

# Package ‘ggrounded’

May 11, 2023

**Title** Rounded Bar Plots

**Version** 0.0.3

**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](https://stackoverflow.com/questions/62176038/r-ggplot2-bar-chart-with-round-corners-on-top-of-bar)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** ggplot2, grid, gridGeometry

**URL** <https://github.com/botan/ggrounded>,  
<https://botan.github.io/ggrounded/>

**BugReports** <https://github.com/botan/ggrounded/issues>

**Suggests** covr, testthat (>= 3.0.0), vdiff

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Botan Ađın [aut, cre, cph]

**Maintainer** Botan Ađın <aginbotan@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-05-11 07:10:09 UTC

## R topics documented:

geom\_bar\_rounded . . . . . 2

**Index** . . . . . 4

---

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 = grid::unit(4, "pt"),
  ...,
  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 = grid::unit(4, "pt"),
  ...,
  width = NULL,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

## Arguments

mapping	Set of aesthetic mappings created by <code>ggplot2::aes()</code> . If specified and <code>inherit.aes = TRUE</code> (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 <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot2::ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>ggplot2::fortify()</code> 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)`).

<code>position</code>	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use <code>position_jitter</code> ), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
<code>radius</code>	The radius of the rounded corners, given as a unit object.
<code>...</code>	Other arguments passed on to <code>ggplot2::layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
<code>width</code>	Bar width. If not supplied, the default is to use the width of 90% of the resolution of the data.
<code>na.rm</code>	If <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
<code>show.legend</code>	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
<code>inherit.aes</code>	If <code>FALSE</code> , 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. <code>ggplot2::borders()</code> .

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

# Index

`geom_bar_rounded`, 2  
`geom_col_rounded` (`geom_bar_rounded`), 2  
`ggplot2::aes()`, 2  
`ggplot2::borders()`, 3  
`ggplot2::fortify()`, 2  
`ggplot2::geom_bar()`, 2  
`ggplot2::geom_col()`, 2  
`ggplot2::ggplot()`, 2  
`ggplot2::layer()`, 3