

# Package ‘regweight’

February 16, 2022

**Title** Convenience Functions for Implicit Regression Weights

**Version** 1.0.2

**Description** A simple wrapper for calculating regression weights as defined by Aronow and Samii (2015) <doi:10.1111/ajps.12185>. Given a model object and a term of interest, 'regweight' will calculate implicit regression weights and provide a variety of useful visualizations and summary statistics.

**URL** <https://github.com/ddimmery/regweight>  
<https://ddimmery.github.io/regweight/>

**BugReports** <https://github.com/ddimmery/regweight/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Suggests** covr, CBPS, devtools, estimatr, knitr, rmarkdown, sf,  
testthat (>= 3.0.0), USAboundaries, tibble, htmltools

**Imports** checkmate, dplyr, ggplot2, glue, gt, lpdensity, rlang, scales,  
tidyr, tidyselect

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Drew Dimmery [aut, cre] (<<https://orcid.org/0000-0001-9602-6325>>)

**Maintainer** Drew Dimmery <drew.dimmery@univie.ac.at>

**Repository** CRAN

**Date/Publication** 2022-02-16 13:50:02 UTC

## R topics documented:

calculate_weights . . . . .	2
hist.regweight . . . . .	3
plot.regweight . . . . .	4

plot_weighting_continuous . . . . .	4
plot_weighting_discrete . . . . .	5
plot_weighting_map . . . . .	6
summary.regweight . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

calculate_weights	<i>Calculate regression weights</i>
-------------------	-------------------------------------

---

## Description

Given a model and a term of interest, calculate the Aronow and Samii (2015) doi: [10.1111/ajps.12185](https://doi.org/10.1111/ajps.12185) regression weights and return an object which can be used to diagnose these implicit weights.

## Usage

```
calculate_weights(mod, term)
```

## Arguments

mod	The linear model object from <code>lm</code> or <code>lm_robust</code> .
term	String indicating the term for which to calculate the implicit regression weights. This must uniquely match a coefficient name (i.e. it must be a string which appears in only one element of <code>coef(mod)</code> ).

## Details

This calculates the implicit regression weights for a particular term in a given regression model. In short, this calculates the weights for a coefficient  $\beta$  such that:

$$\frac{E[w_i \beta_i]}{E[w_i]} \rightarrow \beta$$

where  $\beta_i$  is the unit level effect. The expectation of  $w_i$  is the conditional variance of the variable of interest.

For details and examples, view the vignette: `vignette("example-usage", package = "regweight")`

## Value

An object of class `regweight` containing:

term	The term in the regression for which weights were calculated.
model	The partial regression model object.
weights	The implicit regression weights.

## References

Aronow, P.M. and Samii, C. (2016), "Does Regression Produce Representative Estimates of Causal Effects?". *American Journal of Political Science*, 60: 250-267. doi: [10.1111/ajps.12185](https://doi.org/10.1111/ajps.12185)

## Examples

```
y <- rnorm(100)
a <- rbinom(100, 1, 0.5)
x <- rnorm(100)
m1 <- stats::lm(y ~ a + x)

w1 <- calculate_weights(m1, "a")
```

---

hist.regweight	<i>Plot histogram of implicit regression weights</i>
----------------	--

---

## Description

This provides a simple histogram of the Aronow and Samii (2015) doi: [10.1111/ajps.12185](https://doi.org/10.1111/ajps.12185) implicit regression weights.

## Usage

```
## S3 method for class 'regweight'
hist(x, bw = NULL, ...)
```

## Arguments

x	Weighting model object
bw	Bandwidth for histogram bins. If not provided, the Freedman-Diaconis rule will be used.
...	unused arguments

## Value

A `ggplot2::ggplot` object.

---

plot.regweight      *Plot diagnostics for implicit regression weights*

---

### Description

This provides access to all plotting functions and tries to smartly use the appropriate one based on the covariate provided. If covariate type is inappropriately recognized please use the appropriate plotting function directly.

### Usage

```
## S3 method for class 'regweight'
plot(x, covariate, ...)
```

### Arguments

x	Weighting model object
covariate	Covariate vector
...	additional arguments passed to individual plotting functions

### Value

A `ggplot2::ggplot` object.

### See Also

[calculate\\_weights\(\)](#), [plot\\_weighting\\_discrete\(\)](#), [plot\\_weighting\\_continuous\(\)](#), [plot\\_weighting\\_map\(\)](#)

---

plot\_weighting\_continuous  
*Plot weights across a continuous covariate*

---

### Description

This provides a simple plot for the distribution of a single continuous covariate in the nominal sample and the implicit sample defined by the Aronow and Samii (2015) doi: [10.1111/ajps.12185](https://doi.org/10.1111/ajps.12185) regression weights.

### Usage

```
plot_weighting_continuous(mod, covariate, alpha = 0.05, num_eval = 250, ...)
```

### Arguments

mod	Weighting model object
covariate	Covariate vector
alpha	Number between zero and one indicating the desired alpha level for confidence intervals.
num_eval	Number of points at which to evaluate the density.
...	unused arguments

### Details

Kernel density estimates use the bias-corrected methods of Cattaneo et al (2020).

### Value

A `ggplot2::ggplot` object.

### References

- Cattaneo, Jansson and Ma (2021): `lpdensity`: Local Polynomial Density Estimation and Inference. *Journal of Statistical Software*, forthcoming.
- Cattaneo, Jansson and Ma (2020): Simple Local Polynomial Density Estimators. *Journal of the American Statistical Association* 115(531): 1449-1455.

### See Also

[lpdensity::lpdensity\(\)](#)

### Examples

```
y <- rnorm(100)
a <- rbinom(100, 1, 0.5)
x <- rnorm(100)
cov <- runif(100)
mod <- stats::lm(y ~ a + x)
rw_mod <- calculate_weights(mod, "a")
plot_weighting_continuous(rw_mod, cov, num_eval = 25)
```

---

plot\_weighting\_discrete

*Plot weights subdivided by a discrete covariate*

---

### Description

This provides a simple plot for the distribution of a single discrete covariate in the nominal sample and the implicit sample defined by the Aronow and Samii (2015) doi: [10.1111/ajps.12185](https://doi.org/10.1111/ajps.12185) regression weights.

**Usage**

```
plot_weighting_discrete(mod, covariate, alpha = 0.05, ...)
```

**Arguments**

mod	Weighting model object
covariate	Covariate vector
alpha	Number between zero and one indicating the desired alpha level for confidence intervals.
...	unused arguments

**Value**

A `ggplot2::ggplot` object.

**Examples**

```
y <- rnorm(100)
a <- rbinom(100, 1, 0.5)
x <- rnorm(100)
g <- sample(1:4, 100, replace = TRUE)
mod <- stats::lm(y ~ a + x)
rw_mod <- calculate_weights(mod, "a")
plot_weighting_discrete(rw_mod, g)
```

---

plot\_weighting\_map      *Plot weights in a choropleth map*

---

**Description**

This provides a choropleth map showing the distribution over geometries under the implicit sample defined by the Aronow and Samii (2015) doi: [10.1111/ajps.12185](https://doi.org/10.1111/ajps.12185) regression weights.

**Usage**

```
plot_weighting_map(mod, geometry, ...)
```

**Arguments**

mod	Weighting model object
geometry	A column of class <code>sf::sfc</code> with the geometry associated with each observation.
...	unused arguments

**Value**

A `ggplot2::ggplot` object.

**See Also**

[hist.regweight\(\)](#), [plot.regweight\(\)](#), [sf::sf\(\)](#)

---

summary.regweight      *Create summary statistics for implicit sample*

---

**Description**

This provides a simple table of summary statistics for the implicit sample defined by Aronow and Samii (2015) doi: [10.1111/ajps.12185](https://doi.org/10.1111/ajps.12185).

**Usage**

```
## S3 method for class 'regweight'  
summary(object, df, output = "tibble", ...)
```

**Arguments**

object	Weighting model object
df	dataframe with one column for each covariate to include in the resulting table of summary statistics.
output	Desired output type. Default is to return a tibble, but can also select from "latex" and "html" to return a formatted table for inclusion in a paper or report.
...	unused

**Value**

One of three outputs depending on the requested type:

- tibble: Returns a tibble object (see [tibble::tibble\(\)](#)).
- latex: Returns a knit\_asis object (see [knitr::asis\\_output\(\)](#)).
- html: Returns an html object (see [htmltools::HTML\(\)](#)).

# Index

`calculate_weights`, 2  
`calculate_weights()`, 4

`hist.regweight`, 3  
`hist.regweight()`, 7  
`htmltools::HTML()`, 7

`knitr::asis_output()`, 7

`lpdensity::lpdensity()`, 5

`plot.regweight`, 4  
`plot.regweight()`, 7  
`plot_weighting_continuous`, 4  
`plot_weighting_continuous()`, 4  
`plot_weighting_discrete`, 5  
`plot_weighting_discrete()`, 4  
`plot_weighting_map`, 6  
`plot_weighting_map()`, 4

`sf::sfc()`, 7  
`summary.regweight`, 7

`tibble::tibble()`, 7