Package 'ggrounded'

April 10, 2026

Title Rounded Bar Plots

Version 0.1.0

Author Botan Ağın [aut, cre, cph]

Maintainer Botan Ağın <aginbotan@gmail.com>

Description Creates bar plots with rounded corners using 'ggplot2'.
The code in this package was adapted from a solution provided by
Stack Overflow user 'sthoch' in the following post
<https://stackoverflow.com/questions/62176038/
r-ggplot2-bar-chart-with-round-corners-on-top-of-bar>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports ggplot2, grid

URL https://github.com/botan/ggrounded,
https://botan.github.io/ggrounded/

BugReports https://github.com/botan/ggrounded/issues

Suggests covr, testthat (>= 3.0.0), vdiffr

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2026-04-10 12:20:02 UTC

Contents

geom_bar_rounded 2
geom_histogram_rounded 4

Index **6**

geom_bar_rounded *Bar plot with rounded corners*

Description

Create a bar plot with rounded corners using 'ggplot2'. `geom_col_rounded()` and `geom_bar_rounded()` are extensions of the `ggplot2::geom_col()` and `ggplot2::geom_bar()` functions and they provide additional aesthetics for more visually appealing outputs.

Usage

```
geom_bar_rounded(
  mapping = NULL,
  data = NULL,
  position = ggplot2::position_stack(reverse = TRUE),
  radius = 0.2,
  ...,
  width = NULL,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_col_rounded(
  mapping = NULL,
  data = NULL,
  position = ggplot2::position_stack(reverse = TRUE),
  radius = 0.2,
  ...,
  width = NULL,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `ggplot2::aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot2::ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `ggplot2::fortify()` for which variables will be created.

	A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
position	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use position_jitter), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
radius	A normalized rounding amount between 0 and 1. Use 0 for square corners and 1 for the maximum rounding each bar can safely support.
...	Other arguments passed on to ggplot2::layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.
width	Bar width. If not supplied, the default is to use the width of 90% of the resolution of the data.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. ggplot2::borders().

Value

A ggplot object.

Note

The code in this function was adapted from a solution provided by Stack Overflow user 'sthoch' in the following post: https://stackoverflow.com/questions/62176038/r-ggplot2-bar-chart-with-round-corners

Examples

```
library(ggplot2)

ggplot(data.frame(x = letters[1:3], y = c(2.3, 1.9, 3.2)), aes(x, y)) +
  geom_col_rounded()

ggplot(data.frame(x = letters[1:3], y = c(2.3, 1.9, 3.2)), aes(x, y)) +
  geom_col_rounded(radius = 1)

ggplot(mpg, aes(class)) +
  geom_bar_rounded()
```

geom_histogram_rounded *Histogram with rounded bars*

Description

Create a histogram with rounded corners using 'ggplot2'. `geom_histogram_rounded()` is an extension of the `ggplot2::geom_histogram()` function and uses the same binning parameters as `ggplot2::stat_bin()` while drawing bars with rounded corners.

Usage

```
geom_histogram_rounded(
  mapping = NULL,
  data = NULL,
  position = "stack",
  radius = 0.2,
  ...,
  binwidth = NULL,
  bins = NULL,
  center = NULL,
  boundary = NULL,
  closed = c("right", "left"),
  pad = FALSE,
  breaks = NULL,
  drop = "none",
  orientation = NA,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `ggplot2::aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot2::ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `ggplot2::fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).

position	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use position_jitter), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
radius	A normalized rounding amount between 0 and 1. Use 0 for square corners and 1 for the maximum rounding each bar can safely support.
...	Other arguments passed on to ggplot2::layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.
binwidth	The width of the bins. The default uses 30 bins that cover the range of the data. You should usually override this value, exploring multiple widths to find the best to illustrate the stories in your data.
bins	Number of bins. Overridden by binwidth. Defaults to 30.
center, boundary	Bin position specifiers passed to ggplot2::stat_bin().
closed	One of "right" or "left" indicating whether right or left edges of bins are included.
pad	If TRUE, adds empty bins at either side of the range.
breaks	A numeric vector of bin boundaries.
drop	Controls whether empty bins are dropped. Passed to ggplot2::stat_bin().
orientation	The orientation of the layer. Passed to ggplot2::stat_bin().
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. ggplot2::borders().

Value

A ggplot object.

Examples

```
library(ggplot2)

ggplot(faithful, aes(waiting)) +
  geom_histogram_rounded()

ggplot(faithful, aes(waiting)) +
  geom_histogram_rounded(bins = 10, radius = 1)
```

Index

geom_bar_rounded, 2
geom_col_rounded *(*geom_bar_rounded*)*, 2
geom_histogram_rounded, 4
ggplot2::aes(), *2*, *4*
ggplot2::borders(), *3*, *5*
ggplot2::fortify(), *2*, *4*
ggplot2::geom_bar(), *2*
ggplot2::geom_col(), *2*
ggplot2::geom_histogram(), *4*
ggplot2::ggplot(), *2*, *4*
ggplot2::layer(), *3*, *5*
ggplot2::stat_bin(), *4*, *5*