Description

Zero-order, part and partial correlations

Usage

```
ols_correlations(model)
```

Arguments

`model`	an object of class `lm`

Details

`correlations` returns the relative importance of independent variables in determining response variable. How much each variable uniquely contributes to rsquare over and above that which can be accounted for by the other predictors? Zero order correlation is the Pearson correlation coefficient between the dependent variable and the independent variables. Part correlations indicates how much rsquare will decrease if that variable is removed from the model and partial correlations indicates amount of variance in response variable, which is not estimated by the other independent variables in the model, but is estimated by the specific variable.

Value

`ols_correlations` returns an object of class `"ols_correlations"`. An object of class `"ols_correlations"` is a data frame containing the following components:

`Zero-order`	zero order correlations
`Partial`	partial correlations
`Part`	part correlations

References

Morrison, D. F. 1976. Multivariate statistical methods. New York: McGraw-Hill.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_correlations(model)
```

ols_corr_test *Correlation Test For Normality*

Description

Correlation between observed residuals and expected residuals under normality.

Usage

```
ols_corr_test(model)
```

Arguments

model	an object of class lm

Value

correlation between fitted regression model residuals and expected values of residuals

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_corr_test(model)
```

ols_dfbetas_panel *DFBETAs Panel*

Description

Panel of plots to detect influential observations using DFBETAs.

Usage

```
ols_dfbetas_panel(model)
```

Arguments

model	an object of class lm

Details

DFBETA measures the difference in each parameter estimate with and without the influential point. There is a DFBETA for each data point i.e if there are n observations and k variables, there will be $n * k$ DFBETAs. In general, large values of DFBETAS indicate observations that are influential in estimating a given parameter. Belsley, Kuh, and Welsch recommend 2 as a general cutoff value to indicate influential observations and $2/\sqrt(n)$ as a size-adjusted cutoff.

Value

list; `ols_dfbetas_panel` returns a list of tibbles (for intercept and each predictor) with the observation number and DFBETA of observations that exceed the threshold for classifying an observation as an outlier/influential observation.

References

Belsley, David A.; Kuh, Edwin; Welsh, Roy E. (1980). Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons. pp. ISBN 0-471-05856-4.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_dfbetas_panel(model)
```

`ols_dffits_plot` *DFFITS Plot*

Description

Plot for detecting influential observations using DFFITs.

Usage

```
ols_dffits_plot(model)
```

Arguments

`model`	an object of class `lm`

Details

DFFIT - difference in fits, is used to identify influential data points. It quantifies the number of standard deviations that the fitted value changes when the ith data point is omitted.

Steps to compute DFFITs:

- Delete observations one at a time.
- Refit the regression model on remaining $n - 1$ observations
- examine how much all of the fitted values change when the ith observation is deleted.

An observation is deemed influential if the absolute value of its DFFITS value is greater than:

$$2\sqrt{(}p+1)/(n-p-1)$$

where n is the number of observations and p is the number of predictors including intercept.

Value

`ols_dffits_plot` returns a list containing the following components:

`outliers`	a tibble with observation number and `DFFITs` that exceed `threshold`
`threshold`	`threshold` for classifying an observation as an outlier

References

Belsley, David A.; Kuh, Edwin; Welsh, Roy E. (1980). Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons. ISBN 0-471-05856-4.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_dffits_plot(model)
```

`ols_diagnostic_panel` *Diagnostics Panel*

Description

Panel of plots for regression diagnostics

Usage

```
ols_diagnostic_panel(model)
```

Arguments

`model`	an object of class `lm`

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_diagnostic_panel(model)
```

ols_dsrvsp_plot *Deleted Studentized Residual vs Fitted Values Plot*

Description

Plot for detecting violation of assumptions about residuals such as non-linearity, constant variances and outliers. It can also be used to examine model fit.

Usage

```
ols_dsrvsp_plot(model)
```

Arguments

model	an object of class lm

Details

Studentized deleted residuals (or externally studentized residuals) is the deleted residual divided by its estimated standard deviation. Studentized residuals are going to be more effective for detecting outlying Y observations than standardized residuals. If an observation has an externally studentized residual that is larger than 2 (in absolute value) we can call it an outlier.

Value

ols_dsrvsp_plot returns a list containing the following components:

outliers	a tibble with observation number, fitted values and deleted studentized residuals that exceed the threshold for classifying observations as outliers/influential observations
threshold	threshold for classifying an observation as an outlier/influential observation

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_dsrvsp_plot(model)
```

`ols_fpe` *Final Prediction Error*

Description

Final prediction error

Usage

```
ols_fpe(model)
```

Arguments

`model`	an object of class `lm`

Details

Computes the estimated mean square error of prediction for each model selected assuming that the values of the regressors are fixed and that the model is correct.

$$MSE((n + p)/n)$$

where $MSE = SSE/(n - p)$, n is the sample size and p is the number of predictors including the intercept

Value

Final Prediction Error

References

Akaike, H. (1969). “Fitting Autoregressive Models for Prediction.” Annals of the Institute of Statistical Mathematics 21:243–247.

Judge, G. G., Griffiths, W. E., Hill, R. C., and Lee, T.-C. (1980). The Theory and Practice of Econometrics. New York: John Wiley & Sons.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_fpe(model)
```

ols_f_test *F Test for Constant Variance*

Description

Test for heteroskedasticity under the assumption that the errors are independent and identically distributed (i.i.d.).

Usage

```
ols_f_test(model, fitted_values = TRUE, rhs = FALSE, vars = NULL, ...)
```

Arguments

`model`	an object of class `lm`
`fitted_values`	logical; if TRUE, use fitted values of regression model
`rhs`	logical; if TRUE, specifies that tests for heteroskedasticity be performed for the right-hand-side (explanatory) variables of the fitted regression model
`vars`	variables to be used for for heteroskedasticity test
`...`	other arguments

Value

`ols_f_test` returns an object of class `"ols_f_test"`. An object of class `"ols_f_test"` is a list containing the following components:

`f`	f statistic
`p`	p-value of `f`
`fv`	fitted values of the regression model
`rhs`	names of explanatory variables of fitted regression model
`numdf`	numerator degrees of freedom
`dendf`	denominator degrees of freedom
`vars`	variables to be used for heteroskedasticity test
`resp`	response variable
`preds`	predictors

References

Wooldridge, J. M. 2013. Introductory Econometrics: A Modern Approach. 5th ed. Mason, OH: South-Western.

Examples

```
# model
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)

# using fitted values
ols_f_test(model)

# using all predictors of the model
ols_f_test(model, rhs = TRUE)

# using fitted values
ols_f_test(model, vars = c('disp', 'hp'))
```

---

ols_hadi *Hadi's Measure*

Description

Measure of influence based on the fact that influential observations in either the response variable or in the predictors or both.

Usage

```
ols_hadi(model)
```

Arguments

model	an object of class lm

Value

hadi's measure

References

Chatterjee, Samprit and Hadi, Ali. Regression Analysis by Example. 5th ed. N.p.: John Wiley & Sons, 2012. Print.

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_hadi(model)
```

`ols_hadi_plot` *Hadi Plot*

Description

Hadi's measure of influence based on the fact that influential observations can be present in either the response variable or in the predictors or both. The plot is used to detect influential observations based on Hadi's measure.

Usage

```
ols_hadi_plot(model)
```

Arguments

`model`	an object of class `lm`

References

Chatterjee, Samprit and Hadi, Ali. Regression Analysis by Example. 5th ed. N.p.: John Wiley & Sons, 2012. Print.

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_hadi_plot(model)
```

`ols_hsp` *Hocking's Sp*

Description

Average prediction mean squared error

Usage

```
ols_hsp(model)
```

Arguments

`model`	an object of class `lm`

Details

Hocking's Sp criterion is an adjustment of the residual sum of Squares. Minimize this criterion.

$$MSE/(n - p - 1)$$

where $MSE = SSE/(n - p)$, n is the sample size and p is the number of predictors including the intercept

Value

Hocking's Sp

References

Hocking, R. R. (1976). "The Analysis and Selection of Variables in a Linear Regression." Biometrics 32:1–50.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_hsp(model)
```

---

ols_leverage *Leverage*

Description

The leverage of an observation is based on how much the observation's value on the predictor variable differs from the mean of the predictor variable. The greater an observation's leverage, the more potential it has to be an influential observation.

Usage

```
ols_leverage(model)
```

Arguments

`model`	an object of class `lm`

Value

leverage

References

Kutner, MH, Nachtscheim CJ, Neter J and Li W., 2004, Applied Linear Statistical Models (5th edition). Chicago, IL., McGraw Hill/Irwin.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_leverage(model)
```

`ols_mallows_cp` *Mallow's Cp*

Description

Mallow's Cp

Usage

```
ols_mallows_cp(model, fullmodel)
```

Arguments

`model`	an object of class `lm`
`fullmodel`	an object of class `lm`

Details

Mallows' Cp statistic estimates the size of the bias that is introduced into the predicted responses by having an underspecified model. Use Mallows' Cp to choose between multiple regression models. Look for models where Mallows' Cp is small and close to the number of predictors in the model plus the constant (p).

Value

Mallow's Cp

References

Hocking, R. R. (1976). "The Analysis and Selection of Variables in a Linear Regression." Biometrics 32:1–50.

Mallows, C. L. (1973). "Some Comments on Cp." Technometrics 15:661–675.

Examples

```
full_model <- lm(mpg ~ ., data = mtcars)
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_mallows_cp(model, full_model)
```

`ols_msep` *MSEP*

Description

Estimated error of prediction, assuming multivariate normality

Usage

```
ols_msep(model)
```

Arguments

`model`	an object of class `lm`

Details

Computes the estimated mean square error of prediction assuming that both independent and dependent variables are multivariate normal.

$$MSE(n+1)(n-2)/n(n-p-1)$$

where $MSE = SSE/(n-p)$, n is the sample size and p is the number of predictors including the intercept

Value

MSEP

References

Stein, C. (1960). "Multiple Regression." In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling, edited by I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow, and H. B. Mann, 264–305. Stanford, CA: Stanford University Press.

Darlington, R. B. (1968). "Multiple Regression in Psychological Research and Practice." Psychological Bulletin 69:161–182.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_msep(model)
```

ols_norm_test *Test for normality*

Description

Test for detecting violation of normality assumption.

Usage

```
ols_norm_test(y, ...)

## S3 method for class 'lm'
ols_norm_test(y, ...)
```

Arguments

y	a numeric vector or an object of class `lm`
...	other arguments

Value

`norm_test` returns an object of class `"norm_test"`. An object of class `"norm_test"` is a list containing the following components:

kolmogorv	kolmogorv smirnov statistic
shapiro	shapiro wilk statistic
cramer	cramer von mises statistic
anderson	anderson darling statistic

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_norm_test(model)
```

ols_ovsp_plot *Actual vs Fitted Values Plot*

Description

Plot of actual vs fitted values to assess the fit of the model.

Usage

```
ols_ovsp_plot(model)
```

Arguments

`model`	an object of class `lm`

Details

Ideally, all your points should be close to a regressed diagonal line. Draw such a diagonal line within your graph and check out where the points lie. If your model had a high R Square, all the points would be close to this diagonal line. The lower the R Square, the weaker the Goodness of fit of your model, the more foggy or dispersed your points are from this diagonal line.

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_ovsp_plot(model)
```

`ols_potrsd_plot` *Potential Residual Plot*

Description

Plot to aid in classifying unusual observations as high-leverage points, outliers, or a combination of both.

Usage

```
ols_potrsd_plot(model)
```

Arguments

`model`	an object of class `lm`

References

Chatterjee, Samprit and Hadi, Ali. Regression Analysis by Example. 5th ed. N.p.: John Wiley & Sons, 2012. Print.

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_potrsd_plot(model)
```

ols_pred_rsq *Predicted Rsquare*

Description

Use predicted rsquared to determine how well the model predicts responses for new observations. Larger values of predicted R2 indicate models of greater predictive ability.

Usage

```
ols_pred_rsq(model)
```

Arguments

`model`	an object of class `lm`

Value

Predicted Rsquare

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_pred_rsq(model)
```

ols_press *PRESS (Prediction Sum of Squares)*

Description

PRESS tells you how well the model will predict new data.

Usage

```
ols_press(model)
```

Arguments

`model`	an object of class `lm`

Details

The prediction sum of squares (PRESS) is the sum of squares of the prediction error. Each fitted value for PRESS is obtained from the remaining n – 1 observations, then using the fitted regression function to obtain the predicted value for the ith observation. Use PRESS to assess your model's predictive ability. Usually, the smaller the PRESS value, the better the model's predictive ability.

Value

Predicted Sum of Squares

References

Kutner, MH, Nachtscheim CJ, Neter J and Li W., 2004, Applied Linear Statistical Models (5th edition). Chicago, IL., McGraw Hill/Irwin.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_press(model)
```

`ols_pure_error_anova` *Lack of Fit F Test*

Description

Assess how much of the error in prediction is due to lack of model fit.

Usage

```
ols_pure_error_anova(model, ...)
```

Arguments

`model`	an object of class `lm`
`...`	other parameters

Details

The residual sum of squares resulting from a regression can be decomposed into 2 components:

- Due to lack of fit
- Due to random variation

If most of the error is due to lack of fit and not just random error, the model should be discarded and a new model must be built.

Value

`ols_pure_error_anova` returns an object of class `"ols_pure_error_anova"`. An object of class `"ols_pure_error_anova"` is a list containing the following components:

`lackoffit`	lack of fit sum of squares
`pure_error`	pure error sum of squares
`rss`	regression sum of squares

`ess`	error sum of squares
`total`	total sum of squares
`rms`	regression mean square
`ems`	error mean square
`lms`	lack of fit mean square
`pms`	pure error mean square
`rf`	f statistic
`lf`	lack of fit f statistic
`pr`	p-value of f statistic
`pl`	p-value pf lack of fit f statistic
`mpred`	tibble containing data for the response and predictor of the `model`
`df_rss`	regression sum of squares degrees of freedom
`df_ess`	error sum of squares degrees of freedom
`df_lof`	lack of fit degrees of freedom
`df_error`	pure error degrees of freedom
`final`	data.frame; contains computed values used for the lack of fit f test
`resp`	character vector; name of `response variable`
`preds`	character vector; name of `predictor variable`

Note

The lack of fit F test works only with simple linear regression. Moreover, it is important that the data contains repeat observations i.e. replicates for at least one of the values of the predictor x. This test generally only applies to datasets with plenty of replicates.

References

Kutner, MH, Nachtscheim CJ, Neter J and Li W., 2004, Applied Linear Statistical Models (5th edition). Chicago, IL., McGraw Hill/Irwin.

Examples

```
model <- lm(mpg ~ disp, data = mtcars)
ols_pure_error_anova(model)
```

`ols_regress` *Ordinary Least Squares Regression*

Description

Ordinary Least Squares Regression

Usage

```
ols_regress(object, ...)

## S3 method for class 'lm'
ols_regress(object, ...)
```

Arguments

`object`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted or class `lm`
`...`	other inputs

Value

`ols_regress` returns an object of class `"ols_regress"`. An object of class `"ols_regress"` is a list containing the following components:

`r`	square root of rsquare, correlation between observed and predicted values of dependent variable
`rsq`	coefficient of determination or r-square
`adjr`	adjusted rsquare
`sigma`	root mean squared error
`cv`	coefficient of variation
`mse`	mean squared error
`mae`	mean absolute error
`aic`	akaike information criteria
`sbc`	bayesian information criteria
`sbic`	sawa bayesian information criteria
`prsq`	predicted rsquare
`error_df`	residual degrees of freedom
`model_df`	regression degrees of freedom
`total_df`	total degrees of freedom
`ess`	error sum of squares
`rss`	regression sum of squares

`tss`	total sum of squares
`rms`	regression mean square
`ems`	error mean square
`f`	f statistis
`p`	p-value for `f`
`n`	number of predictors including intercept
`betas`	betas; estimated coefficients
`sbetas`	standardized betas
`std_errors`	standard errors
`tvalues`	t values
`pvalues`	p-value of `tvalues`
`df`	degrees of freedom of `betas`
`conf_lm`	confidence intervals for coefficients
`title`	title for the model
`dependent`	character vector; name of the dependent variable
`predictors`	character vector; name of the predictor variables
`mvars`	character vector; name of the predictor variables including intercept
`model`	input model for `ols_regress`

Examples

```
ols_regress(mpg ~ disp + hp + wt, data = mtcars)
```

`ols_reg_line` *Simple Linear Regression Line*

Description

Regression line always passes through xbar and ybar

Usage

```
ols_reg_line(response, predictor)
```

Arguments

`response`	response variable
`predictor`	explanatory variable

Examples

```
ols_reg_line(mtcars$mpg, mtcars$disp)
```

ols_resp_viz *Visualize Response Variable*

Description

Panel of plots to explore and visualize the response variable.

Usage

```
ols_resp_viz(model)
```

Arguments

`model`	an object of class `lm`

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_resp_viz(model)
```

ols_rfs_plot *Residual Fit Spread Plot*

Description

Plot to detect non-linearity, influential observations and outliers.

Usage

```
ols_rfs_plot(model)

ols_fm_plot(model)

ols_rsd_plot(model)
```

Arguments

`model`	an object of class `lm`

Details

Consists of side-by-side quantile plots of the centered fit and the residuals. It shows how much variation in the data is explained by the fit and how much remains in the residuals. For inappropriate models, the spread of the residuals in such a plot is often greater than the spread of the centered fit.

References

Cleveland, W. S. (1993). Visualizing Data. Summit, NJ: Hobart Press.

Examples

```
# model
model <- lm(mpg ~ disp + hp + wt, data = mtcars)

# residual fit spread plot
ols_rfs_plot(model)

# fit mean plot
ols_fm_plot(model)

# residual spread plot
ols_rsd_plot(model)
```

`ols_rpc_plot` *Residual Plus Component Plot*

Description

The residual plus component plot indicates whether any non-linearity is present in the relationship between response and predictor variables and can suggest possible transformations for linearizing the data.

Usage

```
ols_rpc_plot(model)
```

Arguments

`model`	an object of class `lm`

References

Chatterjee, Samprit and Hadi, Ali. Regression Analysis by Example. 5th ed. N.p.: John Wiley & Sons, 2012. Print. Kutner, MH, Nachtscheim CJ, Neter J and Li W., 2004, Applied Linear Statistical Models (5th edition). Chicago, IL., McGraw Hill/Irwin.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_rpc_plot(model)
```

ols_rsdlev_plot *Studentized Residuals vs Leverage Plot*

Description

Graph for detecting outliers and/or observations with high leverage.

Usage

```
ols_rsdlev_plot(model)
```

Arguments

model	an object of class lm

Examples

```
model <- lm(read ~ write + math + science, data = hsb)
ols_rsdlev_plot(model)
```

ols_rsd_boxplot *Residual Box Plot*

Description

Box plot of residuals to examine if residuals are normally distributed.

Usage

```
ols_rsd_boxplot(model)
```

Arguments

model	an object of class lm model <- lm(mpg ~ disp + hp + wt, data = mtcars) ols_rsd_boxplot(model)

ols_rsd_hist *Residual Histogram*

Description

Histogram of residuals for detecting violation of normality assumption.

Usage

```
ols_rsd_hist(model)
```

Arguments

`model`	an object of class `lm`

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_rsd_hist(model)
```

ols_rsd_qqplot *Residual QQ Plot*

Description

Graph for detecting violation of normality assumption.

Usage

```
ols_rsd_qqplot(model)
```

Arguments

`model`	an object of class `lm`

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_rsd_qqplot(model)
```

ols_rvsp_plot *Residual vs Fitted Plot*

Description

It is a scatter plot of residuals on the y axis and fitted values on the x axis to detect non-linearity, unequal error variances, and outliers.

Usage

```
ols_rvsp_plot(model)
```

Arguments

`model`	an object of class `lm`

Details

Characteristics of a well behaved residual vs fitted plot:

- The residuals spread randomly around the 0 line indicating that the relationship is linear.
- The residuals form an approximate horizontal band around the 0 line indicating homogeneity of error variance.
- No one residual is visibly away from the random pattern of the residuals indicating that there are no outliers.

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_rvsp_plot(model)
```

ols_rvsr_plot *Residual vs Regressors Plot*

Description

Graph to determine whether we should add a new predictor to the model already containing other predictors. The residuals from the model is regressed on the new predictor and if the plot shows non random pattern, you should consider adding the new predictor to the model.

Usage

```
ols_rvsr_plot(model, variable)
```

Arguments

`model`	an object of class `lm`
`variable`	new predictor to be added to the `model`

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_rvsr_plot(model, mtcars$drat)
```

`ols_sbc` *SBC*

Description

Bayesian Information Criterion

Usage

```
ols_sbc(model, method = c("R", "STATA", "SAS"))
```

Arguments

`model`	an object of class `lm`
`method`	a character vector; specify the method to compute BIC. Valid options include R, STATA and SAS

Details

SBC provides a means for model selection. Given a collection of models for the data, SBC estimates the quality of each model, relative to each of the other models. R and STATA use loglikelihood to compute SBC. SAS uses residual sum of squares. Below is the formula in each case:

R & STATA

$$AIC = -2(loglikelihood) + ln(n) * 2p$$

SAS

$$AIC = n * ln(SSE/n) + p * ln(n)$$

where *n* is the sample size and *p* is the number of model parameters including intercept.

Value

Bayesian Information Criterion

References

Schwarz, G. (1978). “Estimating the Dimension of a Model.” Annals of Statistics 6:461–464.

Judge, G. G., Griffiths, W. E., Hill, R. C., and Lee, T.-C. (1980). The Theory and Practice of Econometrics. New York: John Wiley & Sons.

Examples

```
# using R computation method
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_sbc(model)

# using STATA computation method
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_sbc(model, method = 'STATA')

# using SAS computation method
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_sbc(model, method = 'SAS')
```

`ols_sbic` *SBIC*

Description

Sawa's Bayesian Information Criterion

Usage

```
ols_sbic(model, full_model)
```

Arguments

`model`	an object of class `lm`
`full_model`	an object of class `lm`

Details

Sawa (1978) developed a model selection criterion that was derived from a Bayesian modification of the AIC criterion. Sawa's Bayesian Information Criteria (BIC) is a function of the number of observations n, the SSE, the pure error variance fitting the full model, and the number of independent variables including the intercept.

$$SBIC = n * ln(SSE/n) + 2(p + 2)q - 2(q^2)$$

where $q = n(\sigma^2)/SSE$, n is the sample size, p is the number of model parameters including intercept *SSE* is the residual sum of squares.

Value

Sawa's Bayesian Information Criterion

References

Sawa, T. (1978). “Information Criteria for Discriminating among Alternative Regression Models.” Econometrica 46:1273–1282.

Judge, G. G., Griffiths, W. E., Hill, R. C., and Lee, T.-C. (1980). The Theory and Practice of Econometrics. New York: John Wiley & Sons.

Examples

```
full_model <- lm(mpg ~ ., data = mtcars)
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_sbic(model, full_model)
```

`ols_score_test` *Score Test for heteroskedasticity*

Description

Test for heteroskedasticity under the assumption that the errors are independent and identically distributed (i.i.d.).

Usage

```
ols_score_test(model, fitted_values = TRUE, rhs = FALSE, vars = NULL)
```

Arguments

`model`	an object of class `lm`
`fitted_values`	logical; if TRUE, use fitted values of regression model
`rhs`	logical; if TRUE, specifies that tests for heteroskedasticity be performed for the right-hand-side (explanatory) variables of the fitted regression model
`vars`	variables to be used for for heteroskedasticity test

Value

`ols_score_test` returns an object of class `"ols_score_test"`. An object of class `"ols_score_test"` is a list containing the following components:

`score`	f statistic
`p`	p value of `score`
`df`	degrees of freedom
`fv`	fitted values of the regression model
`rhs`	names of explanatory variables of fitted regression model
`resp`	response variable
`preds`	predictors

References

Breusch, T. S. and Pagan, A. R. (1979) A simple test for heteroscedasticity and random coefficient variation. Econometrica 47, 1287–1294.

Cook, R. D. and Weisberg, S. (1983) Diagnostics for heteroscedasticity in regression. Biometrika 70, 1–10.

Koenker, R. 1981. A note on studentizing a test for heteroskedasticity. Journal of Econometrics 17: 107–112.

Examples

```
# model
model <- lm(mpg ~ disp + hp + wt, data = mtcars)

# using fitted values of the model
ols_score_test(model)

# using predictors from the model
ols_score_test(model, rhs = TRUE)

# specify predictors from the model
ols_score_test(model, vars = c('disp', 'wt'))
```

---

ols_srsd_chart *Standardized Residual Chart*

Description

Chart for identifying outliers

Usage

```
ols_srsd_chart(model)
```

Arguments

model	an object of class lm

Details

Standardized residual (internally studentized) is the residual divided by estimated standard deviation.

Value

`ols_srsd_chart` returns a list containing the following components:

`outliers` a tibble with observation number and `standardized resiudals` that exceed `threshold`

for classifying an observation as an outlier

`threshold` `threshold` for classifying an observation as an outlier

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_srsd_chart(model)
```

`ols_srsd_plot` *Studentized Residual Plot*

Description

Graph for identifying outliers

Usage

```
ols_srsd_plot(model)
```

Arguments

`model` an object of class `lm`

Details

Studentized deleted residuals (or externally studentized residuals) is the deleted residual divided by its estimated standard deviation. Studentized residuals are going to be more effective for detecting outlying Y observations than standardized residuals. If an observation has an externally studentized residual that is larger than 3 (in absolute value) we can call it an outlier.

Value

`ols_srsd_plot` returns a list containing the following components:

`outliers` a tibble with observation number and `studentized residuals` that exceed `threshold`

for classifying an observation as an outlier

`threshold` `threshold` for classifying an observation as an outlier

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_srsd_plot(model)
```

ols_stepaic_backward *Stepwise AIC Backward Regression*

Description

Build regression model from a set of candidate predictor variables by removing predictors based on Akaike Information Criteria, in a stepwise manner until there is no variable left to remove any more.

Usage

```
ols_stepaic_backward(model, ...)

## Default S3 method:
ols_stepaic_backward(model, details = FALSE, ...)

## S3 method for class 'ols_stepaic_backward'
plot(x, ...)
```

Arguments

`model`	an object of class `lm`; the model should include all candidate predictor variables
`...`	other arguments
`details`	logical; if `TRUE`, will print the regression result at each step
`x`	an object of class `ols_stepaic_backward`

Value

`ols_stepaic_backward` returns an object of class `"ols_stepaic_backward"`. An object of class `"ols_stepaic_backward"` is a list containing the following components:

`steps`	total number of steps
`predictors`	variables removed from the model
`aics`	akaike information criteria
`ess`	error sum of squares
`rss`	regression sum of squares
`rsq`	rsquare
`arsq`	adjusted rsquare

References

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

Examples

```
# stepwise backward regression
model <- lm(y ~ ., data = surgical)
ols_stepaic_backward(model)

# stepwise backward regression plot
model <- lm(y ~ ., data = surgical)
k <- ols_stepaic_backward(model)
plot(k)
```

`ols_stepaic_both` *Stepwise AIC Regression*

Description

Build regression model from a set of candidate predictor variables by entering and removing predictors based on Akaike Information Criteria, in a stepwise manner until there is no variable left to enter or remove any more.

Usage

```
ols_stepaic_both(model, details = FALSE)

## S3 method for class 'ols_stepaic_both'
plot(x, ...)
```

Arguments

`model`	an object of class `lm`
`details`	logical; if `TRUE`, details of variable selection will be printed on screen
`x`	an object of class `ols_stepaic_both`
`...`	other arguments

Value

`ols_stepaic_both` returns an object of class `"ols_stepaic_both"`. An object of class `"ols_stepaic_both"` is a list containing the following components:

`predictors`	variables added/removed from the model
`method`	addition/deletion
`aics`	akaike information criteria
`ess`	error sum of squares
`rss`	regression sum of squares
`rsq`	rsquare
`arsq`	adjusted rsquare
`steps`	total number of steps

References

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

Examples

```
# stepwise regression
model <- lm(y ~ ., data = surgical)
ols_stepaic_both(model)

# stepwise regression plot
model <- lm(y ~ ., data = surgical)
k <- ols_stepaic_both(model)
plot(k)
```

---

ols_stepaic_forward *Stepwise AIC Forward Regression*

Description

Build regression model from a set of candidate predictor variables by entering predictors based on Akaike Information Criteria, in a stepwise manner until there is no variable left to enter any more.

Usage

```
ols_stepaic_forward(model, ...)

## Default S3 method:
ols_stepaic_forward(model, details = FALSE, ...)

## S3 method for class 'ols_stepaic_forward'
plot(x, ...)
```

Arguments

`model`	an object of class `lm`
`...`	other arguments
`details`	logical; if `TRUE`, will print the regression result at each step
`x`	an object of class `ols_stepaic_forward`

Value

`ols_stepaic_forward` returns an object of class `"ols_stepaic_forward"`. An object of class `"ols_stepaic_forward"` is a list containing the following components:

`steps`	total number of steps
`predictors`	variables added to the model

aics	akaike information criteria
ess	error sum of squares
rss	regression sum of squares
rsq	rsquare
arsq	adjusted rsquare

References

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

Examples

```
# stepwise forward regression
model <- lm(y ~ ., data = surgical)
ols_stepaic_forward(model)

# stepwise forward regression plot
model <- lm(y ~ ., data = surgical)
k <- ols_stepaic_forward(model)
plot(k)
```

---

ols_stepwise *Stepwise Regression*

Description

Build regression model from a set of candidate predictor variables by entering and removing predictors based on p values, in a stepwise manner until there is no variable left to enter or remove any more.

Usage

```
ols_stepwise(model, ...)

## Default S3 method:
ols_stepwise(model, pent = 0.1, prem = 0.3,
  details = FALSE, ...)

## S3 method for class 'ols_stepwise'
plot(x, model = NA, ...)
```

Arguments

`model`	an object of class `lm`; the model should include all candidate predictor variables
`...`	other arguments
`pent`	p value; variables with p value less than `pent` will enter into the model
`prem`	p value; variables with p more than `prem` will be removed from the model
`details`	logical; if `TRUE`, will print the regression result at each step
`x`	an object of class `ols_stepwise`

Value

`ols_stepwise` returns an object of class `"ols_stepwise"`. An object of class `"ols_stepwise"` is a list containing the following components:

`orders`	candidate predictor variables according to the order by which they were added or removed from the model
`method`	addition/deletion
`steps`	total number of steps
`predictors`	variables retained in the model (after addition)
`rsquare`	coefficient of determination
`aic`	akaike information criteria
`sbc`	bayesian information criteria
`sbic`	sawa's bayesian information criteria
`adjr`	adjusted r-square
`rmse`	root mean square error
`mallows_cp`	mallow's Cp
`indvar`	predictors

References

Chatterjee, Samprit and Hadi, Ali. Regression Analysis by Example. 5th ed. N.p.: John Wiley & Sons, 2012. Print.

Examples

```
# stepwise regression
model <- lm(y ~ ., data = surgical)
ols_stepwise(model)

# stepwise regression plot
model <- lm(y ~ ., data = surgical)
k <- ols_stepwise(model)
plot(k)
```

ols_step_backward *Stepwise Backward Regression*

Description

Build regression model from a set of candidate predictor variables by removing predictors based on p values, in a stepwise manner until there is no variable left to remove any more.

Usage

```
ols_step_backward(model, ...)

## Default S3 method:
ols_step_backward(model, prem = 0.3, details = FALSE, ...)

## S3 method for class 'ols_step_backward'
plot(x, model = NA, ...)
```

Arguments

`model`	an object of class `lm`; the model should include all candidate predictor variables
`...`	other inputs
`prem`	p value; variables with p more than `prem` will be removed from the model
`details`	logical; if `TRUE`, will print the regression result at each step
`x`	an object of class `ols_step_backward`

Value

`ols_step_backward` returns an object of class `"ols_step_backward"`. An object of class `"ols_step_backward"` is a list containing the following components:

`steps`	total number of steps
`removed`	variables removed from the model
`rsquare`	coefficient of determination
`aic`	akaike information criteria
`sbc`	bayesian information criteria
`sbic`	sawa's bayesian information criteria
`adjr`	adjusted r-square
`rmse`	root mean square error
`mallows_cp`	mallow's Cp
`indvar`	predictors

References

Chatterjee, Samprit and Hadi, Ali. Regression Analysis by Example. 5th ed. N.p.: John Wiley & Sons, 2012. Print.

Examples

```
# stepwise backward regression
model <- lm(y ~ ., data = surgical)
ols_step_backward(model)

# stepwise backward regression plot
model <- lm(y ~ ., data = surgical)
k <- ols_step_backward(model)
plot(k)
```

`ols_step_forward` *Stepwise Forward Regression*

Description

Build regression model from a set of candidate predictor variables by entering predictors based on p values, in a stepwise manner until there is no variable left to enter any more.

Usage

```
ols_step_forward(model, ...)

## Default S3 method:
ols_step_forward(model, penter = 0.3, details = FALSE,
  ...)

## S3 method for class 'ols_step_forward'
plot(x, model = NA, ...)
```

Arguments

`model`	an object of class `lm`; the model should include all candidate predictor variables
`...`	other arguments
`penter`	p value; variables with p value less than `penter` will enter into the model
`details`	logical; if `TRUE`, will print the regression result at each step
`x`	an object of class `ols_step_forward`

Value

ols_step_forward returns an object of class "ols_step_forward". An object of class "ols_step_forward" is a list containing the following components:

steps	number of steps
predictors	variables added to the model
rsquare	coefficient of determination
aic	akaike information criteria
sbc	bayesian information criteria
sbic	sawa's bayesian information criteria
adjr	adjusted r-square
rmse	root mean square error
mallows_cp	mallow's Cp
indvar	predictors

References

Chatterjee, Samprit and Hadi, Ali. Regression Analysis by Example. 5th ed. N.p.: John Wiley & Sons, 2012. Print.

Kutner, MH, Nachtscheim CJ, Neter J and Li W., 2004, Applied Linear Statistical Models (5th edition). Chicago, IL., McGraw Hill/Irwin.

Examples

```
# stepwise forward regression
model <- lm(y ~ ., data = surgical)
ols_step_forward(model)

# stepwise forward regression plot
model <- lm(y ~ ., data = surgical)
k <- ols_step_forward(model)
plot(k)
```

---

rivers *Test Data Set*

Description

Test Data Set

Usage

```
rivers
```

Format

An object of class `data.frame` with 20 rows and 6 columns.

`rvsr_plot_shiny` *Residual vs Regressors Plot Shiny*

Description

Graph to determine whether we should add a new predictor to the model already containing other predictors. The residuals from the model is regressed on the new predictor and if the plot shows non random pattern, you should consider adding the new predictor to the model.

Usage

```
rvsr_plot_shiny(model, data, variable)
```

Arguments

`model`	an object of class `lm`
`data`	dataframe
`variable`	character; new predictor to be added to the `model`

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
rvsr_plot_shiny(model, mtcars, 'drat')
```

`surgical` *Surgical Unit Data Set*

Description

A dataset containing data about survival of patients undergoing liver operation.

Usage

```
surgical
```

Format

A data frame with 54 rows and 9 variables:

bcs blood clotting score

pindex prognostic index

enzyme_test enzyme function test score

liver_test liver function test score

age age, in years

gender indicator variable for gender (0 = male, 1 = female)

alc_mod indicator variable for history of alcohol use (0 = None, 1 = Moderate)

alc_heavy indicator variable for history of alcohol use (0 = None, 1 = Heavy)

y Survival Time

Source

Kutner, MH, Nachtscheim CJ, Neter J and Li W., 2004, Applied Linear Statistical Models (5th edition). Chicago, IL., McGraw Hill/Irwin.

Index