Package 'mcvis'

October 13, 2022

Type Package

Title Multi-Collinearity Visualization

Version 1.0.8

Description
Visualize the relationship between linear regression variables and causes of multi-collinearity. Implements the method in Lin et. al. (2020) <doi:10.1080/10618600.2020.1779729>.

Encoding UTF-8

Imports assertthat, igraph, ggplot2, purrr, magrittr, reshape2, shiny, dplyr, psych, rlang

RoxygenNote 7.1.1.9001

License GPL-3

Suggests testthat (>= 2.1.0), covr, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Kevin Wang [aut, cre],
Chen Lin [aut],
Samuel Mueller [aut]

Maintainer Kevin Wang <kevin.wang09@gmail.com>

Repository CRAN

Date/Publication 2021-07-30 08:20:05 UTC

R topics documented:

alt_mcvis *Multi-collinearity Visualization plots*

Description

Multi-collinearity Visualization plots

Multi-collinearity Visualization plots

Multi-collinearity Visualization plots

Usage

```
alt_mcvis(mcvis_result, eig_max = 1L, var_max = ncol(mcvis_result$MC))

ggplot_mcvis(
  mcvis_result,
  eig_max = 1L,
  var_max = ncol(mcvis_result$MC),
  label_dodge = FALSE
)

igraph_mcvis(mcvis_result, eig_max = 1L, var_max = ncol(mcvis_result$MC))

## S3 method for class 'mcvis'
plot(
  x,
  type = c("ggplot", "igraph", "alt"),
  eig_max = 1L,
  var_max = ncol(x$MC),
  label_dodge = FALSE,
  ...
)
```

Arguments

mcvis_result	Output of the mcvis function
eig_max	The maximum number of eigenvalues to be displayed on the plot.
var_max	The maximum number of variables to be displayed on the plot.
label_dodge	If variable names are too long, it might be helpful to dodge the labelling. Default to FALSE.
x	Output of the mcvis function
type	Plotting mcvis result using "igraph" or "ggplot". Default to "ggplot".
...	additional arguments (currently unused)

Value

A mcvis visualization plot

Author(s)

Chen Lin, Kevin Wang, Samuel Mueller

Examples

```
set.seed(1)
p = 10
n = 100
X = matrix(rnorm(n*p), ncol = p)
X[,1] = X[,2] + rnorm(n, 0, 0.1)
mcvis_result = mcvis(X)
plot(mcvis_result)
plot(mcvis_result, type = "igraph")
plot(mcvis_result, type = "alt")
```

`mcvis` *Multi-collinearity Visualization*

Description

Multi-collinearity Visualization

Usage

```
mcvis(
  X,
  sampling_method = "bootstrap",
  standardise_method = "studentise",
  times = 1000L,
  k = 10L
)
```

Arguments

`X`	A matrix of regressors (without intercept terms).
`sampling_method`	The resampling method for the data. Currently supports 'bootstrap' or 'cv' (cross-validation).
`standardise_method`	The standardisation method for the data. Currently supports 'euclidean' (default, centered by mean and divide by Euclidiean length) and 'studentise' (centred by mean and divide by standard deviation)
`times`	Number of resampling runs we perform. Default is set to 1000.
`k`	Number of partitions in averaging the MC-index. Default is set to 10.

Value

A list of outputs:

- t_square:The t^2 statistics for the regression between the VIFs and the tau's.
- MC:The MC-indices
- col_names:Column names (export for plotting purposes)

Author(s)

Chen Lin, Kevin Wang, Samuel Mueller

Examples

```
set.seed(1)
p = 10
n = 100
X = matrix(rnorm(n*p), ncol = p)
X[,1] = X[,2] + rnorm(n, 0, 0.1)
mcvis_result = mcvis(X = X)
mcvis_result
```

---

shiny_mcvis *Shiny app for mcvis exploration*

Description

Shiny app for mcvis exploration

Usage

```
shiny_mcvis(mcvis_result, X)
```

Arguments

mcvis_result	Output of the mcvis function
X	The original X matrix

Value

A shiny app allowing for interactive exploration of mcvis results

Author(s)

Chen Lin, Kevin Wang, Samuel Mueller

Examples

```
if(interactive()){
set.seed(1)
p = 10
n = 100
X = matrix(rnorm(n*p), ncol = p)
mcvis_result = mcvis(X)
shiny_mcvis(mcvis_result = mcvis_result, X = X)
}
```

Index