Package ‘olsrr’

October 14, 2022

Type Package

Title Tools for Building OLS Regression Models

Version 0.5.3

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, data.table, ggplot2, goftest, graphics, gridExtra, nortest, Rcpp, stats, utils

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

License MIT + file LICENSE


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

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 6.1.1

LinkingTo Rcpp

NeedsCompilation yes

Author Aravind Hebbali [aut, cre]

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

Repository CRAN

Date/Publication 2020-02-10 12:00:02 UTC
R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto</td>
<td>3</td>
</tr>
<tr>
<td>cement</td>
<td>4</td>
</tr>
<tr>
<td>fitness</td>
<td>4</td>
</tr>
<tr>
<td>hsb</td>
<td>4</td>
</tr>
<tr>
<td>olsrr</td>
<td>5</td>
</tr>
<tr>
<td>ols_aic</td>
<td>5</td>
</tr>
<tr>
<td>ols_apc</td>
<td>6</td>
</tr>
<tr>
<td>ols_coll_diag</td>
<td>7</td>
</tr>
<tr>
<td>ols_correlations</td>
<td>9</td>
</tr>
<tr>
<td>ols_fpe</td>
<td>10</td>
</tr>
<tr>
<td>ols_hadi</td>
<td>11</td>
</tr>
<tr>
<td>ols_hsp</td>
<td>12</td>
</tr>
<tr>
<td>ols_launch_app</td>
<td>13</td>
</tr>
<tr>
<td>ols_leverage</td>
<td>13</td>
</tr>
<tr>
<td>ols_mallows_cp</td>
<td>14</td>
</tr>
<tr>
<td>ols_msep</td>
<td>15</td>
</tr>
<tr>
<td>ols_plot_added_variable</td>
<td>16</td>
</tr>
<tr>
<td>ols_plot_comp_plus_resid</td>
<td>17</td>
</tr>
<tr>
<td>ols_plot_cooksd_bar</td>
<td>18</td>
</tr>
<tr>
<td>ols_plot_cooksd_chart</td>
<td>19</td>
</tr>
<tr>
<td>ols_plot_dfbetas</td>
<td>20</td>
</tr>
<tr>
<td>ols_plot_dffits</td>
<td>21</td>
</tr>
<tr>
<td>ols_plot_diagnostics</td>
<td>22</td>
</tr>
<tr>
<td>ols_plot_hadi</td>
<td>23</td>
</tr>
<tr>
<td>ols_plot_obs_fit</td>
<td>24</td>
</tr>
<tr>
<td>ols_plot_reg_line</td>
<td>24</td>
</tr>
<tr>
<td>ols_plot_resid_box</td>
<td>25</td>
</tr>
<tr>
<td>ols_plot_resid_fit</td>
<td>26</td>
</tr>
<tr>
<td>ols_plot_resid_fit_spread</td>
<td>27</td>
</tr>
<tr>
<td>ols_plot_resid_hist</td>
<td>28</td>
</tr>
<tr>
<td>ols_plot_resid_lev</td>
<td>28</td>
</tr>
<tr>
<td>ols_plot_resid_pot</td>
<td>29</td>
</tr>
<tr>
<td>ols_plot_resid_qq</td>
<td>30</td>
</tr>
<tr>
<td>ols_plot_residRegressor</td>
<td>31</td>
</tr>
<tr>
<td>ols_plot_resid_stand</td>
<td>31</td>
</tr>
<tr>
<td>ols_plot_resid_stud</td>
<td>32</td>
</tr>
<tr>
<td>ols_plot_resid_stud_fit</td>
<td>33</td>
</tr>
<tr>
<td>ols_plot_response</td>
<td>34</td>
</tr>
<tr>
<td>ols_pred_rsq</td>
<td>35</td>
</tr>
<tr>
<td>ols_prep_avplot_data</td>
<td>35</td>
</tr>
<tr>
<td>ols_prep_cdplot_data</td>
<td>36</td>
</tr>
<tr>
<td>ols_prep_cdplot_outliers</td>
<td>36</td>
</tr>
<tr>
<td>ols_prep_dfbeta_data</td>
<td>37</td>
</tr>
<tr>
<td>ols_prep_dfbeta_outliers</td>
<td>37</td>
</tr>
<tr>
<td>ols_prep_dsrvf_data</td>
<td>38</td>
</tr>
<tr>
<td>ols_prep_outlier_obs</td>
<td>38</td>
</tr>
</tbody>
</table>
auto

ols_prep_regress_x .............................................. 39
ols_prep_regress_y ............................................. 39
ols_prep_rfspplot_fmdata .................................... 40
ols_prep_rstudlev_data ...................................... 41
ols_prep_rvsrplot_data ....................................... 41
ols_prep_rschart_data ......................................... 42
ols_prep_srchart_data ......................................... 42
ols_press ......................................................... 43
ols_pure_error_anova .......................................... 44
ols_regress ....................................................... 45
ols_sbc .......................................................... 47
ols_sbic .......................................................... 48
ols_step_all_possible ........................................ 49
ols_step_all_possible_betas .................................. 51
ols_step_backward_aic ......................................... 52
ols_step_backward_p ............................................ 53
ols_step_best_subset .......................................... 55
ols_step_both_aic .............................................. 56
ols_step_both_p .................................................. 58
ols_step_forward_aic .......................................... 60
ols_step_forward_p ............................................. 61
ols_test_bartlett ................................................ 63
ols_test_breusch_pagan ....................................... 64
ols_test_correlation ........................................... 66
ols_test_f ......................................................... 67
ols_test_normality .............................................. 68
ols_test_outlier ................................................ 69
ols_test_score ................................................... 70
rivers ............................................................. 71
rvsr_plot_shiny .................................................. 71
stepdata .......................................................... 72
surgical ............................................................ 72

Index 74

<table>
<thead>
<tr>
<th>auto</th>
<th>Test Data Set</th>
</tr>
</thead>
</table>

Description

Test Data Set

Usage

auto

Format

An object of class tbl_df (inherits from tbl.data.frame) with 74 rows and 11 columns.
### cement

**Test Data Set**

**Usage**

cement

**Format**

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

### fitness

**Test Data Set**

**Usage**

fitness

**Format**

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

### hsb

**Test Data Set**

**Usage**

hsb

**Format**

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

Description

Tools for teaching and learning OLS regression

Details

See the README on GitHub

ols_aic

Akaike information criterion

Description

Akaike information criterion for model selection.

Usage

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

Arguments

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

Details

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

R & STATA

\[ AIC = -2(\text{loglikelihood}) + 2p \]

SAS

\[ AIC = n \star \ln(\text{SSE}/n) + 2p \]

where \( n \) is the sample size and \( p \) is the number of model parameters including intercept.

Value

Akaike information criterion of the model.
References


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")

Description

Amemiya’s prediction error.

Usage

ols_apc(model)

Arguments

model An object of class lm.

Details

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

\[ \left(\frac{n+p}{n-p}\right)(1 - (R^2)) \]
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**


**See Also**

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

**Examples**

```r
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**

```r
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 $\beta_k$ 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 $\beta_k$ 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


Examples

```r
# 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**

```r
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


Examples

```r
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

```r
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


See Also

Other model selection criteria: \texttt{ols\_aic}, \texttt{ols\_apc}, \texttt{ols\_hsp}, \texttt{ols\_mallows\_cp}, \texttt{ols\_msep}, \texttt{ols\_sbc}, \texttt{ols\_sbic}

Examples

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

---

\texttt{ols\_hadi} \hspace{1cm} \textit{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

\texttt{ols\_hadi(model)}

Arguments

\texttt{model} \hspace{1cm} An object of class \texttt{lm}.

Value

Hadi’s measure of the model.

References


See Also

Other influence measures: \texttt{ols\_leverage}, \texttt{ols\_pred\_rsq}, \texttt{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


See Also

Other model selection criteria: \texttt{ols_aic}, \texttt{ols_apc}, \texttt{ols_fpe}, \texttt{ols_mallows_cp}, \texttt{ols_msep}, \texttt{ols_sbc}, \texttt{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
```r
ols_launch_app()
```

Examples
```r
## 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
```r
ols_leverage(model)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>An object of class <code>lm</code>.</td>
</tr>
</tbody>
</table>

Value
Leverage of the model.

References

See Also
Other influence measures: `ols_hadi`, `ols_pred_rsq`, `ols_press`
Examples

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

- `ols_mallows_cp`  

<table>
<thead>
<tr>
<th>Description</th>
<th>Mallow’s Cp</th>
</tr>
</thead>
</table>

**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**


**See Also**

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

**Examples**

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


See Also

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

Examples

```r
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**

```r
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.

**Deprecated Function**

`ols_avplots()` has been deprecated. Instead use `ols_plot_added_variable()`.
References


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)

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.

Deprecated Function

ols_rpc_plot() has been deprecated. Instead use ols_plot_comp_plus_resid().

References


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, print_plot = TRUE)

Arguments

model An object of class lm.
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.

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
**ols_plot_cooksd_chart**

**Deprecated Function**

`ols_cooksd_barplot()` has been deprecated. Instead use `ols_plot_cooksd_bar()`.

**See Also**

[`ols_plot_cooksd_chart()`]

**Examples**

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

---

**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**

```r
ols_plot_cooksd_chart(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**

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

Steps to compute Cook’s distance:

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

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

**Value**

`ols_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
Deprecated Function

`ols_cooksd_chart()` has been deprecated. Instead use `ols_plot_cooksd_chart()`.

See Also

`[ols_plot_cooksd_bar()]`

Examples

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

Description

Panel of plots to detect influential observations using DFBETAs.

Usage

```r
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

DFBETAs are 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.

Deprecated Function

`ols_dfbetas_panel()` has been deprecated. Instead use `ols_plot_dfbetas()`.
References

See Also
[ols_plot_dffits()]

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

Description
Plot for detecting influential observations using DFFITs.

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

Steps to compute DFFITs:
- Delete observations one at a time.
- Refit the regression model on remaining \( n - 1 \) observations
- examine how much all of the fitted values change when the ith observation is deleted.

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

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

where \( n \) is the number of observations and \( p \) is the number of predictors including intercept.
Value

ols_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

Deprecated Function

ols_dffits_plot() has been deprecated. Instead use ols_plot_dffits().

References

Belsley, David A.; Kuh, Edwin; Welsh, Roy E. (1980). Regression Diagnostics: Identifying Influ-
ential Data and Sources of Collinearity.

0-471-05856-4.

See Also

[ols_plot_dfbetas()]

Examples

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

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.

# @section Deprecated Function: ols_diagnostic_panel() has been depre-
cated. Instead use ols_plot_diagnostics().
Examples

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

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

```r
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.

Deprecated Function

`ols_hadi_plot()` has been deprecated. Instead use `ols_plot_hadi()`.

References


See Also

[ols_plot_resid_pot()]

Examples

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

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

**ols_plot_obs_fit**  
*Observed vs fitted values plot*

Description

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

Usage

```r
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.

Deprecated Function

`ols_ovsp_plot()` has been deprecated. Instead use `ols_plot_obs_fit()`.

Examples

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

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

Deprecated Function
ols_reg_line() has been deprecated. Instead use ols_plot_reg_line().

Examples
ols_plot_reg_line(mtcars$mpg, mtcars$disp)

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.

Deprecated Function
ols_rsd_boxplot() has been deprecated. Instead use ols_plot_resid_box().

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.

Deprecated Function

ols_rvsp_plot() has been deprecated. Instead use ols_plot_resid_fit().

See Also

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

Examples

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

Deprecated Function

`ols_rfs_plot()`, `ols_fm_plot()` and `ols_rsd_plot()` has been deprecated. Instead use `ols_plot_resid_fit_spread()`, `ols_plot_fm()` and `ols_plot_resid_spread()`.

References


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
```
**Description**

Histogram of residuals for detecting violation of normality assumption.

**Usage**

```r
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.

**Deprecated Function**

`ols_rsd_hist()` has been deprecated. Instead use `ols_plot_resid_hist()`.

**See Also**

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

**Examples**

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

---

**Description**

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

**Usage**

```r
ols_plot_resid_lev(model, print_plot = TRUE)
```
Arguments

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

Deprecated Function

ols_rsdlev_plot() has been deprecated. Instead use ols_plot_resid_lev().

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_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.

Deprecated Function

ols_potrsd_plot() has been deprecated. Instead use ols_plot_resid_pot().

References


See Also

[ols_plot_hadi()]
Examples

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

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.

Deprecated Function

ols_rsd_qqplot() has been deprecated. Instead use ols_plot_resid_qq().

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**

```r
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.

**Deprecated Function**

`ols_rvsr_plot()` has been deprecated. Instead use `ols_plot_resid_regressor()`.

**See Also**

[ols_plot_added_variable()], [ols_plot_comp_plus_resid()]

**Examples**

```r
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**

```r
ols_plot_resid_stand(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**

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 residuals that exceed threshold for classifying an observation as an outlier
- **threshold**: threshold for classifying an observation as an outlier

** Deprecated Function**

`ols_srsd_chart()` has been deprecated. Instead use `ols_plot_resid_stand()`.

**See Also**

- `[ols_plot_resid_stud()]`

**Examples**

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

---

**Description**

Graph for identifying outliers.

**Usage**

`ols_plot_resid_stud(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
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

Deprecated Function
ols_srsd_plot() has been deprecated. Instead use ols_plot_resid_stud().

See Also
[ols_plot_resid_stand()]

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

### 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
```r
ols_plot_resid_stud_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
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

Deprecated Function
ols_dsrvsp_plot() has been deprecated. Instead use ols_plot_resid_stud_fit().

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)

Description
Response variable profile
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.

Deprecated Function
ols_resp_viz() has been deprecated. Instead use ols_plot_response().
### Examples

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

<table>
<thead>
<tr>
<th>ols_pred_rsq</th>
<th>Predicted rsquare</th>
</tr>
</thead>
</table>

**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**

```r
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

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

<table>
<thead>
<tr>
<th>ols_prep_avplot_data</th>
<th>Added variable plot data</th>
</tr>
</thead>
</table>

**Description**

Data for generating the added variable plots.

**Usage**

```r
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  
\textit{Cooks' D plot data}

Description
Prepare data for cook’s \( \delta \) bar plot.

Usage
ols_prep_cdplot_data(model)

Arguments
model An object of class lm.

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

ols_prep_cdplot_outliers  
\textit{Cooks' d outlier data}

Description
Outlier data for cook’s \( \delta \) bar plot.

Usage
ols_prep_cdplot_outliers(k)

Arguments
k Cooks’ \( \delta \) bar plot data.
Examples

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

```r
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)
```

Description

Prepares the data for dfbetas plot.

Usage

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

Arguments

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

Examples

```r
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)
```

Description

Data for identifying outliers in dfbetas plot.

Usage

```r
ols_prep_dfbeta_outliers(d)
```
Arguments

d       A tibble or data.frame.

Examples

```r
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

```r
ols_prep_dsrvf_data(model)
```

Arguments

```r
model       An object of class lm.
```

Examples

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

---

**ols_prep_outlier_obs**  
*Cooks' D outlier observations*

Description

Identify outliers in cook's d plot.

Usage

```r
ols_prep_outlier_obs(k)
```
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)

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, i)

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 residual vs leverage plot.

**Usage**
`ols_prep_rstudlev_data(model)`

**Arguments**
- `model`: An object of class `lm`.

**Examples**
```r
model <- lm(read ~ write + math + science, data = hsb)
ols_prep_rstudlev_data(model)
```

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**
```r
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)

Arguments
model An object of class lm.

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

ols_prep_srplot_data  Studentized residual plot data

Description
Generates data for studentized residual plot.

Usage
ols_prep_srplot_data(model)

Arguments
model An object of class lm.

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


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

```r
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
**ols_regress**

Ordinary least squares regression

**Description**

Ordinary least squares regression.

**Usage**

```r
ols_regress(object, ...)
```

```r
# 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.

**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**


**Examples**

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

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

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

Examples

```r
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**

```r
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**

\[ AIC = -2(\text{loglikelihood}) + \ln(n) \times 2p \]

**SAS**

\[ AIC = n \times \ln(\text{SSE}/n) + p \times \ln(n) \]

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


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_sbic(model)

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

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

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


See Also

Other model selection criteria: \texttt{ols_aic}, \texttt{ols_apc}, \texttt{ols_fpe}, \texttt{ols_hsp}, \texttt{ols_mallows_cp}, \texttt{ols_msep}, \texttt{ols_sbc}

Examples

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

Description

All possible regression

Usage

\texttt{ols_step_all_possible(model, ...)}

## S3 method for class 'ols_step_all_possible'
\texttt{plot(x, model = NA, print_plot = TRUE, ...)}
Arguments

model    An object of class lm.
...     Other arguments.
x    An object of class ols_best_subset.
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:

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

Deprecated Function

ols_all_subset() has been deprecated. Instead use ols_step_all_possible().

References


See Also

Other variable selection procedures: ols_step_backward_aic, ols_step_backward_p, ols_step_best_subset, ols_step_both_aic, ols_step_forward_aic, ols_step_forward_p
Examples

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

# plot
plot(k)
```

---

**ols_step_all_possible_betas**

*All possible regression variable coefficients*

**Description**

Returns the coefficients for each variable from each model.

**Usage**

```r
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**

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

## End(Not run)
```
ols_step_backward_aic

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, progress = FALSE,
details = FALSE, ...)

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

Arguments

model An object of class lm; the model should include all candidate predictor variables.

... Other arguments.

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_aic.

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

Value

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

model model with the least AIC; an object of class lm
steps total number of steps
predictors variables removed from the model
aics akaike information criteria
ess error sum of squares
rss regression sum of squares
rsq rsquare
arsq adjusted rsquare
**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**

```r
ols_step_backward_p(model, ...)  
```

## Default S3 method:

```r
ols_step_backward_p(model, prem = 0.3,  
            progress = FALSE, details = FALSE, ...)  
```

## S3 method for class 'ols_step_backward_p'

```r
plot(x, model = NA, print_plot = TRUE,  
    ...)  
```

**Deprecated Function**

`ols_stepaic_backward()` has been deprecated. Instead use `ols_step_backward_aic()`.

**References**


**See Also**

Other variable selection procedures: `ols_step_all_possible`, `ols_step_backward_p`, `ols_step_best_subset`, `ols_step_both_aic`, `ols_step_forward_aic`, `ols_step_forward_p`
Arguments

model
An object of class lm; the model should include all candidate predictor variables.

...  Other inputs.

prem  p value; variables with p more than prem will be removed from the model.

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

steps  total number of steps

removed  variables removed from the model

rsquare  coefficient of determination

aic  akaike information criteria

sbc  bayesian information criteria

sbic  sawa's bayesian information criteria

adjr  adjusted r-square

rmse  root mean square error

mallows_cp  mallow's Cp

indvar  predictors

Deprecated Function

ols_step_backward() has been deprecated. Instead use ols_step_backward_p().

References


See Also

Other variable selection procedures: ols_step_all_possible, ols_step_backward_aic, ols_step_best_subset, ols_step_both_aic, ols_step_forward_aic, ols_step_forward_p
Examples

```r
# 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)

# final model
k$model
```

## Description

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

## Usage

```r
ols_step_best_subset(model, ...)
```

### S3 method for class 'ols_step_best_subset'

```r
plot(x, model = NA, print_plot = TRUE, ...
```

## Arguments

- `model`: An object of class `lm`.
- `...`: Other inputs.
- `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 data frame containing the following components:

- `n`: model number
- `predictors`: predictors in the model
- `rsquare`: rsquare of the model
- `adjr`: adjusted rsquare of the model
ols_step_both_aic

predictedsq  predicted rsquare of the model
cp  mallow's Cp
aic  akaike information criteria
s bic  sawa bayesian information criteria
sbc  schwarz bayesian information criteria
gmse p  estimated MSE of prediction, assuming multivariate normality
jp  final prediction error
pc  amemiya prediction criteria
sp  hocking's Sp

Deprecated Function

ols_best_subset() has been deprecated. Instead use ols_step_best_subset().

References


See Also

Other variable selection procedures: ols_step_all_possible, ols_step_backward_aic, ols_step_backward_p, ols_step_both_aic, ols_step_forward_aic, ols_step_forward_p

Examples

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

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

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

```r
ols_step_both_aic(model, progress = FALSE, details = FALSE)

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

Arguments

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

Value

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

- `model`: model with the least AIC; an object of class `lm`
- `predictors`: variables added/removed from the model
- `method`: addition/deletion
- `aics`: akaike information criteria
- `ess`: error sum of squares
- `rss`: regression sum of squares
- `rsq`: rsquare
- `arsq`: adjusted rsquare
- `steps`: total number of steps

Deprecated Function

`ols_stepaic_both()` has been deprecated. Instead use `ols_step_both_aic()`.

References


See Also

Other variable selection procedures: `ols_step_all_possible, ols_step_backward_aic, ols_step_backward_p, ols_step_best_subset, ols_step_forward_aic, ols_step_forward_p`
Examples

```r
## 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)

# final model
k$model
```

## 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**

```r
ols_step_both_p(model, ...) 
```

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

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

**Arguments**

- `model` An object of class `lm`; the model should include all candidate predictor variables.
- `...` Other arguments.
- `pent` p value; variables with p value less than `pent` will enter into the model.
- `prem` p value; variables with p more than `prem` will be removed from the model.
- `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.
ols_step_both_p

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

Deprecated Function

`ols_stepwise()` has been deprecated. Instead use `ols_step_both_p()`.

References


Examples

```r
# 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)

# final model
k$model
```
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, progress = FALSE, 
                      details = FALSE, ...)

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

Arguments

model  An object of class lm.
...

Other arguments.

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

Value

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

model  model with the least AIC; an object of class lm
steps  total number of steps
predictors  variables added to the model
aics  akaike information criteria
ess  error sum of squares
rss  regression sum of squares
rsq  rsquare
arsq  adjusted rsquare

Deprecated Function

ols_stepaic_forward() has been deprecated. Instead use ols_step_forward_aic().
References


See Also

Other variable selection procedures: ols_step_all_possible, ols_step_backward_aic, ols_step_backward_p, ols_step_best_subset, ols_step_both_aic, ols_step_forward_p

Examples

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

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

# final model
k$model

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, penter = 0.3,
    progress = FALSE, details = FALSE, ...)

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

ols_step_forward_p  Stepwise forward regression
**Arguments**

- **model**: An object of class `lm`; the model should include all candidate predictor variables.
- **...**: Other arguments.
- **penter**: p value; variables with p value less than `penter` will enter into the model.
- **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`
- **steps**: number of steps
- **predictors**: variables added to the model
- **rsquare**: coefficient of determination
- **aic**: akaike information criteria
- **sbc**: bayesian information criteria
- **sbic**: sawa’s bayesian information criteria
- **adjr**: adjusted r-square
- **rmse**: root mean square error
- **mallows_cp**: mallow’s Cp
- **indvar**: predictors

**Deprecated Function**

`ols_step_forward()` has been deprecated. Instead use `ols_step_forward_p()`.

**References**


**See Also**

Other variable selection procedures: `ols_step_all_possible, ols_step_backward_aic, ols_step_backward_p, ols_step_best_subset, ols_step_both_aic, ols_step_forward_aic`
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)

# final model
k$model

---

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
Deprecation Function

ols_bartlett_test() has been deprecated. Instead use ols_test_bartlett().

References


See Also

Other heteroskedasticity tests: ols_test_breusch_pagan, ols_test_f, ols_test_score

Examples

# using grouping variable
library(descriptr)
ols_test_bartlett(mtcarz, 'mpg', group_var = 'cyl')

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

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

Deprecated Function

`ols_bp_test()` has been deprecated. Instead use `ols_test_breusch_pagan()`.

References


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')

Description

Correlation test for 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.

Deprecated Function

ols_corr_test() has been deprecated. Instead use ols_test_correlation().

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**

```r
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**

```r
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 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

**Deprecated Function**

`ols_f_test()` has been deprecated. Instead use `ols_test_f()`.
References


See Also

Other heteroskedasticity tests: `ols_test_bartlett, ols_test_breusch_pagan, ols_test_score`

Examples

```r
# 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**

```r
ols_test_normality(y, ...)
```

## S3 method for class 'lm'

```r
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

Deprecated Function

`ols_norm_test()` has been deprecated. Instead use `ols_test_normality()`.

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

```r
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

```r
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

```r
# 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

Deprecated Function

`ols_score_test()` has been deprecated. Instead use `ols_test_score()`.

References


See Also

Other heteroskedasticity tests: \texttt{ols_test_bartlett}, \texttt{ols_test_breusch_pagan}, \texttt{ols_test_f}

Examples

\begin{verbatim}
# 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'))
\end{verbatim}

---

rivers

\textit{Test Data Set}

Description

Test Data Set

Usage

rivers

Format

An object of class \texttt{data.frame} with 20 rows and 6 columns.

---

rvsr_plot_shiny

\textit{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

\texttt{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
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

Index

* datasets
  auto, 3
cement, 4
fitness, 4
hsb, 4
rivers, 71
stepdata, 72
surgical, 72
* heteroskedasticity tests
  ols_test_bartlett, 63
  ols_test_breusch_pagan, 64
  ols_test_f, 67
  ols_test_score, 70
* influence measures
  ols_hadi, 11
  ols_leverage, 13
  ols_pred_rsq, 35
  ols_press, 43
* model selection criteria
  ols_aic, 5
  ols_apc, 6
  ols_fpe, 10
  ols_hsp, 12
  ols_mallows_cp, 14
  ols_msep, 15
  ols_sbc, 47
  ols_sbic, 48
* residual diagnostics
  ols_plot_resid_box, 25
  ols_plot_resid_fit, 26
  ols_plot_resid_hist, 28
  ols_plot_resid_qq, 30
  ols_test_correlation, 66
  ols_test_normality, 68
* variable selection procedures
  ols_step_all_possible, 49
  ols_step_backward_aic, 52
  ols_step_backward_p, 53
  ols_step_best_subset, 55
  ols_step_both_aic, 56
  ols_step_forward_aic, 60
  ols_step_forward_p, 61
* variable selection procedures
  ols_step_both_p, 58
  auto, 3
cement, 4
fitness, 4
hsb, 4
ols_aic, 5, 7, 11, 12, 14, 15, 48, 49
ols_all_subset (ols_step_all_possible), 49
ols_all_subset_betas
  (ols_step_all_possible_betas), 51
ols_apc, 6, 11, 12, 14, 15, 48, 49
ols_avplots (ols_plot_added_variable), 16
ols_bartlett_test (ols_test_bartlett), 63
ols_best_subset (ols_step_best_subset), 55
ols_bp_test (ols_test_breusch_pagan), 64
ols_coll_diag, 7
ols_cooksd_barplot
  (ols_plot_cooksd_bar), 18
ols_cooksd_chart
  (ols_plot_cooksd_chart), 19
ols_corr_test (ols_test_correlation), 66
ols_correlations, 9
ols_dfbetas_panel (ols_plot_dfbetas), 20
ols_dffits_plot (ols_plot_dffits), 21
ols_diagnostic_panel
  (ols_plot_diagnostics), 22
ols_dsrvsp_plot
  (ols_plot_resid_stud_fit), 33
INDEX

ols_eigen_cindex (ols_coll_diag), 7
ols_f_test (ols_test_f), 67
ols_fm_plot
  (ols_plot_resid_fit_spread), 27
ols_fpe, 6, 7, 10, 12, 14, 15, 48, 49
ols_hadi, 11, 13, 35, 43
ols_hadi_plot (ols_plot_hadi), 23
ols_hsp, 6, 7, 11, 12, 14, 15, 48, 49
ols_launch_app, 13
ols_leverage, 11, 13, 35, 43
ols_mallows_cp, 6, 7, 11, 12, 14, 15, 48, 49
ols_msep, 6, 7, 11, 12, 14, 15, 48, 49
ols_norm_test (ols_test_normality), 68
ols_ovsp_plot (ols_plot_obs_fit), 24
ols_plot_added_variable, 16
ols_plot_comp_plus_resid, 17
ols_plot_cooksd_bar, 18
ols_plot_cooksd_chart, 19
ols_plot_dfbetas, 20
ols_plot_dffits, 21
ols_plot_diagnostics, 22
ols_plot_fm
  (ols_plot_resid_fit_spread), 27
ols_plot_hadi, 23
ols_plot_obs_fit, 24
ols_plot_reg_line, 24
ols_plot_resid_box, 25, 26, 28, 30, 66, 69
ols_plot_resid_fit, 25, 26, 28, 30, 66, 69
ols_plot_resid_fit_spread, 27
ols_plot_resid_hist, 25, 26, 28, 30, 66, 69
ols_plot_resid_level, 28
ols_plot_resid_pot, 29
ols_plot_resid_qq, 25, 26, 28, 30, 66, 69
ols_plot_resid_regressor, 31
ols_plot_resid_spread
  (ols_plot_resid_fit_spread), 27
ols_plot_resid_stand, 31
ols_plot_resid_study, 32
ols_plot_reg_line, 33
ols_plot_response, 34
ols_potrsd_plot (ols_plot_resid_pot), 29
ols_pred_rsq, 11, 13, 35, 43
ols_prep_avplot_data, 35
ols_prep_cdplot_data, 36
ols_prep_cdplot_outliers, 36
ols_prep_dfbeta_data, 37
ols_prep_dfbeta_outliers, 37
ols_prep_dsrvf_data, 38
ols_prep_outlier_obs, 38
ols_prep_regress_x, 39
ols_prep_regress_y, 39
ols_prep_rfsplot_fmdata, 40
ols_prep_rfsplot_rsddata
  (ols_prep_rfsplot_fmdata), 40
ols_prep_rstudlev_data, 41
ols_prep_rvrsplot_data, 41
ols_prep_srchart_data, 42
ols_prep_srsrplot_data, 42
ols_press, 11, 13, 35, 43
ols_pure_error_anova, 44
ols_reg_line (ols_plot_reg_line), 24
ols_regress, 45
ols_resp_viz (ols_plot_response), 34
ols_rfs_plot
  (ols_plot_resid_fit_spread), 27
ols_rpc_plot
  (ols_plot_comp_plus_resid), 17
ols_rsd_boxplot (ols_plot_resid_box), 25
ols_rsd_hist (ols_plot_resid_hist), 28
ols_rsd_plot
  (ols_plot_resid_fit_spread), 27
ols_rsd_qqplot (ols_plot_resid_qq), 30
ols_rsdlev_plot (ols_plot_resid_level), 28
ols_rsvp_plot (ols_plot_resid_fit), 26
ols_rvrs_plot
  (ols_plot_resid_regressor), 31
ols_sbc, 6, 7, 11, 12, 14, 15, 47, 49
ols_sbic, 6, 7, 11, 12, 14, 15, 48, 48
ols_score_test (ols_test_score), 70
ols_srsr_chart (ols_plot_resid_stand), 31
ols_srsr_plot (ols_plot_resid_stud), 32
ols_step_all_possible, 49, 53, 54, 56, 57, 61, 62
ols_step_all_possible_betas, 51
ols_step_backward
  (ols_step_backward_p), 53
ols_step_backward_aic, 50, 52, 54, 56, 57, 61, 62
ols_step_backward_p, 50, 53, 56, 57, 61, 62
ols_step_best_subset, 50, 53, 54, 55, 57, 61, 62
ols_step_both_aic, 50, 53, 54, 56, 57, 61, 62
ols_step_both_p, 58
ols_step_forward (ols_step_forward_p),
ols_step_forward_aic, 50, 53, 54, 56, 57, 60, 62
ols_step_forward_p, 50, 53, 54, 56, 57, 61, 61
ols_stepaic_backward
   (ols_step_backward_aic), 52
ols_stepaic_both (ols_step_both_aic), 56
ols_stepaic_forward
   (ols_step_forward_aic), 60
ols_stepwise (ols_step_both_p), 58
ols_test_bartlett, 63, 65, 68, 71
ols_test_breusch_pagan, 64, 64, 68, 71
ols_test_correlation, 25, 26, 28, 30, 66, 69
ols_test_f, 64, 65, 67, 71
ols_test_normality, 25, 26, 28, 30, 66, 68
ols_test_outlier, 69
ols_test_score, 64, 65, 68, 70
ols_vif_tol (ols_coll_diag), 7
olsrr, 5
olsrr-package (olsrr), 5
plot.ols_step_all_possible
   (ols_step_all_possible), 49
plot.ols_step_backward_aic
   (ols_step_backward_aic), 52
plot.ols_step_backward_p
   (ols_step_backward_p), 53
plot.ols_step_best_subset
   (ols_step_best_subset), 55
plot.ols_step_both_aic
   (ols_step_both_aic), 56
plot.ols_step_both_p (ols_step_both_p), 58
plot.ols_step_forward_aic
   (ols_step_forward_aic), 60
plot.ols_step_forward_p
   (ols_step_forward_p), 61
rivers, 71
rvsr_plot_shiny, 71
stepdata, 72
surgical, 72