

# Package ‘ggwidth’

May 8, 2026

**Title** Publication-Quality 'ggplot2' Geom Width

**Version** 0.1.1

**Description** Width helper functions for publication-quality 'ggplot2' visualisation. These functions make it easier to create geoms such as bars that have a consistent width appearance across plots.

**License** MIT + file LICENSE

**URL** <https://github.com/davidhodge931/ggwidth>,  
<https://davidhodge931.github.io/ggwidth/>

**BugReports** <https://github.com/davidhodge931/ggwidth/issues>

**Depends** R (>= 4.1.0)

**Imports** ggplot2 (>= 4.0.0), grid, rlang

**Suggests** dplyr, forcats, patchwork, testthat (>= 3.0.0), tibble, withr

**Encoding** UTF-8

**Language** en-GB

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** David Hodge [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0002-3868-7501>>)

**Maintainer** David Hodge <davidhodge931@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-05-04 03:10:02 UTC

## Contents

get_width . . . . .	2
set_equiwidth . . . . .	4
<b>Index</b>	<b>6</b>

---

get\_width                      *Standardise 'ggplot2' geom width*

---

### Description

Standardise the width in 'ggplot2' geoms to appear visually consistent across plots with different numbers of categories, panel dimensions, and orientations.

This can be used in geoms such as `ggplot2::geom_bar()/ggplot2::geom_col()`, `ggplot2::geom_boxplot()`, `ggplot2::geom_errorbar()`.

### Usage

```
get_width(
  ...,
  n = NULL,
  n_dodge = NULL,
  orientation = c("x", "y"),
  equiwidth = NULL,
  panel_widths = NULL,
  panel_heights = NULL
)
```

### Arguments

...	Must be empty. Forces all other arguments to be named and allows trailing commas.
n	Number of categories in the orientation aesthetic (i.e. "x" or "y"). For faceted plots, use the maximum n within a facet.
n_dodge	Number of dodge categories. Must match the number of groups in the fill or colour aesthetic when using <code>position_dodge()</code> .
orientation	Orientation: "x" for vertical (width appearance equiwidth to panel width), "y" for horizontal (width appearance equiwidth to panel height).
equiwidth	Numeric. Scaling factor that controls the width appearance. A value of 1 is the default. Increase to make a wider appearance, and decrease to make a thinner appearance. If NULL, uses the value set by <code>set_equiwidth()</code> , falling back to 1.
panel_widths	A <code>grid::unit</code> object specifying the panel width. If NULL (default), uses the value set in the current theme.
panel_heights	A <code>grid::unit</code> object specifying the panel height. If NULL (default), uses the value set in the current theme.

### Value

A numeric width value passed to the width argument of `geom_bar()`, `geom_col()`, or similar geoms.

**See Also**[set\\_equiwidth\(\)](#)**Examples**

```
library(ggplot2)
library(dplyr)
library(patchwork)

set_theme(
  theme_grey() +
  theme(panel.widths = rep(unit(75, "mm"), 2)) +
  theme(panel.heights = rep(unit(50, "mm"), 2))
)
set_equiwidth(1)

p1 <- mpg |>
  ggplot(aes(x = drv)) +
  geom_bar(
    width = get_width(n = 3),
    colour = "black",
    fill = "grey",
  )

p2 <- diamonds |>
  ggplot(aes(x = color)) +
  geom_bar(
    width = get_width(n = 7),
    colour = "black",
    fill = "grey",
  )

p3 <- diamonds |>
  ggplot(aes(y = color)) +
  geom_bar(
    width = get_width(n = 7, orientation = "y"),
    colour = "black",
    fill = "grey",
  )

p4 <- mpg |>
  ggplot(aes(x = drv, group = factor(cyl))) +
  geom_bar(
    position = position_dodge(preserve = "single"),
    width = get_width(n = 3, n_dodge = 4),
    colour = "black",
    fill = "grey",
  )

p1 + p2 + p3 + p4

d <- tibble::tibble(
```

```

continent = c("Europe", "Europe", "Europe", "Europe", "Europe",
              "South America", "South America"),
country    = c("AT", "DE", "DK", "ES", "PK", "TW", "BR"),
value      = c(10L, 15L, 20L, 25L, 17L, 13L, 5L)
)

max_n <- d |>
  count(continent) |>
  pull(n) |>
  max()

d |>
  mutate(country = forcats::fct_rev(country)) |>
  ggplot(aes(y = country, x = value)) +
  geom_col(
    width = get_width(n = max_n, orientation = "y"),
    colour = "black",
    fill = "grey",
  ) +
  facet_wrap(~continent, scales = "free_y") +
  scale_y_discrete(continuous.limits = c(1, max_n)) +
  coord_cartesian(reverse = "y", clip = "off")

mpg |>
  ggplot(aes(x = drv)) +
  geom_bar(
    width = get_width(n = 3, panel_widths = unit(160, "mm")),
    colour = "black",
    fill = "grey",
  ) +
  theme(panel_widths = unit(160, "mm"))

```

---

set\_equiwidth

*Set a global equiwidth*


---

## Description

Sets a global default for the equiwidth argument in `get_width()`. This is the scaling factor that controls the width appearance. All subsequent calls to `get_width()` use this value when `equiwidth = NULL`, and fall back to 1.

## Usage

```
set_equiwidth(equiwidth = 1)
```

## Arguments

`equiwidth` Numeric. Multiplicative factor that controls the width appearance. A value of 1 is the default. Increase to make a wider appearance, and decrease to make a thinner appearance.

**Value**

No return value. Side effects only.

**See Also**

[get\\_width\(\)](#)

**Examples**

```
set_equiwidth(1)
set_equiwidth(0.75)
set_equiwidth(1.33)
```

# Index

`get_width`, 2  
`get_width()`, 5  
`ggplot2::geom_bar()`, 2  
`ggplot2::geom_boxplot()`, 2  
`ggplot2::geom_col()`, 2  
`ggplot2::geom_errorbar()`, 2  
  
`set_equiwidth`, 4  
`set_equiwidth()`, 3