Package: ggerror (via r-universe)

October 13, 2026

Title Extended Error Geoms for 'ggplot2'

Version 1.0.0

Description Extends the 'ggplot2' error geoms. geom_error() accepts an error aesthetic with auto-inference of the orientation. It also supports `error_neg` and `error_pos` for asymmetric cases, with full control over aesthetics per side, such as color, width etc...

License MIT + file LICENSE

Encoding UTF-8

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Imports cli, ggplot2 (>= 3.4.0), lifecycle, rlang

Suggests knitr, purrr, rmarkdown, svglite, testthat (>= 3.0.0), vdiffr (>= 1.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://iamyannc.github.io/ggerror/

BugReports https://github.com/iamYannC/ggerror/issues

Repository https://iamyannc.r-universe.dev

Date/Publication 2026-05-16 05:52:10 UTC

RemoteUrl https://github.com/iamyannc/ggerror

RemoteRef HEAD

RemoteSha 98c90cd50a66e2118dab1d529aa7f4325341eb23

Contents

geom_error 2
stat_error 4

Index **6**

geom_error *Error bars with automatic orientation*

Description

A thin wrapper around ggplot2::geom_errorbar(), ggplot2::geom_linerange(), ggplot2::geom_crossbar(), and ggplot2::geom_pointrange() that accepts a single error aesthetic and figures out orientation from the data. For asymmetric errors, use error_neg + error_pos instead of error.

Usage

```
geom_error(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  error_geom = "errorbar",
  orientation = NA,
  sign_aware = FALSE,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_error_linerange(..., error_geom)

geom_error_crossbar(..., error_geom)

geom_error_pointrange(..., error_geom)
```

Arguments

mapping	Set of aesthetic mappings created by ggplot2::aes().
data	The data to be displayed in this layer.
stat	The statistical transformation to use on the data. Defaults to "identity".
position	Position adjustment.
...	Other arguments passed on to ggplot2::layer().
error_geom	One of "errorbar" (default), "linerange", "crossbar", or "pointrange". Chooses which ggplot2 error geom geom_error() dispatches to under the hood.
orientation	Either NA (the default; inferred from the data), "x" (vertical error), or "y" (horizontal error).
sign_aware	If TRUE, signed values in error are routed per row: positive values extend the bar in the positive direction, negative values extend it in the negative direction, and the opposite side is suppressed. Useful for residual plots where x/y is the

	fitted value and the bar extends toward the observed value. Incompatible with `stat = "error"`. Default `FALSE`.
`na.rm`	If `FALSE`, missing values are removed with a warning.
`show.legend`	Logical. Should this layer be included in the legends?
`inherit.aes`	If `FALSE`, overrides the default aesthetics.

Package options

Session-level knobs for the `0 -> NA` migration. Set via `options()`:

- `ggerror.silent_zero_warning` — `TRUE` suppresses the soft deprecation fired when `error_neg` or `error_pos` is set to `0` (You are encouraged to set it to `NA`). Default `FALSE`.
- `ggerror.zero_threshold` — Numeric absolute tolerance for zero-value detection. Values with a magnitude below this threshold are treated as exactly zero, triggering the warning. Defaults to `1e-8`.

Aesthetics

`geom_error()` requires `x`, `y`, and one of:

- `error` — symmetric half-width applied along the non-categorical axis.
- `error_neg` **and** `error_pos` — asymmetric; the bar extends `error_neg` in the negative direction and `error_pos` in the positive direction along the non-categorical axis. For a one-sided bar, set the unused side to `NA` — the cap, stem, and shared-bound cap on that side are all suppressed.

Mixing `error` with `error_neg` / `error_pos` is an error, as is providing only one of the asymmetric pair.

Fixed per-side styling can be supplied through `...` with `_neg` and `_pos` suffixes for `colour`, `fill`, `linewidth`, `linetype`, `alpha`, and `width`. These are fixed scalar parameters, not mapped aesthetics.

Examples

```
library(ggplot2)

ggplot(mtcars, aes(mpg, rownames(mtcars))) +
  geom_point() +
  geom_error(aes(error = drat))

ggplot(mtcars, aes(factor(cyl), mpg)) +
  geom_point() +
  geom_error(aes(error = drat), error_geom = "pointrange")

# Asymmetric: bar extends drat/2 below and drat above each point
ggplot(mtcars, aes(mpg, rownames(mtcars))) +
  geom_point() +
  geom_error(aes(error_neg = drat / 2, error_pos = drat))

# One-sided: set the unused side to NA (cap + stem auto-suppressed)
ggplot(mtcars, aes(mpg, rownames(mtcars))) +
```

```
  geom_point() +
  geom_error(aes(error_neg = NA, error_pos = drat))

# Summarise raw data: mean +/- SE per group (see also stat_error())
ggplot(mtcars, aes(factor(cyl), mpg)) +
  geom_error(stat = "error", error_geom = "pointrange")

# Signed residual plot: bar extends from fitted toward observed
model <- lm(mpg ~ wt, data = mtcars)
ggplot(mtcars, aes(fitted(model), mpg)) +
  geom_point() +
  geom_error(aes(error = resid(model)),
             sign_aware = TRUE, orientation = "x")

# Style the negative and positive halves separately
ggplot(mtcars, aes(mpg, rownames(mtcars))) +
  geom_point() +
  geom_error(
    aes(error_neg = drat / 2, error_pos = drat),
    colour_neg = "steelblue",
    colour_pos = "firebrick"
  )
```

`stat_error` *Summarising stat for* `geom_error()`

Description

`stat_error()` computes the error bounds from raw observation-level data using ggplot2's `fun.data` contract. Where `geom_error()` expects pre- computed error columns, `stat_error()` summarises `y` (or `x`, when orientation is horizontal) within each group via the function supplied to `fun`.

Usage

```
stat_error(
  mapping = NULL,
  data = NULL,
  geom = NULL,
  position = "identity",
  ...,
  fun = "mean_se",
  fun.args = list(),
  error_geom = "errorbar",
  orientation = NA,
  na.rm = FALSE,
  conf.int = 0.95,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

`mapping`, `data`, `position`, `show.legend`, `inherit.aes`
: Standard ggplot2 layer arguments.

`geom`
: The geom to render the summary with. Defaults to GeomErrorStat, which reuses `geom_error()`'s draw path.

`...`
: Additional parameters. Names that match `fun`'s formals (or any name, when `fun` accepts `...`) are forwarded to `fun`; the remainder go to `geom_error()` as per-side styling (`colour_neg`, `width_pos`, ...) or standard aesthetics.

`fun`
: One of `"mean_se"` (default, uses `ggplot2::mean_se()`), `"mean_ci"` (mean with 95% normal-theory CI via `stats::qt()`; no Hmisc dependency), or a function taking a numeric vector and returning a single-row data.frame with columns `y`, `ymin`, `ymax`.

`fun.args`
: Named list of extra arguments to pass to `fun`. Merged with any `...` arguments whose names match `fun`'s formals; `fun.args` wins on collision.

`error_geom`
: One of `"errorbar"` (default), `"linerange"`, `"crossbar"`, `"pointrange"`.

`orientation`
: `NA` (default, inferred), `"x"`, or `"y"`.

`na.rm`
: If `TRUE`, drop `NA` values from the summarised axis before applying `fun`.

`conf.int`
: Confidence level forwarded to `fun` when the function accepts a `conf.int` argument (e.g. `fun = "mean_ci"` or a custom `fun.data` with that formal). Ignored for funs that don't declare it, so it's safe to leave at the default when using `fun = "mean_se"`.

Examples

```
library(ggplot2)

ggplot(mtcars, aes(factor(cyl), mpg)) + stat_error()

ggplot(mtcars, aes(factor(cyl), mpg)) +
  stat_error(fun = "mean_ci", error_geom = "pointrange")

# 90% CI with NA-tolerant summarising:
ggplot(mtcars, aes(factor(cyl), mpg)) +
  stat_error(fun = "mean_ci", conf.int = 0.9, na.rm = TRUE)
```

Index

geom_error, 2
geom_error(), *4*, *5*
geom_error_crossbar *(*geom_error*)*, 2
geom_error_linerange *(*geom_error*)*, 2
geom_error_pointrange *(*geom_error*)*, 2
GeomError *(*geom_error*)*, 2
GeomErrorStat, *5*
GeomErrorStat *(*stat_error*)*, 4
ggplot2::aes(), *2*
ggplot2::geom_crossbar(), *2*
ggplot2::geom_errorbar(), *2*
ggplot2::geom_linerange(), *2*
ggplot2::geom_pointrange(), *2*
ggplot2::layer(), *2*
ggplot2::mean_se(), *5*

options(), *3*

stat_error, 4
StatError *(*stat_error*)*, 4
stats::qt(), *5*