Package 'lmtestrob'

May 23, 2023

Type Package

Title Outlier Robust Specification Testing

Version 0.1

Date 2023-05-22

Author Mikhail Zhelonkin [aut, cre] (<https://orcid.org/0000-0002-6912-4074>)

Maintainer Mikhail Zhelonkin <Mikhail.Zhelonkin@gmail.com>

Description Robust test(s) for model diagnostics in regression. The current version contains a robust test for functional specification (linearity). The test is based on the robust bounded-influence test by Heritier and Ronchetti (1994) <doi:10.1080/01621459.1994.10476822>.

License GPL-2

Imports MASS

Suggests lmtest

NeedsCompilation no

Repository CRAN

Date/Publication 2023-05-23 08:50:05 UTC

R topics documented:

`lmtestrob-package` *Outlier Robust Specification Testing*

Description

The package contains an outlier robust functional miss-specification test.

Details

Package:	lmtestrob
Type:	Package
Version:	0.1
Date:	2023-05-22
License:	GPL-2

Author(s)

Mikhail Zhelonkin (aut, cre).

Maintainer: Mikhail Zhelonkin <Mikhail.Zhelonkin@gmail.com>

References

Heritier, S., and Ronchetti, E. (1994) Robust Bounded-Influence Tests in General Parametric Models. *Journal of the American Statistical Association*, 89, p. 897-904.

Zeileis, A., and Hothorn, T. (2002) Diagnostic Checking in Regression Relationships. *R News*, 2, p. 7-10.

See Also

`robfmtest`

`print.robfmtest` *Print a* `robfmtest` *Object*

Description

Print an object generated by `robfmtest`

Usage

```
## S3 method for class 'robfmtest'
print(x, digits = 4, ...)
```

Arguments

`x`	object returned from the `robfmtest`.
`digits`	number of significant digits to be printed.
`...`	currently not used.

Value

No return value.

Author(s)

Mikhail Zhelonkin

See Also

robfmtest

---

robfmtest *Robust Functional Specification Test*

Description

Outlier robust test for functional miss-specification. It can be used to test linearity. The test is based on the robust Wald-type test by Heritier and Ronchetti (1994).

Usage

```
robfmtest(formula, power = 2:3, type = c("regressor"), data,
          x.weights = c("HAT", "MCD"), testtype = "Wald", ...)
```

Arguments

formula	a symbolic description of the model to be tested.
power	integer(s). A vector of positive integers specifying the powers of the variables that should be tested. The default option tests second and third powers.
type	currently, only powers of regressors can be used.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which robfmtest is called.
x.weights	a string, indicating how the robustness weights on the covariates should be computed. The default option uses hat-matrix-based weights, second option allows to use robust Mahalanobis distance-based weights, where the Minimum Covariance Determinant is used to estimate location and scatter.
testtype	currently, the robust version of Wald test is implemented.
...	currently not used.

Details

Since the classical tests including resettest, raintest and harvtest implemented in lmtest are not resistant to outliers and can become misleading even in the presence of one outlier, we provide a test which is resistant to outliers. The price to pay for robustness is a small loss of power, when the model holds exactly.

Value

A list with class robfmtest containing the following components:

`statistic`	the value of the test statistic.
`dof`	the number of degrees of freedom.
`method`	a character string indicating what type of test was performed.
`p.value`	the p-value of the test.
`data.name`	a character string giving the name(s) of the data.

Author(s)

Mikhail Zhelonkin

References

Heritier, S., and Ronchetti, E. (1994) Robust Bounded-Influence Tests in General Parametric Models. *Journal of the American Statistical Association*, 89, p. 897-904.

Examples

```
set.seed(123)
n <- 50
x = runif(n, -3, 3)
y = rnorm(n)
example.dat <- data.frame(x, y)
robfmtest(y ~ x, data = example.dat)
library(lmtest)
resettest(y ~ x, data = example.dat, type = "fitted")
x[50] <- -3
y[50] <- -10
example.dat <- data.frame(x, y)
robfmtest(y ~ x, data = example.dat)
resettest(y ~ x, data = example.dat, type = "fitted")
```

Index