Package: ggtrace (via r-universe)

November 5, 2024

Type Package

Title Trace and Highlight Groups of Data Points

Version 0.2.0.9000

Description Provides 'ggplot2' geoms that allow groups of data points to be outlined or highlighted for emphasis. This is particularly useful when working with dense datasets that are prone to overplotting.

License MIT + file LICENSE

URL https://github.com/rnabioco/ggtrace

BugReports https://github.com/rnabioco/ggtrace/issues

Depends R (>= 4.0.0)

Imports ggplot2, grid, rlang

Suggests covr, knitr, rmarkdown, tidyr, tibble, dplyr, testthat (>= 3.0.0), vdiffr (>= 1.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

SystemRequirements pandoc

Collate 'a-legend-draw.R' 'data.R' 'geom-path-trace.R' 'geom-point-trace.R' 'ggtrace-package.R' 'grouping.R' 'utilities-ggplot2.R'

Config/Needs/website pkgdown, rnabioco/rbitemplate

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

RemoteUrl https://github.com/rnabioco/ggtrace

RemoteRef HEAD

RemoteSha 7fbf0740b4c21fa099cbd813bf3a053e8a592a8f

Contents

`clusters` *Mock clusters*

Description

Mock clusters

Usage

```
clusters
```

Format

A tibble with 14282 rows and 3 variables

`draw_key` *Key glyphs for legends*

Description

Each geom has an associated function that draws the key when the geom needs to be displayed in a legend. These functions are called `draw_key_*()`, where `*` stands for the name of the respective key glyph. The key glyphs can be customized for individual geoms by providing a geom with the `key_glyph` argument.

Usage

```
draw_key_point_trace(data, params, size)

draw_key_path_trace(data, params, size)
```

Arguments

`data`	A single row data frame containing the scaled aesthetics to display in this key
`params`	A list of additional parameters supplied to the geom.
`size`	Width and height of key in mm.

Value

A grid grob

Examples

```
p <- ggplot2::ggplot(stocks, ggplot2::aes(day, value, color = name))

# key glyphs can be specified by their name
p + ggplot2::geom_line(key_glyph = "point_trace")

# key glyphs can be specified via their drawing function
p + ggplot2::geom_line(key_glyph = ggplot2::draw_key_rect)
```

GeomPathTrace *GeomPathTrace*

Description

GeomPathTrace

Value

ggproto object

See Also

GeomPath

---

geom_path_trace *Trace lines*

Description

These geoms are similar to `ggplot2::geom_path()`, `ggplot2::geom_line()`, and `ggplot2::geom_step()`, but also include the ability to highlight line segments of interest. These geoms accept normal ggplot2 graphical parameters with some modifications. `fill` controls the color of the center line, `color` controls the outline color, and `stroke` controls outline width, similar to how filled shapes are modified for other ggplot2 geoms. Additional parameters including `size`, `alpha`, `linetype`, `linejoin`, `lineend`, and `linemitre` are also accepted.

Usage

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

geom_line_trace(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE,
  trace_position = "all",
  background_params = list(color = NA),
  ...
)

geom_step_trace(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  direction = "hv",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  trace_position = "all",
  background_params = list(color = NA),
  ...
)
```

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.
trace_position	Specifies which data points to outline, can be one of: • "all" to outline every group plotted • A predicate specifying which data points to outline. This must evaluate to TRUE or FALSE within the context of the input data. e.g. value > 100
background_params	Named list specifying aesthetic parameters to use for background data points when a predicate is passed to trace_position, e.g. list(color = "red")
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().

orientation	The orientation of the layer. The default (NA) automatically determines the orientation from the aesthetic mapping. In the rare event that this fails it can be given explicitly by setting orientation to either "x" or "y". See the *Orientation* section for more detail.
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

ggplot object

Aesthetics

geom_path_trace() understands the following aesthetics (required aesthetics are in bold):

- x
- y
- alpha
- colour
- fill
- group
- linetype
- size
- stroke

Learn more about setting these aesthetics in vignette("ggplot2-specs").

See Also

geom_path; geom_line; geom_step

Examples

```
# Modify line color for each group
ggplot2::ggplot(
  stocks,
  ggplot2::aes(day, value, fill = name)
) +
  geom_line_trace() +
  ggplot2::theme_minimal()

# Modify outline color for each group
ggplot2::ggplot(
  stocks,
  ggplot2::aes(day, value, color = name)
) +
  geom_line_trace() +
  ggplot2::theme_minimal()
```

```
# Specify outline color for each group
clrs <- c(
  CAC  = "#E69F00",
  DAX  = "#0072B2",
  FTSE = "#009E73",
  SMI  = "#56B4E9"
)

ggplot2::ggplot(
  stocks,
  ggplot2::aes(day, value, color = name)
) +
  geom_line_trace(stroke = 1) +
  ggplot2::scale_color_manual(values = clrs) +
  ggplot2::theme_minimal()

# Outline a subset of data points
ggplot2::ggplot(
  stocks,
  ggplot2::aes(day, value, color = name)
) +
  geom_line_trace(trace_position = day > 1500, stroke = 1) +
  ggplot2::theme_minimal()

# Modify appearance of background data points
ggplot2::ggplot(
  stocks,
  ggplot2::aes(day, value, color = name)
) +
  geom_line_trace(
    trace_position    = day > 1500,
    background_params = list(color = NA, fill = "grey75"),
    stroke            = 1
  ) +
  ggplot2::theme_minimal()

# Remove outline
ggplot2::ggplot(
  stocks,
  ggplot2::aes(day, value, fill = name)
) +
  geom_line_trace(
    trace_position    = day > 1500,
    background_params = list(fill = "grey75"),
    color             = NA
  ) +
  ggplot2::theme_minimal()
```

---

geom_point_trace *Trace points*

Description

This geom is similar to `ggplot2::geom_point()`, but also includes the ability to outline points of interest. `geom_point_trace()` accepts normal ggplot2 graphical parameters with some modifications. `fill` controls the color of each point, `color` controls the outline color, and `stroke` controls outline width, similar to how filled shapes are modified for other ggplot2 geoms. Additional parameters including `size`, `linetype`, and `alpha` are also accepted.

Usage

```
geom_point_trace(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  trace_position = "all",
  background_params = list(color = NA),
  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.
`trace_position`	Specifies which data points to outline, can be one of:

- "all" to outline every group plotted
- "bottom" to only outline the bottom layer of data points
- A predicate specifying which data points to outline. This must evaluate to TRUE or FALSE within the context of the input data. e.g. value > 100

background_params
: Named list specifying aesthetic parameters to use for background data points when a predicate is passed to trace_position, e.g. list(color = "red")

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().

Value

ggplot object

Aesthetics

geom_point_trace() understands the following aesthetics (required aesthetics are in bold):

- x
- y
- alpha
- colour
- fill
- group
- linetype
- shape
- size
- stroke

Learn more about setting these aesthetics in vignette("ggplot2-specs").

See Also

geom_point

Examples

```
# Modify outline color for each group
ggplot2::ggplot(
  clusters,
  ggplot2::aes(UMAP_1, UMAP_2, color = cluster)
) +
  geom_point_trace() +
  ggplot2::theme_minimal()

# Outline a subset of points
ggplot2::ggplot(
  clusters,
  ggplot2::aes(UMAP_1, UMAP_2, fill = cluster)
) +
  geom_point_trace(trace_position = signal < 0 | signal > 17) +
  ggplot2::theme_minimal()

# Modify appearance of background points
ggplot2::ggplot(
  clusters,
  ggplot2::aes(UMAP_1, UMAP_2, fill = cluster)
) +
  geom_point_trace(
    trace_position    = signal < 0 | signal > 17,
    background_params = list(color = NA, fill = "grey85")
  ) +
  ggplot2::theme_minimal()
```

---

stocks *EuStockMarkets in long format*

Description

EuStockMarkets in long format

Usage

```
stocks
```

Format

A tibble with 74440 rows and 3 variables

Index