Package ‘ggborderline’

July 22, 2025

URL https://github.com/wurli/ggborderline,
https://wurli.github.io/ggborderline/

Type Package

Title Line Plots that Pop

Version 0.2.0

Author Jacob Scott

Maintainer Jacob Scott <jscott2718@gmail.com>

Description A set of geometries to make line plots a little bit nicer. Use along with 'ggplot2' to:
- Improve the clarity of line plots with many overlapping lines
- Draw more realistic worms.

License MIT + file LICENSE

Encoding UTF-8

Imports cli, ggplot2, rlang, utils, vctrs

RoxygenNote 7.2.1

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2022-10-25 13:45:14 UTC

Contents

geom_borderpath *Connect observations*

Description

This set of geoms is very similar to `ggplot2::geom_path()`, `ggplot2::geom_line()` and `ggplot2::geom_step()`, with the only difference being that they accept two additional aesthetics, `bordercolour` and `borderwidth`.
For additional documentation, please refer to the ggplot2 geoms.

Usage

```
geom_borderpath(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  arrow = NULL,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_borderline(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  arrow = NULL,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_borderstep(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  direction = "hv",
```

```
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by 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 ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See 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)).
stat	The statistical transformation to use on the data for this layer, either as a ggproto Geom subclass or as a string naming the stat stripped of the stat_ prefix (e.g. "count" rather than "stat_count")
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.
...	Other arguments passed on to 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.
lineend	Line end style (round, butt, square).
linejoin	Line join style (round, mitre, bevel).
linemitre	Line mitre limit (number greater than 1).
arrow	Arrow specification, as created by grid::arrow().
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. borders().
direction	direction of stairs: 'vh' for vertical then horizontal, 'hv' for horizontal then vertical, or 'mid' for step half-way between adjacent x-values.

Value

A ggproto layer object

Examples

```
require(ggplot2)

# geom_borderline() adds a border around lines
ggplot(economics_long, aes(date, value01, colour = variable)) +
  geom_borderline()

# You can control the linewidth and colour of the border with the
# borderwidth and bordercolour aesthetics:
ggplot(economics_long, aes(date, value01, bordercolour = variable)) +
  geom_borderline(borderwidth = .4, colour = "white")

# The background 'border' part of the geom is always solid, however this
# can be used to create some nice effects:
x <- seq(0, 4 * pi, length.out = 500)
test_data <- data.frame(
  x = rep(x, 2), y = c(sin(x), cos(x)),
  fun = rep(c("sin", "cos"), each = 500)
)
ggplot(test_data, aes(x, y, colour = fun)) +
  geom_borderline(linewidth = 1, linetype = "dashed", lineend = "round")
```

---

scale_bordercolour_continuous

Scales for borderlines

Description

These scales control the linewidth and colour of the borders in borderlines. They work in much the same way as `ggplot2::scale_colour_continuous()`, `ggplot2::scale_linewidth_discrete()`, etc.

Usage

```
scale_bordercolour_continuous(..., aesthetics = "bordercolour")

scale_bordercolour_discrete(..., aesthetics = "bordercolour")

scale_borderwidth_continuous(..., aesthetics = "borderwidth")

scale_borderwidth_discrete(..., aesthetics = "borderwidth")
```

Arguments

...	Passed to the corresponding ggplot2 scales
aesthetics	Character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with. This can be useful, for example, to apply colour settings to the bordercolour and colour aesthetics at the same time, via aesthetics = c("bordercolour", "colour").

Value

A ggproto scale object

Index