Package ‘lineartestr’

October 13, 2022

Type Package
Title Linear Specification Testing
Version 1.0.0
Maintainer Federico Garza <fede.garza.ramirez@gmail.com>
Description Tests whether the linear hypothesis of a model is correct specified using Dominguez-Lobato test. Also Ramsey’s RESET (Regression Equation Specification Error Test) test is implemented and Wald tests can be carried out. Although RESET test is widely used to test the linear hypothesis of a model, Dominguez and Lobato (2019) proposed a novel approach that generalizes well known specification tests such as Ramsey’s. This test relies on wild-bootstrap; this package implements this approach to be usable with any function that fits linear models and is compatible with the update() function such as 'stats':::lm(), 'lfe':::felm() and 'forecast':::Arima(), for ARMA (autoregressive–moving-average) models. Also the package can handle custom statistics such as Cramer von Mises and Kolmogorov Smirnov, described by the authors, and custom distributions such as Mammen (discrete and continuous) and Rademacher.


License GPL (>= 2)
Encoding UTF-8
Depends R (> 3.2)
LazyData true
Imports Matrix, sandwich, dplyr, ggplot2, viridis, tidyr, readr, parallel, forecast
Suggests testthat
RoxygenNote 7.0.2
URL https://github.com/FedericoGarza/lineartestr
BugReports https://github.com/FedericoGarza/lineartestr/issues
NeedsCompilation no
Author Federico Garza [aut, cre] (<https://orcid.org/0000-0001-7015-8186>)
Repository CRAN
Date/Publication 2020-06-12 09:10:20 UTC
dominguez_lobato_test

Tests the specification of a linear model using wild-bootstrap.

Description

Tests the specification of a linear model using wild-bootstrap.

Usage

dominguez_lobato_test(
  model,
  distribution = "rnorm",
  statistic = "cvm_value",
  times = 300,
  quantiles = c(0.9, 0.95, 0.99),
  verbose = FALSE,
  n_cores = 1
)

Arguments

  model      An existing fit from a model function such as ‘lm’, ‘lfe’ and others compatible with ‘update’.
  distribution Type of noise added to residuals, ej ‘rnorm’ or ‘rrademacher’.
  statistic Type of statistic to be used, can be one of ‘cvm_value’ or ‘kmv_value’.
  times Number of bootstrap samples.
  quantiles Vector of quantiles to calculate pvalues.
  verbose TRUE to print each bootstrap iteration.
  n_cores Number of cores to be used.
Value

A list with dataframe results and the ordered values of each bootstrap iteration.

References


Examples

```r
x <- 1:10 + rnorm(10)
y <- 1:10
model <- lm(y~x)
dl_test <- dominguez_lobato_test(model)
dl_test <- dominguez_lobato_test(model, distribution = "rmammen_point", statistic = "kmv_value")
dl_test <- dominguez_lobato_test(model, times = 100)
```

```
plot_dl_test
Plots the Dominguez-Lobato test.
```

Description

Plots the Dominguez-Lobato test.

Usage

```r
plot_dl_test(x)
```

Arguments

- `x`: An object of class 'dl_test'.

Value

Plot of type ggplot.

Examples

```r
x <- 1:10
y <- 1:10
model <- lm(y~x-1)
dl_test <- dominguez_lobato_test(model)
plot_dl_test(dl_test)
```
plot_reset_test

Plot the reset test.

Description

Plot the reset test.

Usage

plot_reset_test(x)

Arguments

x

An object of class ‘reset_test’.

Value

Plot of type ggplot.

Examples

x <- 1:10 + rnorm(10)
y <- 1:10
model <- lm(y~x-1)
r_test <- reset_test(model)
plot_reset_test(r_test)

presiduals

Calculates the accumulated distribution of residuals at each residual point.

Description

Calculates the accumulated distribution of residuals at each residual point.

Usage

presiduals(fitted_values, resids)

Arguments

fitted_values

Vector of fitted values.

resids

Residuals vector of each fitted value.

Value

Vector of size length(resids).
reset_test

Examples

```r
y_hat <- c(4, 8, 7)
resids <- c(1, 5, 3)
presiduals(y_hat, resids)
```

reset_test

*Reset test. Tests the specification of a linear model adding and testing powers of fitted values.*

Description

Reset test. Tests the specification of a linear model adding and testing powers of fitted values.

Usage

```r
reset_test(
  model,  # An existing fit from a model function such as 'lm', 'lfe' and others compatible with 'update'.
  robust = FALSE,  # Use robust 'varcov' matrix.
  vcov = NULL,  # Particular variance and covariances matrix.
  max_power = 3,  # Max power of fitted values to add.
  quantiles = c(0.9, 0.95, 0.99)  # Vector of quantiles to calculate pvalues.
)
```

Arguments

- `model`: An existing fit from a model function such as `lm`, `lfe` and others compatible with `update`.
- `robust`: Use robust `varcov` matrix.
- `vcov`: Particular variance and covariances matrix.
- `max_power`: Max power of fitted values to add.
- `quantiles`: Vector of quantiles to calculate pvalues.

Value

A `tibble` with the Wald value, the corresponding pvalue, and the quantiles of the distribution.

Examples

```r
x <- 1:10 + rnorm(10)
y <- 1:10
model <- lm(y~x)
r_test <- reset_test(model)
r_test <- reset_test(model, robust = TRUE)
r_test <- reset_test(model, quantiles = c(.97))
r_test <- reset_test(model, max_power = 4)
r_test <- reset_test(model, robust = TRUE, max_power = 4)
```
**random_cont**

Random deviates of Mammen continuous distribution.

**Usage**

```r
random_cont(n)
```

**Arguments**

- `n` Number of observations.

**Value**

Random deviates of size n.

**Examples**

```r
random_cont(10)
```

---

**random_point**

Random deviates of Mammen distribution.

**Usage**

```r
random_point(n)
```

**Arguments**

- `n` Number of observations.

**Value**

Random deviates of size n.

**Examples**

```r
random_point(10)
```
**rrademacher**

Random deviates of Rademacher distribution.

**Usage**

`rrademacher(n)`

**Arguments**

- `n` Number of observations.

**Value**

Random deviates of size n.

**Examples**

`rrademacher(10)`

---

**statistic_value**

Calculates the Cramer von Mises value or Kolmogorov value given a linear model compatible with `fitted.values` and `residuals` functions.

**Description**

Calculates the Cramer von Mises value or Kolmogorov value given a linear model compatible with `fitted.values` and `residuals` functions.

**Usage**

`statistic_value(model, value = "cvm_value")`

**Arguments**

- `model` An existing fit from a linear model function.
- `value` Type of value to compute, can be `cvm_value` or `kmv_value`.

**Value**

The statistic value of the model.
**Examples**

```r
dx <- 1:10
y <- 2*x + rnorm(10)
model <- lm(y~x-1)
statistic_value(model)
statistic_value(model, value = "cvm_value")
statistic_value(model, value = "kmv_value")
```

---

**updated_model**

*Constructs a new model with noised residuals: y_new = y_fitted + residuals*noise*

**Description**

Constructs a new model with noised residuals: $y_{new} = y_{fitted} + \text{residuals} \ast \text{noise}$

**Usage**

```r
updated_model(model, fitting_data, distribution = "rnorm")
```

**Arguments**

- **model**: An existing fit from a model function such as ‘lm’, ‘lfe’, ‘Arima’ and others compatible with ‘update’.
- **fitting_data**: Data used to adjust a linear model.
- **distribution**: Type of noise added to residuals, e.g. "rnorm" or "rrademacher".

**Value**

Constructed linear model.

**Examples**

```r
x <- 1:100
y <- 2*x + rnorm(100)
model <- lm(y~x-1)
fitting_data <- model.frame(model)
updated_model(model, fitting_data)
updated_model(model, fitting_data, distribution = "rnorm")
updated_model(model, fitting_data, distribution = "rmammen_point")
updated_model(model, fitting_data, distribution = "rmammen_cont")
updated_model(model, fitting_data, distribution = "rrademacher")
```

```r
x_arma <- rnorm(100)
arma_model <- forecast::Arima(x_arma, c(1, 0, 1))
fitting_data_arma <- model.frame(arma_model)
updated_model(arma_model, fitting_data_arma)
```
wald_test

Wald test. Tests restrictions*coefficients = value.

Description
Wald test. Tests restrictions*coefficients = value.

Usage
wald_test(
  model,
  restrictions,
  value,
  robust = FALSE,
  vcov = NULL,
  quantiles = c(0.9, 0.95, 0.99)
)

Arguments
model Model compatible with ‘fitted’ and ‘residuals’ functions.
restrictions Matrix of size (number of restrictions) times length(coefficients), for free restrictions use zeros.
value Values of restrictions.
robust Use robust ‘varcov’ matrix.
vcov Particular variance and covariances matrix.
quantiles Vector of quantiles to calculate pvalues.

Value
A ‘tibble’ with the Wald value, the corresponding pvalue and the quantiles of the distribution.

Examples
x <- 1:10
z <- x**2
y <- 1:10
model <- lm(y-x+z)
restrictions <- diag(3)
value <- as.matrix(c(0, 0, 0))
w_test <- wald_test(model, restrictions, value)
w_test <- wald_test(model, restrictions, value, robust = TRUE)
w_test <- wald_test(model, restrictions, value, quantiles = c(.97))
Index

dominguez_lobato_test, 2
plot_dl_test, 3
plot_reset_test, 4
presiduals, 4
reset_test, 5
rmammen_cont, 6
rmammen_point, 6
rrademacher, 7
statistic_value, 7
updated_model, 8
wald_test, 9