Package 'ggpol'

September 20, 2018

Title Visualizing Social Science Data with 'ggplot2'

Version 0.0.4

Description
A 'ggplot2' extension for implementing parliament charts and several other useful visualizations.

URL https://github.com/erocoar/ggpol

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends ggplot2 (>= 2.2.1), R (>= 3.4.0)

Imports grid, gtable, grDevices, plyr, rlang, dplyr

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

Collate 'facet_share.R' 'geom_arcbar.R' 'geom_bartext.R'
'geom_boxjitter.R' 'geom_circle.R' 'geom_parliament.R'
'geom_tshighlight.R' 'ggproto-classes.R' 'stat-arcbar.R'
'stat-boxjitter.R' 'stat-circle.R' 'stat-parliament.R'
'utilities.R'

NeedsCompilation no

Author Frederik Tiedemann [aut, cre]

Maintainer Frederik Tiedemann <fj.tiedemann@googlemail.com>

Repository CRAN

Date/Publication 2018-09-20 13:00:03 UTC

R topics documented:

`FacetShare` *ggplot extensions*

Description

ggplot extensions

`facet_share` *A shared axis for two panels*

Description

'facet_share' uses [facet_wrap()] to build two panels with a shared axis.

Usage

```
facet_share(facets, scales = "fixed", reverse_num = FALSE, shrink = TRUE,
  labeller = "label_value", as.table = TRUE, switch = NULL, drop = TRUE,
  dir = "h", strip.position = "top")
```

Arguments

`facets`	A set of variables or expressions quoted by `vars()` and defining faceting groups on the rows or columns dimension. The variables can be named (the names are passed to `labeller`). For compatibility with the classic interface, can also be a formula or character vector. Use either a one sided formula, '~a • `b, or a character vector,`c("a", "b")'.
`scales`	Should scales be fixed (`"fixed"`, the default), free (`"free"`), or free in one dimension (`"free_x"`, `"free_y"`)?
`reverse_num`	Used when passing on flipped data (times -1) for the second (right/bottom) panel. If 'TRUE', this will multiply the axis labels for that panel by -1.
`shrink`	If `TRUE`, will shrink scales to fit output of statistics, not raw data. If `FALSE`, will be range of raw data before statistical summary.
`labeller`	A function that takes one data frame of labels and returns a list or data frame of character vectors. Each input column corresponds to one factor. Thus there will be more than one with formulae of the type `~cyl + am`. Each output column gets displayed as one separate line in the strip label. This function should inherit from the "labeller" S3 class for compatibility with `labeller()`. See `label_value()` for more details and pointers to other options.

as.table	If TRUE, the default, the facets are laid out like a table with highest values at the bottom-right. If FALSE, the facets are laid out like a plot with the highest value at the top-right.
switch	By default, the labels are displayed on the top and right of the plot. If "x", the top labels will be displayed to the bottom. If "y", the right-hand side labels will be displayed to the left. Can also be set to "both".
drop	If TRUE, the default, all factor levels not used in the data will automatically be dropped. If FALSE, all factor levels will be shown, regardless of whether or not they appear in the data.
dir	Direction: either "h" for horizontal, the default, or "v", for vertical.
strip.position	By default, the labels are displayed on the top of the plot. Using strip.position it is possible to place the labels on either of the four sides by setting strip.position = c("top", "bott

Examples

```
df <- data.frame(age = sample(1:20, 1000, replace = TRUE),
                 gender = c("M","F"), levels = c("M", "F"))

# Get the count per age and sex
df$count <- 1
df <- aggregate(count ~ gender + age, data = df, length)

# For the horizontally shared axis, if we want to mirror the axes,
# we have to multiply the first panel by -1, and use coord_flip().
df_h <- df
df_h$count = ifelse(df_h$gender == "F", df_h$count * -1, df_h$count)

p <- ggplot(df_h, aes(x = factor(age), y = count, fill = gender)) +
  geom_bar(stat = "identity") +
  facet_share(~gender, dir = "h", scales = "free", reverse_num = TRUE) +
  coord_flip() +
  labs(x = "Age", y = "Count") +
  theme(legend.position = "bottom")

p

# When setting direction to vertical, and if we want to mirror the second panel,
# we must multiply the second factor by -1.
# And levels(factor(gender))[2] is M.
df_v <- df
df_v$count <- ifelse(df_v$gender == "M", df_v$count * -1, df_v$count)

p <- ggplot(df_v, aes(x = as.factor(age), y = count, fill = gender)) +
  geom_bar(stat = "identity") +
  facet_share(~gender, dir = "v", reverse_num = TRUE,
              scales = "free", strip.position = "left") +
  labs(x = "Age", y = "Count") +
  theme(legend.position = "left")
p
```

geom_arcbar *Create an Arc-Barchart*

Description

An arc bar diagram that allows for spacing between the individual arc components and spans 180 degrees.

Usage

```
geom_arcbar(mapping = NULL, data = NULL, stat = "arcbar",
  position = "identity", n = 360, sep = 0.05, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE, ...)

stat_arcbar(mapping = NULL, data = NULL, geom = "arcbar",
  position = "identity", n = 360, sep = 0.05, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE, ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or 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.
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
n	The number of
sep	Separation between the different shares, as a total proportion of pi.
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().

...	Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `color = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.
geom	The geometric object to use display the data

Aesthetics

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

- **shares** - **r0** - inner radius - **r1** - outer radius - color - fill - linetype - alpha

Examples

```
bt <- data.frame(
  parties = factor(c("CDU", "CSU", "AfD", "FDP", "SPD",
                     "Linke", "Gruene", "Fraktionslos"),
                   levels = c("CDU", "CSU", "AfD", "FDP", "SPD",
                              "Linke", "Gruene", "Fraktionslos")),
  seats   = c(200, 46, 92, 80, 153, 69, 67, 2),
  colors  = c("black", "blue", "lightblue", "yellow", "red",
              "purple", "green", "grey"),
  stringsAsFactors = FALSE)

ggplot(bt) +
  geom_arcbar(aes(shares = seats, r0 = 5, r1 = 10, fill = parties)) +
  scale_fill_manual(values = bt$colors) +
  coord_fixed() +
  theme_void()
```

---

geom_bartext *Repelling text for GeomBar.*

Description

Repelling text for GeomBar.

Usage

```
geom_bartext(mapping = NULL, data = NULL, stat = "identity",
  position = "identity", parse = FALSE, nudge_x = 0, nudge_y = 0,
  spacing = 0.003, dir = "v", check_overlap = FALSE, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE, ...)
```

Arguments

mapping	Set of aesthetic mappings created by `aes()` or `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.
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
parse	If TRUE, the labels will be parsed into expressions and displayed as described in ?plotmath
nudge_x	Horizontal and vertical adjustment to nudge labels by. Useful for offsetting text from points, particularly on discrete scales.
nudge_y	Horizontal and vertical adjustment to nudge labels by. Useful for offsetting text from points, particularly on discrete scales.
spacing	Defaults to 0.003. Minimum spacing between labels in NPC units.
dir	Defaults to "v", i.e. vertical repel of overlapping groups of labels. Can alternatively be set to "h" for horizontal repel.
check_overlap	If TRUE, text that overlaps previous text in the same layer will not be plotted.
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().
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They may also be parameters to the paired geom/stat.

Examples

```
df <- data.frame(L = rep(LETTERS[1:2], each = 4),
                 l = rep(letters[1:4], 2),
                 val = c(96.5, 1, 2, 0.5, 48, 0.7, 0.3, 51))

ggplot(df, aes(x = L, y = val, fill = l)) +
  geom_bar(stat = "identity") +
  geom_bartext(aes(label = paste0(val, "%")), position = position_stack(vjust = 0.5)) +
  ggtitle("GeomBartext")
```

geom_boxjitter *A hybrid boxplot.*

Description

Half boxplot, half scatterplot with customizable jitter.

Usage

```
geom_boxjitter(mapping = NULL, data = NULL, stat = "BoxJitter",
  position = "dodge", ..., outlier.colour = NULL, outlier.color = NULL,
  outlier.fill = NULL, outlier.shape = 19, outlier.size = 1.5,
  outlier.stroke = 0.5, outlier.alpha = NULL, outlier.intersect = FALSE,
  jitter.colour = NULL, jitter.color = NULL, jitter.fill = NULL,
  jitter.shape = 19, jitter.size = 1.5, jitter.stroke = 0.5,
  jitter.alpha = NULL, jitter.width = NULL, jitter.height = NULL,
  jitter.seed = NULL, boxplot.expand = FALSE, notch = FALSE,
  notchwidth = 0.5, varwidth = FALSE, errorbar.draw = FALSE,
  errorbar.length = 0.5, na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or 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.
stat	Use to override the default connection between geom_boxplot and stat_boxplot.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They may also be parameters to the paired geom/stat.
outlier.colour	Default aesthetics for outliers. Set to NULL to inherit from the aesthetics used for the box. In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.color` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.fill` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.shape` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.size` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.stroke` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.alpha` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.intersect`

Defaults to 'FALSE'. If set to 'TRUE', outliers will be part of the jitter-plot (but keeping the given outlier graphical parameters) rather than plotted vertically above / below the whisker lines.

`jitter.colour, jitter.color, jitter.fill, jitter.shape, jitter.size, jitter.stroke, jitter.alpha`

Default aesthetics for jitter, set to 'NULL' to inherit from the aesthetics used for the box.

`jitter.width` Width passed to position_jitter. Defaults to half the width of the boxplot.

`jitter.height` Height passed to position_jitter. Defaults to 40 percent of the resolution.

`jitter.seed` Seed passed to position_jitter for reproducible jittering.

`boxplot.expand` Defaults to 'FALSE'. If set to 'TRUE', the full boxplots will be plotted.

`notch` If `FALSE` (default) make a standard box plot. If `TRUE`, make a notched box plot. Notches are used to compare groups; if the notches of two boxes do not overlap, this suggests that the medians are significantly different.

`notchwidth` For a notched box plot, width of the notch relative to the body (default 0.5)

`varwidth` If `FALSE` (default) make a standard box plot. If `TRUE`, boxes are drawn with widths proportional to the square-roots of the number of observations in the groups (possibly weighted, using the `weight` aesthetic).

`errorbar.draw` Draw horizontal whiskers at the top and bottom (the IQR). Defaults to 'FALSE'.

`errorbar.length`

Length of the horizontal whiskers (errorbar). Defaults to half the width of the half-boxplot, or half the width of the entire boxplot if 'boxplot.expand' is set to 'TRUE'.

`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()`.

Examples

```
set.seed(221)
df <- data.frame(score = rgamma(150, 4, 1),
                 gender = sample(c("M", "F"), 150, replace = TRUE),
                 genotype = factor(sample(1:3, 150, replace = TRUE)))

ggplot(df) + geom_boxjitter(aes(x = gender, y = score, fill = genotype),
                            jitter.shape = 21, jitter.color = NA,
                            jitter.height = 0, jitter.width = 0.04,
                            outlier.color = NA, errorbar.draw = TRUE) +
  scale_fill_manual(values = c("#CF3721", "#31A9B8", "#258039")) +
  theme_minimal()
```

---

geom_circle *Circles with pre-defined radii.*

Description

Similar to '[geom_point()]', but allows for full control of the size.

Usage

```
geom_circle(mapping = NULL, data = NULL, stat = "circle",
  position = "identity", n = 360, na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, ...)

stat_circle(mapping = NULL, data = NULL, geom = "circle",
  position = "identity", n = 360, na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE, ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or 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.
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.

n	The number of points calculated for the circle polygon, defaults to 360.
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().
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They may also be parameters to the paired geom/stat.
geom	The geometric object to use display the data

Aesthetics

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

- **x** - x-coordinate of center - **y** - y-coordinate of center - **r** - radius - color - fill - linetype - alpha

Examples

```
set.seed(22189)
df <- data.frame(x = sample(1:10, 3), y = sample(1:10, 3),
                 r = sample(3:4, 3, replace = TRUE))

ggplot(df) + geom_circle(aes(x = x, y = y, r = r, fill = gl(3, 1))) +
  coord_fixed()
```

---

geom_parliament *Create a Parliament Diagram*

Description

Draws a parliament diagram based on parties' member counts, where each point in the arc represents a single member of parliament. Parties are plotted right-to-left.

Usage

```
geom_parliament(mapping = NULL, data = NULL, stat = "parliament",
  position = "identity", r0 = 1.5, r1 = 3, n = 360, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE, ...)

stat_parliament(mapping = NULL, data = NULL, geom = "parliament",
  position = "identity", r0 = 1.5, r1 = 3, n = 360, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE, ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or 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.
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
r0	Inner radius, defaults to 1.5.
r1	Outer radius, defaults to 3.
n	Number of passed to 'StatCircle', defaults to 360.
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().
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They may also be parameters to the paired geom/stat.
geom	The geometric object to use display the data

Aesthetics

geom_parliament understands the following aesthetics (required aesthetics are in bold): - **seats** - number of seats of the parties - fill - color - linetype - alpha

Computed variables

x,y coordinates of individual MPs

Examples

```
bt <- data.frame(
  parties = factor(c("CDU", "CSU", "AfD", "FDP", "SPD",
                     "Linke", "Gruene", "Fraktionslos"),
```

```
                    levels = c("CDU", "CSU", "AfD", "FDP", "SPD",
                               "Linke", "Gruene", "Fraktionslos")),
  seats   = c(200, 46, 92, 80, 153, 69, 67, 2),
  colors  = c("black", "blue", "lightblue", "yellow",
              "red","purple", "green", "grey"),
  stringsAsFactors = FALSE)

ggplot(bt) +
  geom_parliament(aes(seats = seats, fill = parties), color = "black") +
  scale_fill_manual(values = bt$colors, labels = bt$parties) +
  coord_fixed() +
  theme_void()
```

`geom_tshighlight` *Timeseries highlighting*

Description

This is a version of [ggplot2::geom_rect()] that defaults to spanning the entirety of the y-axis.

Usage

```
geom_tshighlight(mapping = NULL, data = NULL, stat = "identity",
  position = "identity", ..., na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()` or `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.
`stat`	The statistical transformation to use on the data for this layer, as a string.
`position`	Position adjustment, either as a string, or the result of a call to a position adjustment function.
`...`	Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `color = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

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().

Aesthetics

geom_tshighlight understands the following aesthetics (required aesthetics are in bold): - **xmin** - **xmax** - fill - color

Examples

```
ggplot(economics, aes(x = date, y = unemploy)) +
  geom_line() +
  geom_tshighlight(aes(xmin = as.Date("01/01/1990", format = "%d/%m/%Y"),
                       xmax = as.Date("01/01/2000", format = "%d/%m/%Y")),
                   alpha = 0.01)
```

Index