

Package ‘rwa’

May 9, 2026

Type Package

Title Perform a Relative Weights Analysis

Version 0.1.1

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 multicollinearity. In typical scenarios, RWA returns similar results to Shapley regression, but with a significant advantage on computational performance.

License GPL-3

Encoding UTF-8

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

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

RoxygenNote 7.3.2

Imports dplyr, magrittr, stats, tidyr, ggplot2, boot, purrr, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0), rlang, spelling

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Martin Chan [aut, cre]

Maintainer Martin Chan <martinchan53@gmail.com>

Repository CRAN

Date/Publication 2026-01-20 17:10:07 UTC

Contents

plot_rwa	2
remove_all_na_cols	2
rwa	3

Index	6
--------------	----------

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)
# Use a smaller sample for faster execution
diamonds_small <- diamonds[sample(nrow(diamonds), 1000), ]
diamonds_small %>%
  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,
  sort = TRUE,
  bootstrap = FALSE,
  n_bootstrap = 1000,
  conf_level = 0.95,
  focal = NULL,
  comprehensive = FALSE,
  include_rescaled_ci = FALSE
)
```

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.

<code>appliesigns</code>	Logical value specifying whether to show an estimate that applies the sign. Defaults to FALSE.
<code>sort</code>	Logical value specifying whether to sort results by rescaled relative weights in descending order. Defaults to TRUE.
<code>bootstrap</code>	Logical value specifying whether to calculate bootstrap confidence intervals. Defaults to FALSE.
<code>n_bootstrap</code>	Number of bootstrap samples to use when <code>bootstrap = TRUE</code> . Defaults to 1000.
<code>conf_level</code>	Confidence level for bootstrap intervals. Defaults to 0.95.
<code>focal</code>	Focal variable for bootstrap comparisons (optional).
<code>comprehensive</code>	Whether to run comprehensive bootstrap analysis including random variable and focal comparisons.
<code>include_rescaled_ci</code>	Logical value specifying whether to include confidence intervals for rescaled weights. Defaults to FALSE due to compositional data constraints. Use with caution.

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 `appliesigns` argument is set to TRUE. See <https://www.scotttonidandel.com/rwa-web> 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 (sorted by `Rescaled.RelWeight` in descending order by default).
 - The `Rescaled.RelWeight` column sums up to 100.
 - The `Sign` column indicates whether a predictor is positively or negatively correlated with the outcome.
 - When `bootstrap = TRUE`, includes confidence interval columns for raw weights.
 - Rescaled weight CIs are available via `include_rescaled_ci = TRUE` but not recommended for inference.
- `n`: indicates the number of observations used in the analysis.
- `bootstrap`: bootstrap results (only present when `bootstrap = TRUE`), containing:
 - `ci_results`: confidence intervals for weights
 - `boot_object`: raw bootstrap object for advanced analysis
 - `n_bootstrap`: number of bootstrap samples used
- `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)
# Basic RWA (results sorted by default)
rwa(diamonds,"price",c("depth","carat"))

# RWA without sorting (preserves original predictor order)
rwa(diamonds,"price",c("depth","carat"), sort = FALSE)

# For faster examples, use a subset of data for bootstrap
diamonds_small <- diamonds[sample(nrow(diamonds), 1000), ]

# RWA with bootstrap confidence intervals (raw weights only)
rwa(diamonds_small,"price",c("depth","carat"), bootstrap = TRUE, n_bootstrap = 100)

# Include rescaled weight CIs (use with caution for inference)
rwa(diamonds_small,"price",c("depth","carat"), bootstrap = TRUE,
    include_rescaled_ci = TRUE, n_bootstrap = 100)

# Comprehensive bootstrap analysis with focal variable
result <- rwa(diamonds_small,"price",c("depth","carat","table"),
    bootstrap = TRUE, comprehensive = TRUE, focal = "carat",
    n_bootstrap = 100)
# View confidence intervals
result$bootstrap$ci_results
```

Index

`plot_rwa`, [2](#)

`remove_all_na_cols`, [2](#)

`rwa`, [3](#)