Package 'olsrr'

November 6, 2024

Type Package

Title Tools for Building OLS Regression Models

Version 0.6.1

Description Tools designed to make it easier for users, particularly beginner/intermediate R users to build ordinary least squares regression models. Includes comprehensive regression output, heteroskedasticity tests, collinearity diagnostics, residual diagnostics, measures of influence, model fit assessment and variable selection procedures.

Depends R(>= 3.3)

Imports car, ggplot2, goftest, graphics, gridExtra, nortest, stats, utils, xplorerr

Suggests covr, descriptr, knitr, rmarkdown, testthat, vdiffr

License MIT + file LICENSE

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

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

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Author Aravind Hebbali [aut, cre]

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

Repository CRAN

Date/Publication 2024-11-06 12:50:06 UTC

Contents

ols_aic *Akaike information criterion*

Description

Akaike information criterion for model selection.

Usage

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

Arguments

`model`	An object of class `lm`.
`method`	A character vector; specify the method to compute AIC. Valid options include R, STATA and SAS.
`corrected`	Logical; if `TRUE`, returns corrected akaike information criterion for SAS method.

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

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

corrected

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

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

Value

Akaike information criterion of the model.

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.

See Also

Other model selection criteria: ols_apc(), ols_fpe(), ols_hsp(), ols_mallows_cp(), ols_msep(), ols_sbc(), ols_sbic()

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')

# corrected akaike information criterion
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_aic(model, method = 'SAS', corrected = TRUE)
```

---

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 lower 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 of the model.

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.

See Also

Other model selection criteria: `ols_aic()`, `ols_fpe()`, `ols_hsp()`, `ols_mallows_cp()`, `ols_msep()`, `ols_sbc()`, `ols_sbic()`

Examples

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

`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_correlations *Part and partial correlations*

Description

Zero-order, part and partial correlations.

Usage

```
ols_correlations(model)
```

Arguments

`model`	An object of class `lm`.

Details

`ols_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_fpe *Final prediction error*

Description

Estimated mean square error of prediction.

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 of the model.

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.

See Also

Other model selection criteria: ols_aic(), ols_apc(), ols_hsp(), ols_mallows_cp(), ols_msep(), ols_sbc(), ols_sbic()

Examples

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

---

ols_hadi — *Hadi's influence 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 of the model.

References

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

See Also

Other influence measures: `ols_leverage()`, `ols_pred_rsq()`, `ols_press()`

Examples

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

References

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

See Also

Other model selection criteria: ols_aic(), ols_apc(), ols_fpe(), ols_mallows_cp(), ols_msep(), ols_sbc(), ols_sbic()

Examples

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

`ols_launch_app` *Launch shiny app*

Description

Launches shiny app for interactive model building.

Usage

```
ols_launch_app()
```

Examples

```
## Not run:
ols_launch_app()

## End(Not run)
```

`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 of the model.

References

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

See Also

Other influence measures: `ols_hadi()`, `ols_pred_rsq()`, `ols_press()`

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 of the model.

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.

See Also

Other model selection criteria: `ols_aic()`, `ols_apc()`, `ols_fpe()`, `ols_hsp()`, `ols_msep()`, `ols_sbc()`, `ols_sbic()`

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

Estimated error of prediction of the model.

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.

See Also

Other model selection criteria: `ols_aic()`, `ols_apc()`, `ols_fpe()`, `ols_hsp()`, `ols_mallows_cp()`, `ols_sbc()`, `ols_sbic()`

Examples

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

ols_plot_added_variable
Added variable plots

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_plot_added_variable(model, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

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.

See Also

ols_plot_resid_regressor(), ols_plot_comp_plus_resid()

Examples

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

---

ols_plot_comp_plus_resid

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_plot_comp_plus_resid(model, print_plot = TRUE)
```

Arguments

model	An object of class `lm`.
print_plot	logical; if `TRUE`, prints the plot else returns a plot object.

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.

See Also

ols_plot_added_variable(), ols_plot_resid_regressor()

Examples

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

`ols_plot_cooksd_bar` *Cooks' D bar plot*

Description

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

Usage

```
ols_plot_cooksd_bar(model, type = 1, threshold = NULL, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`type`	An integer between 1 and 5 selecting one of the 5 methods for computing the threshold.
`threshold`	Threshold for detecting outliers.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

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. There are several methods/formulas to compute the threshold used for detecting or classifying observations as outliers and we list them below.

- **Type 1** : 4 / n
- **Type 2** : 4 / (n - k - 1)
- **Type 3** : ~1
- **Type 4** : 1 / (n - k - 1)
- **Type 5** : 3 * mean(Vector of cook's distance values)

where **n** and **k** stand for

- **n**: Number of observations
- **k**: Number of predictors

Value

`ols_plot_cooksd_bar` returns a list containing the following components:

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

See Also

`ols_plot_cooksd_chart()`

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_plot_cooksd_bar(model)
ols_plot_cooksd_bar(model, type = 4)
ols_plot_cooksd_bar(model, threshold = 0.2)
```

`ols_plot_cooksd_chart` *Cooks' D chart*

Description

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

Usage

```
ols_plot_cooksd_chart(model, type = 1, threshold = NULL, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`type`	An integer between 1 and 5 selecting one of the 6 methods for computing the threshold.
`threshold`	Threshold for detecting outliers.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

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. There are several methods/formulas to compute the threshold used for detecting or classifying observations as outliers and we list them below.

- **Type 1** : 4 / n
- **Type 2** : 4 / (n - k - 1)
- **Type 3** : ~1
- **Type 4** : 1 / (n - k - 1)
- **Type 5** : 3 * mean(Vector of cook's distance values)

where **n** and **k** stand for

- **n**: Number of observations
- **k**: Number of predictors

Value

`ols_plot_cooksd_chart` returns a list containing the following components:

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

See Also

`ols_plot_cooksd_bar()`

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_plot_cooksd_chart(model)
ols_plot_cooksd_chart(model, type = 4)
ols_plot_cooksd_chart(model, threshold = 0.2)
```

`ols_plot_dfbetas` *DFBETAs panel*

Description

Panel of plots to detect influential observations using DFBETAs.

Usage

```
ols_plot_dfbetas(model, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

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_plot_dfbetas` returns a list of `data.frame` (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.

See Also

`ols_plot_dffits()`

Examples

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

`ols_plot_dffits` *DFFITS plot*

Description

Plot for detecting influential observations using DFFITs.

Usage

```
ols_plot_dffits(model, size_adj_threshold = TRUE, print_plot = TRUE)
```

Arguments

`model` An object of class `lm`.

`size_adj_threshold`
logical; if `TRUE` (the default), size adjusted threshold is used to determine influential observations.

`print_plot` logical; if `TRUE`, prints the plot else returns a plot object.

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:

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

A size-adjusted cutoff recommended by Belsley, Kuh, and Welsch is

$$2\sqrt{(p/n)}$$

and is used by default in **olsrr**.

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

Value

`ols_plot_dffits` returns a list containing the following components:

`outliers` a `data.frame` 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.

See Also

`ols_plot_dfbetas()`

Examples

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

---

ols_plot_diagnostics *Diagnostics panel*

Description

Panel of plots for regression diagnostics.

Usage

```
ols_plot_diagnostics(model, print_plot = TRUE)
```

Arguments

model	An object of class lm.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

Examples

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

---

ols_plot_hadi *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_plot_hadi(model, print_plot = TRUE)
```

Arguments

model	An object of class lm.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

References

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

See Also

`ols_plot_resid_pot()`

Examples

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

---

ols_plot_obs_fit *Observed vs fitted values plot*

Description

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

Usage

```
ols_plot_obs_fit(model, print_plot = TRUE)
```

Arguments

model	An object of class `lm`.
print_plot	logical; if `TRUE`, prints the plot else returns a plot object.

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_plot_obs_fit(model)
```

ols_plot_reg_line *Simple linear regression line*

Description

Plot to demonstrate that the regression line always passes through mean of the response and predictor variables.

Usage

```
ols_plot_reg_line(response, predictor, print_plot = TRUE)
```

Arguments

`response`	Response variable.
`predictor`	Predictor variable.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

Examples

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

ols_plot_resid_box *Residual box plot*

Description

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

Usage

```
ols_plot_resid_box(model, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

See Also

Other residual diagnostics: `ols_plot_resid_fit()`, `ols_plot_resid_hist()`, `ols_plot_resid_qq()`, `ols_test_correlation()`, `ols_test_normality()`

Examples

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

---

ols_plot_resid_fit *Residual vs fitted plot*

Description

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_plot_resid_fit(model, print_plot = TRUE)
```

Arguments

model	An object of class lm.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

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.

See Also

Other residual diagnostics: ols_plot_resid_box(), ols_plot_resid_hist(), ols_plot_resid_qq(), ols_test_correlation(), ols_test_normality()

Examples

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

ols_plot_resid_fit_spread *Residual fit spread plot*

Description

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

Usage

```
ols_plot_resid_fit_spread(model, print_plot = TRUE)

ols_plot_fm(model, print_plot = TRUE)

ols_plot_resid_spread(model, print_plot = TRUE)
```

Arguments

model	An object of class lm.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

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_plot_resid_fit_spread(model)

# fit mean plot
ols_plot_fm(model)

# residual spread plot
ols_plot_resid_spread(model)
```

`ols_plot_resid_hist` *Residual histogram*

Description

Histogram of residuals for detecting violation of normality assumption.

Usage

```
ols_plot_resid_hist(model, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

See Also

Other residual diagnostics: `ols_plot_resid_box()`, `ols_plot_resid_fit()`, `ols_plot_resid_qq()`, `ols_test_correlation()`, `ols_test_normality()`

Examples

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

`ols_plot_resid_lev` *Studentized residuals vs leverage plot*

Description

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

Usage

```
ols_plot_resid_lev(model, threshold = NULL, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`threshold`	Threshold for detecting outliers. Default is 2.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

See Also

ols_plot_resid_stud_fit(), ols_plot_resid_lev()

Examples

```
model <- lm(read ~ write + math + science, data = hsb)
ols_plot_resid_lev(model)
ols_plot_resid_lev(model, threshold = 3)
```

---

ols_plot_resid_pot   *Potential residual plot*

Description

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

Usage

```
ols_plot_resid_pot(model, print_plot = TRUE)
```

Arguments

model	An object of class lm.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

References

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

See Also

ols_plot_hadi()

Examples

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

ols_plot_resid_qq *Residual QQ plot*

Description

Graph for detecting violation of normality assumption.

Usage

```
ols_plot_resid_qq(model, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

See Also

Other residual diagnostics: `ols_plot_resid_box()`, `ols_plot_resid_fit()`, `ols_plot_resid_hist()`, `ols_test_correlation()`, `ols_test_normality()`

Examples

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

ols_plot_resid_regressor *Residual vs regressor 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_plot_resid_regressor(model, variable, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`variable`	New predictor to be added to the `model`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

See Also

ols_plot_added_variable(), ols_plot_comp_plus_resid()

Examples

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

---

ols_plot_resid_stand   *Standardized residual chart*

Description

Chart for identifying outliers.

Usage

```
ols_plot_resid_stand(model, threshold = NULL, print_plot = TRUE)
```

Arguments

model	An object of class lm.
threshold	Threshold for detecting outliers. Default is 2.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

Details

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

Value

ols_plot_resid_stand returns a list containing the following components:

outliers	a data.frame 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

See Also

ols_plot_resid_stud()

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_plot_resid_stand(model)
ols_plot_resid_stand(model, threshold = 3)
```

`ols_plot_resid_stud` *Studentized residual plot*

Description

Graph for identifying outliers.

Usage

```
ols_plot_resid_stud(model, threshold = NULL, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`threshold`	Threshold for detecting outliers. Default is 3.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

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_plot_resid_stud` returns a list containing the following components:

`outliers`	a `data.frame` 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

See Also

`ols_plot_resid_stand()`

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_plot_resid_stud(model)
ols_plot_resid_stud(model, threshold = 2)
```

---

ols_plot_resid_stud_fit
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_plot_resid_stud_fit(model, threshold = NULL, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`threshold`	Threshold for detecting outliers. Default is 2.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

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_plot_resid_stud_fit` returns a list containing the following components:

`outliers`	a `data.frame` 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

See Also

`ols_plot_resid_lev()`, `ols_plot_resid_stand()`, `ols_plot_resid_stud()`

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_plot_resid_stud_fit(model)
ols_plot_resid_stud_fit(model, threshold = 3)
```

---

ols_plot_response — *Response variable profile*

Description

Panel of plots to explore and visualize the response variable.

Usage

```
ols_plot_response(model, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_plot_response(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 of the model.

See Also

Other influence measures: `ols_hadi()`, `ols_leverage()`, `ols_press()`

Examples

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

`ols_prep_avplot_data` *Added variable plot data*

Description

Data for generating the added variable plots.

Usage

```
ols_prep_avplot_data(model)
```

Arguments

`model`	An object of class `lm`.

Examples

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

`ols_prep_cdplot_data` *Cooks' D plot data*

Description

Prepare data for cook's d bar plot.

Usage

```
ols_prep_cdplot_data(model, type = 1)
```

Arguments

`model`	An object of class `lm`.
`type`	An integer between 1 and 5 selecting one of the 6 methods for computing the threshold.

Examples

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

---

ols_prep_cdplot_outliers *Cooks' d outlier data*

Description

Outlier data for cook's d bar plot.

Usage

```
ols_prep_cdplot_outliers(k)
```

Arguments

k	Cooks' d bar plot data.

Examples

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

---

ols_prep_dfbeta_data *DFBETAs plot data*

Description

Prepares the data for dfbetas plot.

Usage

```
ols_prep_dfbeta_data(d, threshold)
```

Arguments

d	A `tibble` or `data.frame` with dfbetas.
threshold	The threshold for outliers.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
dfb <- dfbetas(model)
n <- nrow(dfb)
threshold <- 2 / sqrt(n)
dbetas  <- dfb[, 1]
df_data <- data.frame(obs = seq_len(n), dbetas = dbetas)
ols_prep_dfbeta_data(df_data, threshold)
```

---

ols_prep_dfbeta_outliers *DFBETAs plot outliers*

Description

Data for identifying outliers in dfbetas plot.

Usage

```
ols_prep_dfbeta_outliers(d)
```

Arguments

d	A `tibble` or `data.frame`.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
dfb <- dfbetas(model)
n <- nrow(dfb)
threshold <- 2 / sqrt(n)
dbetas  <- dfb[, 1]
df_data <- data.frame(obs = seq_len(n), dbetas = dbetas)
d <- ols_prep_dfbeta_data(df_data, threshold)
ols_prep_dfbeta_outliers(d)
```

ols_prep_dsrvf_data *Deleted studentized residual plot data*

Description

Generates data for deleted studentized residual vs fitted plot.

Usage

```
ols_prep_dsrvf_data(model, threshold = NULL)
```

Arguments

`model`	An object of class `lm`.
`threshold`	Threshold for detecting outliers. Default is 2.

Examples

```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_prep_dsrvf_data(model)
ols_prep_dsrvf_data(model, threshold = 3)
```

ols_prep_outlier_obs *Cooks' D outlier observations*

Description

Identify outliers in cook's d plot.

Usage

```
ols_prep_outlier_obs(k)
```

Arguments

`k`	Cooks' d bar plot data.

Examples

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

`ols_prep_regress_x` *Regress predictor on other predictors*

Description

Regress a predictor in the model on all the other predictors.

Usage

```
ols_prep_regress_x(data, i)
```

Arguments

data	A `data.frame`.
i	A numeric vector (indicates the predictor in the model).

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
data <- ols_prep_avplot_data(model)
ols_prep_regress_x(data, 1)
```

`ols_prep_regress_y` *Regress y on other predictors*

Description

Regress y on all the predictors except the ith predictor.

Usage

```
ols_prep_regress_y(data, i)
```

Arguments

data	A `data.frame`.
i	A numeric vector (indicates the predictor in the model).

Examples

```
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
data <- ols_prep_avplot_data(model)
ols_prep_regress_y(data, 1)
```

ols_prep_rfsplot_fmdata

Residual fit spread plot data

Description

Data for generating residual fit spread plot.

Usage

```
ols_prep_rfsplot_fmdata(model)

ols_prep_rfsplot_rsdata(model)
```

Arguments

model	An object of class lm.

Examples

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

ols_prep_rstudlev_data

Studentized residual vs leverage plot data

Description

Generates data for studentized resiudual vs leverage plot.

Usage

```
ols_prep_rstudlev_data(model, threshold = NULL)
```

Arguments

model	An object of class lm.
threshold	Threshold for detecting outliers. Default is 2.

Examples

```
model <- lm(read ~ write + math + science, data = hsb)
ols_prep_rstudlev_data(model)
ols_prep_rstudlev_data(model, threshold = 3)
```

---

ols_prep_rvsrplot_data   *Residual vs regressor plot data*

Description

Data for generating residual vs regressor plot.

Usage

```
ols_prep_rvsrplot_data(model)
```

Arguments

`model`	An object of class `lm`.

Examples

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

---

ols_prep_srchart_data   *Standardized residual chart data*

Description

Generates data for standardized residual chart.

Usage

```
ols_prep_srchart_data(model, threshold = NULL)
```

Arguments

`model`	An object of class `lm`.
`threshold`	Threshold for detecting outliers. Default is 2.

Examples

```
model <- lm(read ~ write + math + science, data = hsb)
ols_prep_srchart_data(model)
ols_prep_srchart_data(model, threshold = 3)
```

`ols_prep_srplot_data` *Studentized residual plot data*

Description

Generates data for studentized residual plot.

Usage

```
ols_prep_srplot_data(model, threshold = NULL)
```

Arguments

`model`	An object of class `lm`.
`threshold`	Threshold for detecting outliers. Default is 3.

Examples

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

`ols_press` *PRESS*

Description

PRESS (prediction sum of squares) 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 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 of the model.

References

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

See Also

Other influence measures: `ols_hadi()`, `ols_leverage()`, `ols_pred_rsq()`

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	data.frame 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
`rmse`	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`

Interaction Terms

If the model includes interaction terms, the standardized betas are computed after scaling and centering the predictors.

References

https://www.ssc.wisc.edu/~hemken/Stataworkshops/stdBeta/Getting%20Standardized%20Coefficients%20Right.pdf

Examples

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

# if model includes interaction terms set iterm to TRUE
ols_regress(mpg ~ disp * wt, data = mtcars, iterm = TRUE)
```

`ols_sbc` *Bayesian information criterion*

Description

Bayesian information criterion for model selection.

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

The bayesian information criterion of the model.

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.

See Also

Other model selection criteria: `ols_aic()`, `ols_apc()`, `ols_fpe()`, `ols_hsp()`, `ols_mallows_cp()`, `ols_msep()`, `ols_sbic()`

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` *Sawa's bayesian information criterion*

Description

Sawa's bayesian information criterion for model selection.

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 Criterion (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.

See Also

Other model selection criteria: ols_aic(), ols_apc(), ols_fpe(), ols_hsp(), ols_mallows_cp(), ols_msep(), ols_sbc()

Examples

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

---

ols_step_all_possible *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_step_all_possible(model, ...)

## Default S3 method:
ols_step_all_possible(model, max_order = NULL, ...)

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

Arguments

model	An object of class lm.
...	Other arguments.
max_order	Maximum subset order.
x	An object of class ols_step_all_possible.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

Value

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

`mindex`	model index
`n`	number of predictors
`predictors`	predictors in the model
`rsquare`	rsquare of the model
`adjr`	adjusted rsquare of the model
`rmse`	root mean squared error 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
`msep`	estimated MSE of prediction, assuming multivariate normality
`fpe`	final prediction error
`apc`	amemiya prediction criteria
`hsp`	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_step_all_possible(model)
k

# plot
plot(k)

# maximum subset
model <- lm(mpg ~ disp + hp + drat + wt + qsec, data = mtcars)
ols_step_all_possible(model, max_order = 3)
```

`ols_step_all_possible_betas` *All possible regression variable coefficients*

Description

Returns the coefficients for each variable from each model.

Usage

```
ols_step_all_possible_betas(object, ...)
```

Arguments

`object`	An object of class `lm`.
`...`	Other arguments.

Value

`ols_step_all_possible_betas` returns a `data.frame` containing:

`model_index`	model number
`predictor`	predictor
`beta_coef`	coefficient for the predictor

Examples

```
## Not run:
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_step_all_possible_betas(model)

## End(Not run)
```

`ols_step_backward_adj_r2` *Stepwise Adjusted R-Squared backward regression*

Description

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

Usage

```
ols_step_backward_adj_r2(model, ...)

## Default S3 method:
ols_step_backward_adj_r2(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_backward_adj_r2'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`; the model should include all candidate predictor variables.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, will print the regression result at each step.
`x`	An object of class `ols_step_backward_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other backward selection procedures: `ols_step_backward_aic()`, `ols_step_backward_p()`, `ols_step_backward_r2()`, `ols_step_backward_sbc()`, `ols_step_backward_sbic()`

Examples

```
# stepwise backward regression
model <- lm(y ~ ., data = surgical)
ols_step_backward_adj_r2(model)

# final model and selection metrics
k <- ols_step_backward_aic(model)
k$metrics
k$model

# include or exclude variable
# force variables to be included in the selection process
ols_step_backward_adj_r2(model, include = c("alc_mod", "gender"))

# use index of variable instead of name
ols_step_backward_adj_r2(model, include = c(7, 6))

# force variable to be excluded from selection process
ols_step_backward_adj_r2(model, exclude = c("alc_heavy", "bcs"))

# use index of variable instead of name
ols_step_backward_adj_r2(model, exclude = c(8, 1))
```

---

ols_step_backward_aic *Stepwise AIC backward regression*

Description

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

Usage

```
ols_step_backward_aic(model, ...)

## Default S3 method:
ols_step_backward_aic(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_backward_aic'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`; the model should include all candidate predictor variables.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, will print the regression result at each step.
`x`	An object of class `ols_step_backward_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other backward selection procedures: `ols_step_backward_adj_r2()`, `ols_step_backward_p()`, `ols_step_backward_r2()`, `ols_step_backward_sbc()`, `ols_step_backward_sbic()`

Examples

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

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

# selection metrics
k$metrics

# final model
k$model

# include or exclude variable
# force variables to be included in the selection process
```

```
ols_step_backward_aic(model, include = c("alc_mod", "gender"))

# use index of variable instead of name
ols_step_backward_aic(model, include = c(7, 6))

# force variable to be excluded from selection process
ols_step_backward_aic(model, exclude = c("alc_heavy", "bcs"))

# use index of variable instead of name
ols_step_backward_aic(model, exclude = c(8, 1))
```

`ols_step_backward_p` *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_p(model, ...)

## Default S3 method:
ols_step_backward_p(
  model,
  p_val = 0.3,
  include = NULL,
  exclude = NULL,
  hierarchical = FALSE,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_backward_p'
plot(x, model = NA, print_plot = TRUE, details = TRUE, ...)
```

Arguments

`model`	An object of class `lm`; the model should include all candidate predictor variables.
`...`	Other inputs.
`p_val`	p value; variables with p more than `p_val` will be removed from the model.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`hierarchical`	Logical; if `TRUE`, performs hierarchical selection.

progress	Logical; if TRUE, will display variable selection progress.
details	Logical; if TRUE, will print the regression result at each step.
x	An object of class ols_step_backward_p.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

Value

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

model	final model; an object of class lm
metrics	selection metrics

References

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

See Also

Other backward selection procedures: ols_step_backward_adj_r2(), ols_step_backward_aic(), ols_step_backward_r2(), ols_step_backward_sbc(), ols_step_backward_sbic()

Examples

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

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

# selection metrics
k$metrics

# final model
k$model

# include or exclude variables
# force variable to be included in selection process
ols_step_backward_p(model, include = c("age", "alc_mod"))

# use index of variable instead of name
ols_step_backward_p(model, include = c(5, 7))

# force variable to be excluded from selection process
ols_step_backward_p(model, exclude = c("pindex"))

# use index of variable instead of name
```

```
ols_step_backward_p(model, exclude = c(2))

# hierarchical selection
model <- lm(y ~ bcs + alc_heavy + pindex + age + alc_mod, data = surgical)
ols_step_backward_p(model, 0.1, hierarchical = TRUE)

# plot
k <- ols_step_backward_p(model, 0.1, hierarchical = TRUE)
plot(k)
```

`ols_step_backward_r2` *Stepwise R-Squared backward regression*

Description

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

Usage

```
ols_step_backward_r2(model, ...)

## Default S3 method:
ols_step_backward_r2(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_backward_r2'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`; the model should include all candidate predictor variables.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, will print the regression result at each step.
`x`	An object of class `ols_step_backward_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other backward selection procedures: `ols_step_backward_adj_r2()`, `ols_step_backward_aic()`, `ols_step_backward_p()`, `ols_step_backward_sbc()`, `ols_step_backward_sbic()`

Examples

```
# stepwise backward regression
model <- lm(y ~ ., data = surgical)
ols_step_backward_r2(model)

# final model and selection metrics
k <- ols_step_backward_aic(model)
k$metrics
k$model

# include or exclude variable
# force variables to be included in the selection process
ols_step_backward_r2(model, include = c("alc_mod", "gender"))

# use index of variable instead of name
ols_step_backward_r2(model, include = c(7, 6))

# force variable to be excluded from selection process
ols_step_backward_r2(model, exclude = c("alc_heavy", "bcs"))

# use index of variable instead of name
ols_step_backward_r2(model, exclude = c(8, 1))
```

`ols_step_backward_sbc` *Stepwise SBC backward regression*

Description

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

Usage

```
ols_step_backward_sbc(model, ...)

## Default S3 method:
ols_step_backward_sbc(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_backward_sbc'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

model	An object of class lm; the model should include all candidate predictor variables.
...	Other arguments.
include	Character or numeric vector; variables to be included in selection process.
exclude	Character or numeric vector; variables to be excluded from selection process.
progress	Logical; if TRUE, will display variable selection progress.
details	Logical; if TRUE, will print the regression result at each step.
x	An object of class ols_step_backward_*.
print_plot	logical; if TRUE, prints the plot else returns a plot object.
digits	Number of decimal places to display.

Value

List containing the following components:

model	final model; an object of class lm
metrics	selection metrics
others	list; info used for plotting and printing

References

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

See Also

Other backward selection procedures: ols_step_backward_adj_r2(), ols_step_backward_aic(), ols_step_backward_p(), ols_step_backward_r2(), ols_step_backward_sbic()

Examples

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

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

# selection metrics
k$metrics

# final model
k$model

# include or exclude variable
# force variables to be included in the selection process
ols_step_backward_sbc(model, include = c("alc_mod", "gender"))

# use index of variable instead of name
ols_step_backward_sbc(model, include = c(7, 6))

# force variable to be excluded from selection process
ols_step_backward_sbc(model, exclude = c("alc_heavy", "bcs"))

# use index of variable instead of name
ols_step_backward_sbc(model, exclude = c(8, 1))
```

`ols_step_backward_sbic` *Stepwise SBIC backward regression*

Description

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

Usage

```
ols_step_backward_sbic(model, ...)

## Default S3 method:
ols_step_backward_sbic(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
```

```
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_backward_sbic'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

model	An object of class lm; the model should include all candidate predictor variables.
...	Other arguments.
include	Character or numeric vector; variables to be included in selection process.
exclude	Character or numeric vector; variables to be excluded from selection process.
progress	Logical; if TRUE, will display variable selection progress.
details	Logical; if TRUE, will print the regression result at each step.
x	An object of class ols_step_backward_*.
print_plot	logical; if TRUE, prints the plot else returns a plot object.
digits	Number of decimal places to display.

Value

List containing the following components:

model	final model; an object of class lm
metrics	selection metrics
others	list; info used for plotting and printing

References

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

See Also

Other backward selection procedures: ols_step_backward_adj_r2(), ols_step_backward_aic(), ols_step_backward_p(), ols_step_backward_r2(), ols_step_backward_sbc()

Examples

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

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

```
# selection metrics
k$metrics

# final model
k$model

# include or exclude variable
# force variables to be included in the selection process
ols_step_backward_sbic(model, include = c("alc_mod", "gender"))

# use index of variable instead of name
ols_step_backward_sbic(model, include = c(7, 6))

# force variable to be excluded from selection process
ols_step_backward_sbic(model, exclude = c("alc_heavy", "bcs"))

# use index of variable instead of name
ols_step_backward_sbic(model, exclude = c(8, 1))
```

---

ols_step_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. The default metric used for selecting the model is R2 but the user can choose any of the other available metrics.

Usage

```
ols_step_best_subset(model, ...)

## Default S3 method:
ols_step_best_subset(
  model,
  max_order = NULL,
  include = NULL,
  exclude = NULL,
  metric = c("rsquare", "adjr", "predrsq", "cp", "aic", "sbic", "sbc", "msep", "fpe",
    "apc", "hsp"),
  ...
)

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

Arguments

`model`	An object of class `lm`.
`...`	Other inputs.
`max_order`	Maximum subset order.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`metric`	Metric to select model.
`x`	An object of class `ols_step_best_subset`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

Value

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

`metrics`	selection metrics

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_step_best_subset(model)
ols_step_best_subset(model, metric = "adjr")
ols_step_best_subset(model, metric = "cp")

# maximum subset
model <- lm(mpg ~ disp + hp + drat + wt + qsec, data = mtcars)
ols_step_best_subset(model, max_order = 3)

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

# return only models including `qsec`
ols_step_best_subset(model, include = c("qsec"))

# exclude `hp` from selection process
ols_step_best_subset(model, exclude = c("hp"))
```

ols_step_both_adj_r2 *Stepwise Adjusted R-Squared regression*

Description

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

Usage

```
ols_step_both_adj_r2(model, ...)

## Default S3 method:
ols_step_both_adj_r2(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_both_adj_r2'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, details of variable selection will be printed on screen.
`x`	An object of class `ols_step_both_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other both direction selection procedures: ols_step_both_aic(), ols_step_both_r2(), ols_step_both_sbc(), ols_step_both_sbic()

Examples

```
## Not run:
# stepwise regression
model <- lm(y ~ ., data = stepdata)
ols_step_both_adj_r2(model)

# stepwise regression plot
model <- lm(y ~ ., data = stepdata)
k <- ols_step_both_adj_r2(model)
plot(k)

# selection metrics
k$metrics

# final model
k$model

# include or exclude variables
# force variable to be included in selection process
model <- lm(y ~ ., data = stepdata)

ols_step_both_adj_r2(model, include = c("x6"))

# use index of variable instead of name
ols_step_both_adj_r2(model, include = c(6))

# force variable to be excluded from selection process
ols_step_both_adj_r2(model, exclude = c("x2"))

# use index of variable instead of name
ols_step_both_adj_r2(model, exclude = c(2))

# include & exclude variables in the selection process
ols_step_both_adj_r2(model, include = c("x6"), exclude = c("x2"))

# use index of variable instead of name
ols_step_both_adj_r2(model, include = c(6), exclude = c(2))

## End(Not run)
```

`ols_step_both_aic` *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_step_both_aic(model, ...)

## Default S3 method:
ols_step_both_aic(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_both_aic'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, details of variable selection will be printed on screen.
`x`	An object of class `ols_step_both_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other both direction selection procedures: ols_step_both_adj_r2(), ols_step_both_r2(), ols_step_both_sbc(), ols_step_both_sbic()

Examples

```
## Not run:
# stepwise regression
model <- lm(y ~ ., data = stepdata)
ols_step_both_aic(model)

# stepwise regression plot
model <- lm(y ~ ., data = stepdata)
k <- ols_step_both_aic(model)
plot(k)

# selection metrics
k$metrics

# final model
k$model

# include or exclude variables
# force variable to be included in selection process
model <- lm(y ~ ., data = stepdata)

ols_step_both_aic(model, include = c("x6"))

# use index of variable instead of name
ols_step_both_aic(model, include = c(6))

# force variable to be excluded from selection process
ols_step_both_aic(model, exclude = c("x2"))

# use index of variable instead of name
ols_step_both_aic(model, exclude = c(2))

# include & exclude variables in the selection process
ols_step_both_aic(model, include = c("x6"), exclude = c("x2"))

# use index of variable instead of name
ols_step_both_aic(model, include = c(6), exclude = c(2))

## End(Not run)
```

ols_step_both_p *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_step_both_p(model, ...)

## Default S3 method:
ols_step_both_p(
  model,
  p_enter = 0.1,
  p_remove = 0.3,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_both_p'
plot(x, model = NA, print_plot = TRUE, details = TRUE, ...)
```

Arguments

`model`	An object of class `lm`; the model should include all candidate predictor variables.
`...`	Other arguments.
`p_enter`	p value; variables with p value less than `p_enter` will enter into the model.
`p_remove`	p value; variables with p more than `p_remove` will be removed from the model.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, will print the regression result at each step.
`x`	An object of class `ols_step_both_p`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

Value

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

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`beta_pval`	beta and p values of models in each selection step

References

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

Examples

```
## Not run:
# stepwise regression
model <- lm(y ~ ., data = surgical)
ols_step_both_p(model)

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

# selection metrics
k$metrics

# final model
k$model

# include or exclude variables
model <- lm(y ~ ., data = stepdata)

# force variable to be included in selection process
ols_step_both_p(model, include = c("x6"))

# use index of variable instead of name
ols_step_both_p(model, include = c(6))

# force variable to be excluded from selection process
ols_step_both_p(model, exclude = c("x1"))

# use index of variable instead of name
ols_step_both_p(model, exclude = c(1))

## End(Not run)
```

ols_step_both_r2 *Stepwise R-Squared regression*

Description

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

Usage

```
ols_step_both_r2(model, ...)

## Default S3 method:
ols_step_both_r2(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_both_r2'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, details of variable selection will be printed on screen.
`x`	An object of class `ols_step_both_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other both direction selection procedures: ols_step_both_adj_r2(), ols_step_both_aic(), ols_step_both_sbc(), ols_step_both_sbic()

Examples

```
## Not run:
# stepwise regression
model <- lm(y ~ ., data = stepdata)
ols_step_both_r2(model)

# stepwise regression plot
model <- lm(y ~ ., data = stepdata)
k <- ols_step_both_r2(model)
plot(k)

# selection metrics
k$metrics

# final model
k$model

# include or exclude variables
# force variable to be included in selection process
model <- lm(y ~ ., data = stepdata)

ols_step_both_r2(model, include = c("x6"))

# use index of variable instead of name
ols_step_both_r2(model, include = c(6))

# force variable to be excluded from selection process
ols_step_both_r2(model, exclude = c("x2"))

# use index of variable instead of name
ols_step_both_r2(model, exclude = c(2))

# include & exclude variables in the selection process
ols_step_both_r2(model, include = c("x6"), exclude = c("x2"))

# use index of variable instead of name
ols_step_both_r2(model, include = c(6), exclude = c(2))

## End(Not run)
```

ols_step_both_sbc *Stepwise SBC regression*

Description

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

Usage

```
ols_step_both_sbc(model, ...)

## Default S3 method:
ols_step_both_sbc(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_both_sbc'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, details of variable selection will be printed on screen.
`x`	An object of class `ols_step_both_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other both direction selection procedures: ols_step_both_adj_r2(), ols_step_both_aic(), ols_step_both_r2(), ols_step_both_sbic()

Examples

```
## Not run:
# stepwise regression
model <- lm(y ~ ., data = stepdata)
ols_step_both_sbc(model)

# stepwise regression plot
model <- lm(y ~ ., data = stepdata)
k <- ols_step_both_sbc(model)
plot(k)

# selection metrics
k$metrics

# final model
k$model

# include or exclude variables
# force variable to be included in selection process
model <- lm(y ~ ., data = stepdata)

ols_step_both_sbc(model, include = c("x6"))

# use index of variable instead of name
ols_step_both_sbc(model, include = c(6))

# force variable to be excluded from selection process
ols_step_both_sbc(model, exclude = c("x2"))

# use index of variable instead of name
ols_step_both_sbc(model, exclude = c(2))

# include & exclude variables in the selection process
ols_step_both_sbc(model, include = c("x6"), exclude = c("x2"))

# use index of variable instead of name
ols_step_both_sbc(model, include = c(6), exclude = c(2))

## End(Not run)
```

ols_step_both_sbic — *Stepwise SBIC regression*

Description

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

Usage

```
ols_step_both_sbic(model, ...)

## Default S3 method:
ols_step_both_sbic(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_both_sbic'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, details of variable selection will be printed on screen.
`x`	An object of class `ols_step_both_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other both direction selection procedures: ols_step_both_adj_r2(), ols_step_both_aic(), ols_step_both_r2(), ols_step_both_sbc()

Examples

```
## Not run:
# stepwise regression
model <- lm(y ~ ., data = stepdata)
ols_step_both_sbic(model)

# stepwise regression plot
model <- lm(y ~ ., data = stepdata)
k <- ols_step_both_sbic(model)
plot(k)

# selection metrics
k$metrics

# final model
k$model

# include or exclude variables
# force variable to be included in selection process
model <- lm(y ~ ., data = stepdata)

ols_step_both_sbic(model, include = c("x6"))

# use index of variable instead of name
ols_step_both_sbic(model, include = c(6))

# force variable to be excluded from selection process
ols_step_both_sbic(model, exclude = c("x2"))

# use index of variable instead of name
ols_step_both_sbic(model, exclude = c(2))

# include & exclude variables in the selection process
ols_step_both_sbic(model, include = c("x6"), exclude = c("x2"))

# use index of variable instead of name
ols_step_both_sbic(model, include = c(6), exclude = c(2))

## End(Not run)
```

`ols_step_forward_adj_r2`

Stepwise Adjusted R-Squared forward regression

Description

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

Usage

```
ols_step_forward_adj_r2(model, ...)

## Default S3 method:
ols_step_forward_adj_r2(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_forward_adj_r2'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, will print the regression result at each step.
`x`	An object of class `ols_step_forward_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other forward selection procedures: ols_step_forward_aic(), ols_step_forward_p(), ols_step_forward_r2(), ols_step_forward_sbc(), ols_step_forward_sbic()

Examples

```
# stepwise forward regression
model <- lm(y ~ ., data = surgical)
ols_step_forward_adj_r2(model)

# stepwise forward regression plot
k <- ols_step_forward_adj_r2(model)
plot(k)

# selection metrics
k$metrics

# extract final model
k$model

# include or exclude variables
# force variable to be included in selection process
ols_step_forward_adj_r2(model, include = c("age"))

# use index of variable instead of name
ols_step_forward_adj_r2(model, include = c(5))

# force variable to be excluded from selection process
ols_step_forward_adj_r2(model, exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_adj_r2(model, exclude = c(4))

# include & exclude variables in the selection process
ols_step_forward_adj_r2(model, include = c("age"), exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_adj_r2(model, include = c(5), exclude = c(4))
```

---

ols_step_forward_aic *Stepwise AIC forward regression*

Description

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

Usage

```
ols_step_forward_aic(model, ...)

## Default S3 method:
ols_step_forward_aic(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_forward_aic'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

`model`	An object of class `lm`.
`...`	Other arguments.
`include`	Character or numeric vector; variables to be included in selection process.
`exclude`	Character or numeric vector; variables to be excluded from selection process.
`progress`	Logical; if `TRUE`, will display variable selection progress.
`details`	Logical; if `TRUE`, will print the regression result at each step.
`x`	An object of class `ols_step_forward_*`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.
`digits`	Number of decimal places to display.

Value

List containing the following components:

`model`	final model; an object of class `lm`
`metrics`	selection metrics
`others`	list; info used for plotting and printing

References

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

See Also

Other forward selection procedures: `ols_step_forward_adj_r2()`, `ols_step_forward_p()`, `ols_step_forward_r2()`, `ols_step_forward_sbc()`, `ols_step_forward_sbic()`

Examples

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

# stepwise forward regression plot
k <- ols_step_forward_aic(model)
plot(k)

# selection metrics
k$metrics

# extract final model
k$model

# include or exclude variables
# force variable to be included in selection process
ols_step_forward_aic(model, include = c("age"))

# use index of variable instead of name
ols_step_forward_aic(model, include = c(5))

# force variable to be excluded from selection process
ols_step_forward_aic(model, exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_aic(model, exclude = c(4))

# include & exclude variables in the selection process
ols_step_forward_aic(model, include = c("age"), exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_aic(model, include = c(5), exclude = c(4))
```

`ols_step_forward_p` *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_p(model, ...)

## Default S3 method:
ols_step_forward_p(
```

```
  model,
  p_val = 0.3,
  include = NULL,
  exclude = NULL,
  hierarchical = FALSE,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_forward_p'
plot(x, model = NA, print_plot = TRUE, details = TRUE, ...)
```

Arguments

model	An object of class lm; the model should include all candidate predictor variables.
...	Other arguments.
p_val	p value; variables with p value less than p_val will enter into the model
include	Character or numeric vector; variables to be included in selection process.
exclude	Character or numeric vector; variables to be excluded from selection process.
hierarchical	Logical; if TRUE, performs hierarchical selection.
progress	Logical; if TRUE, will display variable selection progress.
details	Logical; if TRUE, will print the regression result at each step.
x	An object of class ols_step_forward_p.
print_plot	logical; if TRUE, prints the plot else returns a plot object.

Value

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

model	final model; an object of class lm
metrics	selection metrics

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.

See Also

Other forward selection procedures: ols_step_forward_adj_r2(), ols_step_forward_aic(), ols_step_forward_r2(), ols_step_forward_sbc(), ols_step_forward_sbic()

Examples

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

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

# selection metrics
k$metrics

# final model
k$model

# include or exclude variables
# force variable to be included in selection process
ols_step_forward_p(model, include = c("age", "alc_mod"))

# use index of variable instead of name
ols_step_forward_p(model, include = c(5, 7))

# force variable to be excluded from selection process
ols_step_forward_p(model, exclude = c("pindex"))

# use index of variable instead of name
ols_step_forward_p(model, exclude = c(2))

# hierarchical selection
model <- lm(y ~ bcs + alc_heavy + pindex + enzyme_test, data = surgical)
ols_step_forward_p(model, 0.1, hierarchical = TRUE)

# plot
k <- ols_step_forward_p(model, 0.1, hierarchical = TRUE)
plot(k)
```

`ols_step_forward_r2` *Stepwise R-Squared forward regression*

Description

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

Usage

```
ols_step_forward_r2(model, ...)
```

```
## Default S3 method:
ols_step_forward_r2(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_forward_r2'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

model	An object of class lm.
...	Other arguments.
include	Character or numeric vector; variables to be included in selection process.
exclude	Character or numeric vector; variables to be excluded from selection process.
progress	Logical; if TRUE, will display variable selection progress.
details	Logical; if TRUE, will print the regression result at each step.
x	An object of class ols_step_forward_*.
print_plot	logical; if TRUE, prints the plot else returns a plot object.
digits	Number of decimal places to display.

Value

List containing the following components:

model	final model; an object of class lm
metrics	selection metrics
others	list; info used for plotting and printing

References

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

See Also

Other forward selection procedures: ols_step_forward_adj_r2(), ols_step_forward_aic(), ols_step_forward_p(), ols_step_forward_sbc(), ols_step_forward_sbic()

Examples

```
# stepwise forward regression
model <- lm(y ~ ., data = surgical)
ols_step_forward_r2(model)

# stepwise forward regression plot
k <- ols_step_forward_r2(model)
plot(k)

# selection metrics
k$metrics

# extract final model
k$model

# include or exclude variables
# force variable to be included in selection process
ols_step_forward_r2(model, include = c("age"))

# use index of variable instead of name
ols_step_forward_r2(model, include = c(5))

# force variable to be excluded from selection process
ols_step_forward_r2(model, exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_r2(model, exclude = c(4))

# include & exclude variables in the selection process
ols_step_forward_r2(model, include = c("age"), exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_r2(model, include = c(5), exclude = c(4))
```

`ols_step_forward_sbc` *Stepwise SBC forward regression*

Description

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

Usage

```
ols_step_forward_sbc(model, ...)

## Default S3 method:
ols_step_forward_sbc(
```

```
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
)

## S3 method for class 'ols_step_forward_sbc'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

model	An object of class lm.
...	Other arguments.
include	Character or numeric vector; variables to be included in selection process.
exclude	Character or numeric vector; variables to be excluded from selection process.
progress	Logical; if TRUE, will display variable selection progress.
details	Logical; if TRUE, will print the regression result at each step.
x	An object of class ols_step_forward_*.
print_plot	logical; if TRUE, prints the plot else returns a plot object.
digits	Number of decimal places to display.

Value

List containing the following components:

model	final model; an object of class lm
metrics	selection metrics
others	list; info used for plotting and printing

References

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

See Also

Other forward selection procedures: ols_step_forward_adj_r2(), ols_step_forward_aic(), ols_step_forward_p(), ols_step_forward_r2(), ols_step_forward_sbic()

Examples

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

# stepwise forward regression plot
```

```
k <- ols_step_forward_sbc(model)
plot(k)

# selection metrics
k$metrics

# extract final model
k$model

# include or exclude variables
# force variable to be included in selection process
ols_step_forward_sbc(model, include = c("age"))

# use index of variable instead of name
ols_step_forward_sbc(model, include = c(5))

# force variable to be excluded from selection process
ols_step_forward_sbc(model, exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_sbc(model, exclude = c(4))

# include & exclude variables in the selection process
ols_step_forward_sbc(model, include = c("age"), exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_sbc(model, include = c(5), exclude = c(4))
```

---

ols_step_forward_sbic *Stepwise SBIC forward regression*

Description

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

Usage

```
ols_step_forward_sbic(model, ...)

## Default S3 method:
ols_step_forward_sbic(
  model,
  include = NULL,
  exclude = NULL,
  progress = FALSE,
  details = FALSE,
  ...
```

```
)

## S3 method for class 'ols_step_forward_sbic'
plot(x, print_plot = TRUE, details = TRUE, digits = 3, ...)
```

Arguments

model	An object of class lm.
...	Other arguments.
include	Character or numeric vector; variables to be included in selection process.
exclude	Character or numeric vector; variables to be excluded from selection process.
progress	Logical; if TRUE, will display variable selection progress.
details	Logical; if TRUE, will print the regression result at each step.
x	An object of class ols_step_forward_*.
print_plot	logical; if TRUE, prints the plot else returns a plot object.
digits	Number of decimal places to display.

Value

List containing the following components:

model	final model; an object of class lm
metrics	selection metrics
others	list; info used for plotting and printing

References

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

See Also

Other forward selection procedures: ols_step_forward_adj_r2(), ols_step_forward_aic(), ols_step_forward_p(), ols_step_forward_r2(), ols_step_forward_sbc()

Examples

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

# stepwise forward regression plot
k <- ols_step_forward_sbic(model)
plot(k)

# selection metrics
k$metrics

# extract final model
```

```
k$model

# include or exclude variables
# force variable to be included in selection process
ols_step_forward_sbic(model, include = c("age"))

# use index of variable instead of name
ols_step_forward_sbic(model, include = c(5))

# force variable to be excluded from selection process
ols_step_forward_sbic(model, exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_sbic(model, exclude = c(4))

# include & exclude variables in the selection process
ols_step_forward_sbic(model, include = c("age"), exclude = c("liver_test"))

# use index of variable instead of name
ols_step_forward_sbic(model, include = c(5), exclude = c(4))
```

`ols_test_bartlett` *Bartlett test*

Description

Test if k samples are from populations with equal variances.

Usage

```
ols_test_bartlett(data, ...)

## Default S3 method:
ols_test_bartlett(data, ..., group_var = NULL)
```

Arguments

`data`	A `data.frame` or `tibble`.
`...`	Columns in `data`.
`group_var`	Grouping variable.

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

`fstat`	f statistic
`pval`	p-value of `fstat`
`df`	degrees of freedom

References

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

See Also

Other heteroskedasticity tests: `ols_test_breusch_pagan()`, `ols_test_f()`, `ols_test_score()`

Examples

```
# using grouping variable
if (require("descriptr")) {
  library(descriptr)
  ols_test_bartlett(mtcarz, 'mpg', group_var = 'cyl')
}

# using variables
ols_test_bartlett(hsb, 'read', 'write')
```

`ols_test_breusch_pagan`   *Breusch pagan test*

Description

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

Usage

```
ols_test_breusch_pagan(
  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`	Adjustment for p value, the following options are available: bonferroni, holm, sidak and none.
`vars`	Variables to be used 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_test_breusch_pagan` returns an object of class `"ols_test_breusch_pagan"`. An object of class `"ols_test_breusch_pagan"` 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.

See Also

Other heteroskedasticity tests: `ols_test_bartlett()`, `ols_test_f()`, `ols_test_score()`

Examples

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

# use fitted values of the model
ols_test_breusch_pagan(model)

# use independent variables of the model
ols_test_breusch_pagan(model, rhs = TRUE)

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

# bonferroni p value adjustment
ols_test_breusch_pagan(model, rhs = TRUE, multiple = TRUE, p.adj = 'bonferroni')

# sidak p value adjustment
ols_test_breusch_pagan(model, rhs = TRUE, multiple = TRUE, p.adj = 'sidak')

# holm's p value adjustment
ols_test_breusch_pagan(model, rhs = TRUE, multiple = TRUE, p.adj = 'holm')
```

`ols_test_correlation` *Correlation test for normality*

Description

Correlation between observed residuals and expected residuals under normality.

Usage

```
ols_test_correlation(model)
```

Arguments

`model`	An object of class `lm`.

Value

Correlation between fitted regression model residuals and expected values of residuals.

See Also

Other residual diagnostics: `ols_plot_resid_box()`, `ols_plot_resid_fit()`, `ols_plot_resid_hist()`, `ols_plot_resid_qq()`, `ols_test_normality()`

Examples

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

`ols_test_f` *F test*

Description

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

Usage

```
ols_test_f(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_test_f` returns an object of class `"ols_test_f"`. An object of class `"ols_test_f"` 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.

See Also

Other heteroskedasticity tests: `ols_test_bartlett()`, `ols_test_breusch_pagan()`, `ols_test_score()`

Examples

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

# using fitted values
ols_test_f(model)

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

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

---

ols_test_normality *Test for normality*

Description

Test for detecting violation of normality assumption.

Usage

```
ols_test_normality(y, ...)

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

Arguments

y	A numeric vector or an object of class lm.
...	Other arguments.

Value

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

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

See Also

Other residual diagnostics: `ols_plot_resid_box()`, `ols_plot_resid_fit()`, `ols_plot_resid_hist()`, `ols_plot_resid_qq()`, `ols_test_correlation()`

Examples

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

`ols_test_outlier` *Bonferroni Outlier Test*

Description

Detect outliers using Bonferroni p values.

Usage

```
ols_test_outlier(model, cut_off = 0.05, n_max = 10, ...)
```

Arguments

`model`	An object of class `lm`.
`cut_off`	Bonferroni p-values cut off for reporting observations.
`n_max`	Maximum number of observations to report, default is 10.
`...`	Other arguments.

Examples

```
# model
model <- lm(y ~ ., data = surgical)
ols_test_outlier(model)
```

ols_test_score *Score test*

Description

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

Usage

```
ols_test_score(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_test_score returns an object of class "ols_test_score". An object of class "ols_test_score" 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.

See Also

Other heteroskedasticity tests: ols_test_bartlett(), ols_test_breusch_pagan(), ols_test_f()

Examples

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

# using fitted values of the model
ols_test_score(model)

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

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

---

rvsr_plot_shiny — *Residual vs regressors plot for shiny app*

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, print_plot = TRUE)
```

Arguments

`model`	An object of class `lm`.
`data`	A `data.frame` or `tibble`.
`variable`	Character; new predictor to be added to the `model`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

Examples

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

stepdata — *Test Data Set*

Description

Test Data Set

Usage

```
stepdata
```

Format

An object of class `data.frame` with 20000 rows and 7 columns.

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