

Package: rwa (via r-universe)

September 14, 2024

Type Package

Title Perform a Relative Weights Analysis

Version 0.0.3

Author Martin Chan <martinchan53@gmail.com>

Maintainer Martin Chan <martinchan53@gmail.com>

Description Perform a Relative Weights Analysis (RWA) (a.k.a. Key Drivers Analysis) as per the method described in Tonidandel & LeBreton (2015) <[DOI:10.1007/s10869-014-9351-z](https://doi.org/10.1007/s10869-014-9351-z)>, with its original roots in Johnson (2000) <[DOI:10.1207/S15327906MBR3501_1](https://doi.org/10.1207/S15327906MBR3501_1)>. In essence, RWA decomposes the total variance predicted in a regression model into weights that accurately reflect the proportional contribution of the predictor variables, which addresses the issue of multi-collinearity. In typical scenarios, RWA returns similar results to Shapley regression, but with a significant advantage on computational performance.

License GPL-3

Encoding UTF-8

LazyData true

URL <https://github.com/martinctc/rwa>

BugReports <https://github.com/martinctc/rwa/issues>

RoxygenNote 7.1.1

Roxygen list(markdown = TRUE)

Imports dplyr, magrittr, stats, tidyr, ggplot2

Repository <https://martinctc.r-universe.dev>

RemoteUrl <https://github.com/martinctc/rwa>

RemoteRef HEAD

RemoteSha 7980b82aef19e276d3f38ccb45f38b5bcd8dd0d3

Contents

plot_rwa	2
remove_all_na_cols	2
rwa	3

Index	5
--------------	----------

plot_rwa	<i>Plot the rescaled importance values from the output of rwa()</i>
----------	---

Description

Pass the output of `rwa()` and plot a bar chart of the rescaled importance values. Signs are always calculated and taken into account, which is equivalent to setting the `applysigns` argument to `TRUE` in `rwa()`.

Usage

```
plot_rwa(rwa)
```

Arguments

`rwa` Direct list output from `rwa()`.

Examples

```
library(ggplot2)
diamonds %>%
  rwa(outcome = "price",
      predictors = c("depth", "carat", "x", "y", "z"),
      applysigns = TRUE) %>%
  plot_rwa()
```

remove_all_na_cols	<i>Remove any columns where all the values are missing</i>
--------------------	--

Description

Pass a data frame and returns a version where all columns made up of entirely missing values are removed.

Usage

```
remove_all_na_cols(df)
```

Arguments

df Data frame to be passed through.

Details

This is used within `rwa()`.

rwa *Create a Relative Weights Analysis (RWA)*

Description

This function creates a Relative Weights Analysis (RWA) and returns a list of outputs. RWA provides a heuristic method for estimating the relative weight of predictor variables in multiple regression, which involves creating a multiple regression with on a set of transformed predictors which are orthogonal to each other but maximally related to the original set of predictors. `rwa()` is optimised for dplyr pipes and shows positive / negative signs for weights.

Usage

```
rwa(df, outcome, predictors, applysigns = FALSE, plot = TRUE)
```

Arguments

df Data frame or tibble to be passed through.

outcome Outcome variable, to be specified as a string or bare input. Must be a numeric variable.

predictors Predictor variable(s), to be specified as a vector of string(s) or bare input(s). All variables must be numeric.

applysigns Logical value specifying whether to show an estimate that applies the sign. Defaults to FALSE.

plot Logical value specifying whether to plot the rescaled importance metrics.

Details

`rwa()` produces raw relative weight values (epsilons) as well as rescaled weights (scaled as a percentage of predictable variance) for every predictor in the model. Signs are added to the weights when the `applysigns` argument is set to TRUE. See <https://relativeimportance.davidson.edu/multipleregression.html> for the original implementation that inspired this package.

Value

rwa() returns a list of outputs, as follows:

- predictors: character vector of names of the predictor variables used.
- rsquare: the rsquare value of the regression model.
- result: the final output of the importance metrics.
 - The Rescaled.RelWeight column sums up to 100.
 - The Sign column indicates whether a predictor is positively or negatively correlated with the outcome.
- n: indicates the number of observations used in the analysis.
- lambda:
- RXX: Correlation matrix of all the predictor variables against each other.
- RXY: Correlation values of the predictor variables against the outcome variable.

Examples

```
library(ggplot2)
rwa(diamonds, "price", c("depth", "carat"))
```

Index

plot_rwa, [2](#)

remove_all_na_cols, [2](#)

rwa, [3](#)