Package ‘MTest’

September 11, 2025

Type Package

Title A Procedure for Multicollinearity Testing using Bootstrap

Version 1.0.4

Date 2025-09-09

Maintainer Víctor Morales-Oñate <vmorales.ppb@gmail.com>

Description
Functions for detecting multicollinearity. This test gives statistical support to two of the most famous methods for detecting multicollinearity in applied work: Klein’s rule and Variance Inflation Factor (VIF). See the URL for the papers associated with this package, as for instance, Morales-Oñate and Morales-Oñate (2015) <doi:10.33333/rp.vol51n2.05>.

Depends R (>= 4.1.0)

License GPL (>= 3)

Encoding UTF-8

Imports ggplot2,plotly

Repository CRAN

URL https://github.com/vmoprojs/MTest

BugReports https://github.com/vmoprojs/MTest/issues

LazyData true

NeedsCompilation no

Author Víctor Morales-Oñate [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1922-6571>),
Bolívar Morales-Oñate [aut] (ORCID:
<https://orcid.org/0000-0003-4980-8759>)

Date/Publication 2025-09-11 10:00:23 UTC

Contents

MTest 2
pairwiseKStest 4
plot.MTest 6
simDataMTest 7

Index **8**

`MTest` *Bootstrap-based test for multicollinearity (Klein and VIF)*

Description

`MTest` implements a nonparametric (pairs) bootstrap to assess multicollinearity by providing achieved significance levels (ASL) for two widely used diagnostics: Klein's rule and the Variance Inflation Factor (VIF). It returns bootstrap distributions of the global R^2 and the auxiliary R_j^2 (from regressions of each predictor on the remaining predictors), along with p-values for both rules.

Usage

```
MTest(object, nboot = 100, nsam = NULL, trace = FALSE, seed = NULL,
      valor_vif = 0.9)
```

Arguments

`object`	A fitted model, typically of class `"lm"`. The function uses its `model.frame` and `model.matrix` as the fixed design for resampling.
`nboot`	Integer. Number of bootstrap iterations (rows resampled with replacement).
`nsam`	Integer. Bootstrap sample size per iteration. Defaults to the original number of rows.
`trace`	Logical. If `TRUE`, shows a progress bar.
`seed`	Integer. Optional RNG seed for reproducibility.
`valor_vif`	Numeric in $[0, 1)$. Threshold applied to R_j^2 for the VIF rule: the ASL is $P(R_j^2 > c)$, where $c =$ `valor_vif`. Since $\text{VIF}_j = 1/(1 - R_j^2)$, `valor_vif = 0.9` corresponds roughly to `VIF >= 10`.

Details

Model. Consider the linear regression model

$$Y_i = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_p X_{pi} + u_i, \quad i = 1, \ldots, n,$$

and the *auxiliary* regressions obtained by regressing each predictor X_j on the remaining predictors X_{-j}. Let R_g^2 be the global coefficient of determination and R_j^2 the coefficient of determination of the j-th auxiliary regression.

Diagnostics and achieved significance levels (ASL).

- *Klein's rule*: flag multicollinearity if $R_j^2 > R_g^2$. We estimate the ASL as $P(R_g^2 < R_j^2)$ using the bootstrap distribution.
- *VIF rule*: flag multicollinearity if `VIF` exceeds a threshold. Since $\text{VIF}_j = 1/(1 - R_j^2)$, this is equivalent to testing R_j^2 against `valor_vif`. We estimate $P(R_j^2 >$ `valor_vif`$)$.

Bootstrap scheme. The function resamples rows of the model frame (pairs bootstrap) and, for each bootstrap sample, computes R_g^2 and R_j^2 (hence VIF) using the same expanded design matrix as the original fit. This makes the procedure robust to transformed terms on either side of the formula (e.g., `log(y)`, `I(X1^2)`, interactions, factors, `poly()`, etc.).

Value

An object of class `MTest`, which is a list containing:

`pval_vif`	Named numeric vector of ASL for the VIF rule, $P(R_j^2 > \texttt{valor\_vif})$.
`pval_klein`	Named numeric vector of ASL for Klein's rule, $P(R_g^2 < R_j^2)$.
`Bvals`	Numeric matrix of size `nboot x (p+1)` with columns `"global"` for R_g^2 and one column per predictor for R_j^2.
`VIFvals`	Numeric matrix `nboot x p` with VIF values per predictor (one column per design column).
`vif.tot`	Observed VIF per predictor from the original design.
`R.tot`	Named numeric vector with observed R_g^2 and all R_j^2.
`nsam`	Bootstrap sample size actually used.
`nboot`	Number of bootstrap iterations actually performed.

Interpretation

- Larger `pval_klein[j]` indicates stronger evidence that predictor j violates Klein's rule (R_j^2 often exceeds R_g^2).
- Larger `pval_vif[j]` indicates that R_j^2 frequently exceeds `valor_vif` (equivalently, `VIF` exceeds the implied threshold).

Notes

For factor predictors, the underlying design includes multiple columns; `VIFvals` and VIF-related summaries are returned *per design column*. In singular bootstrap samples some statistics may be `NA`.

Author(s)

Víctor Morales Oñate <`vmorales.ppb@gmail.com`>
Bolívar Morales Oñate <`bmoralesonate@gmail.com`>
`https://sites.google.com/site/moralesonatevictor/`
`https://www.linkedin.com/in/vmoralesonate/`

References

Morales-Oñate, V., and Morales-Oñate, B. (2023). *MTest: a Bootstrap Test for Multicollinearity*. Revista Politécnica, 51(2), 53–62. doi:10.33333/rp.vol51n2.05

See Also

`vif` for classical VIF computation.

Examples

```
## Minimal example (small nboot for speed)
set.seed(1)
data(simDataMTest, package = "MTest")
m1 <- stats::lm(y ~ ., data = simDataMTest)

boot.sol <- MTest(m1, nboot = 50, trace = FALSE, seed = 123, valor_vif = 0.90)

boot.sol$pval_vif
boot.sol$pval_klein
head(boot.sol$Bvals)
print(boot.sol)
```

`pairwiseKStest` *Pairwise Kolmogorov–Smirnov tests for matrix columns*

Description

Computes pairwise Kolmogorov–Smirnov (KS) tests between all columns of a numeric matrix or data frame, returning the matrix of p-values. Typical inputs include the `Bvals` matrix from `MTest`. For one-sided alternatives (`"greater"` or `"less"`), the p-value matrix is directional: rows correspond to `x` and columns to `y`.

Usage

```
pairwiseKStest(X,
               alternative = c("greater","less","two.sided"),
               use = c("asis","pairwise.complete.obs"),
               exact = NULL)
```

Arguments

`X`	Numeric matrix or data frame. Columns are compared pairwise by KS tests. A common use is `Bvals` from `MTest`. If you want to compare only predictors, pass `Bvals[ , -1]` to exclude the `"global"` column.
`alternative`	Character string: `"greater"`, `"less"`, or `"two.sided"`. See `ks.test` for the meaning of these options.
`use`	Character string: `"asis"` (default; no NA filtering, replicates the original behavior), or `"pairwise.complete.obs"` (remove NAs pairwise for each test).
`exact`	Logical or `NULL`. Forwarded to `ks.test`. Leave as `NULL` to use the default decision of `ks.test` (as in the original function).

Details

The function performs a KS test for each ordered pair of columns `(i, j)` using `ks.test(X[, i], X[, j], alternative = alternative, exact = exact)`. For one-sided alternatives, the result is not symmetric, since rows play the role of `x` and columns the role of `y`.

The returned `Suggestion` follows the same rule as the original function: for `alternative = "greater"`, it sorts the row sums of the p-value matrix (descending); for `"less"`, it sorts the column sums; for `"two.sided"`, no suggestion is returned.

Value

A list of class `pairwiseKStest` with components:

`KSpwMatrix`	Numeric matrix of p-values. Rows are `x`, columns are `y`.
`alternative`	Character string describing the alternative hypothesis used.
`Suggestion`	For `"greater"`: row sums of `KSpwMatrix` (sorted decreasing). For `"less"`: column sums (sorted decreasing). For `"two.sided"`: a message indicating no suggestions.

Author(s)

Víctor Morales Oñate <`vmorales.ppb@gmail.com`>
Bolívar Morales Oñate <`bmoralesonate@gmail.com`>
`https://sites.google.com/site/moralesonatevictor/`
`https://www.linkedin.com/in/vmoralesonate/`

References

Morales-Oñate, V., and Morales-Oñate, B. (2023). *MTest: a Bootstrap Test for Multicollinearity*. Revista Politécnica, 51(2), 53–62. doi:10.33333/rp.vol51n2.05

See Also

`ks.test`, `MTest`

Examples

```
## Typical workflow with MTest:
## (use small nboot for speed in examples)
set.seed(1)
data(simDataMTest, package = "MTest")
m1 <- stats::lm(y ~ ., data = simDataMTest)
boot.sol <- MTest(m1, nboot = 30, trace = FALSE, seed = 123)

## Compare only predictors (exclude "global"):
ks_res_greater <- pairwiseKStest(boot.sol$Bvals[, -1],
                                 alternative = "greater",
                                 use = "asis",    # same behavior as the original
                                 exact = NULL)    # let ks.test decide

ks_res_greater$KSpwMatrix
```

```
ks_res_greater$Suggestion

## Two-sided (no suggestion by design):
ks_res_twosided <- pairwiseKStest(boot.sol$Bvals[, -1],
                                  alternative = "two.sided")
ks_res_twosided$KSpwMatrix
```

`plot.MTest` *Plot density or empirical cumulative distribution from MTest*

Description

Plot density or empirical cumulative distribution from `Bvals` in `MTest` `output`.

Usage

```
## S3 method for class 'MTest'
plot(x, type=1,plotly = FALSE,...)
```

Arguments

`x`	an object of the class `"MTest"`
`type`	Numeric; 1 if density, 2 if ecdf plot is returned
`plotly`	Logical; if `FALSE`, a `ggplotly` plot is returned
`...`	other arguments to be passed to the function `ggplot`

Details

This function plots density or empirical cumulative distribution function from MTest bootstrap replications.

Value

Produces a plot. No values are returned.

See Also

`MTest` for procedure and examples.

`simDataMTest` *Simulated data for MTest*

Description

This data set helps testing functions in MTest package, the generating process is documented in the reference.

Usage

```
simDataMTest
```

Format

A dataframe containing 10000 observations and four columns.

References

Morales-Oñate, V., and Morales-Oñate, B. (2023). *MTest: a Bootstrap Test for Multicollinearity*. Revista Politécnica, 51(2), 53–62. doi:10.33333/rp.vol51n2.05

Index

∗ **MTest**
    plot.MTest, 6
∗ **bootstrap**
    MTest, 2
∗ **datasets**
    simDataMTest, 7
∗ **ks-test**
    pairwiseKStest, 4
∗ **multicollinearity**
    MTest, 2
    pairwiseKStest, 4
∗ **nonparametric**
    pairwiseKStest, 4
∗ **regression**
    MTest, 2

ggplot, *6*

ks.test, *4*, *5*

MTest, 2, *5*, *6*

pairwiseKStest, 4
plot.MTest, 6

simDataMTest, 7

vif, *3*