Package 'geomtextpath'

January 24, 2022

Type Package

Title Curved Text in 'ggplot2'

Version 0.1.0

Description A 'ggplot2' extension that allows text to follow curved paths. Curved text makes it easier to directly label paths or neatly annotate in polar co-ordinates.

License MIT + file LICENSE

LazyData true

URL https://allancameron.github.io/geomtextpath/

BugReports https://github.com/AllanCameron/geomtextpath/issues

Encoding UTF-8

Depends ggplot2, R (>= 3.6.0)

Imports grid, scales, systemfonts, rlang, textshaping

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0), covr, knitr, rmarkdown, ragg, roxygen2, sf, xml2, markdown

Config/testthat/edition 3

VignetteBuilder knitr

Collate 'borrowed_gridtext.R' 'coord_curvedpolar.R' 'data.R' 'text_params.R' 'geom_textpath.R' 'geom_labelpath.R' 'geom_textsegment.R' 'geom_textabline.R' 'geom_textcontour.R' 'geom_textcurve.R' 'geom_textdensity.R' 'utils.R' 'geom_textdensity2d.R' 'geom_texthline.R' 'geom_textsf.R' 'geom_textsmooth.R' 'geom_textvline.R' 'geomtextpath-package.R' 'grob_labelpath.R' 'grob_textpath.R' 'onload.R' 'path_handling.R' 'scales.R' 'sf_helpers.R' 'smoothing.R' 'text_helpers.R' 'text_placement.R' 'trig_helpers.R'

NeedsCompilation no

Author Allan Cameron [aut, cre], Teun van den Brand [aut] (<https://orcid.org/0000-0002-9335-7468>)

Maintainer Allan Cameron <Allan.Cameron@nhs.scot>

Repository CRAN

Date/Publication 2022-01-24 19:32:49 UTC

R topics documented:

`coord_curvedpolar` *Polar coordinates with curved text on x axis*

Description

Polar co-ordinates in 'ggplot2' help to create a range of circular plots, which can be used to present data in a visually appealing, user-friendly way. However, the standard 'coord_polar' uses a 'textGrob' to render the labels on the circumferential (theta) axis, meaning that labels do not rotate or curve in line with the axis. 'coord_curvedpolar' aims to be identical to 'coord_polar', except that the text on the theta axis follows the curve of the plot, correcting automatically for resizing to preserve letter spacing and size.

Usage

```
coord_curvedpolar(
  theta = "x",
  start = 0,
  direction = 1,
  clip = "on",
  halign = c("center")
)
```

Arguments

theta	variable to map angle to (‘x‘ or ‘y‘)
start	Offset of starting point from 12 o’clock in radians. Offset is applied clockwise or anticlockwise depending on value of ‘direction‘.
direction	1, clockwise; -1, anticlockwise
clip	Should drawing be clipped to the extent of the plot panel? A setting of ‘"on"‘ (the default) means yes, and a setting of ‘"off"‘ means no. For details, please see [‘coord_cartesian()‘].
halign	A `character(1)` describing how multi-line text should be justified. Can either be `"center"` (default), `"left"` or `"right"`.

Value

A ‘Coord‘ ggproto object that can be added to a plot.

Examples

```
# A pie chart = stacked bar chart + polar coordinates
pie <- ggplot(mtcars, aes(x = factor(1), fill = factor(cyl))) +
 geom_bar(width = 1)
 pie + coord_curvedpolar(theta = "y")

# Demonstrating curved category labels
p <- ggplot(data.frame(x = paste("Category label", 1:5), y = runif(5)),
            aes(x, y, fill = x)) +
       geom_col() +
       theme_bw() +
       theme(panel.border = element_blank(),
             legend.position = "none",
             axis.text.x = element_text(size = 10, vjust = 0.5))

# Standard bar chart in Cartesian Co-ordinates
p

# Standard coord_polar axis labels
p + coord_polar()

# Curved polar co-ordinate labels
p + coord_curvedpolar()
```

`GeomTextpath` *The Geom object for a textpath*

Description

This is the `ggproto` class that creates the textpath layer. It is not intended to be used directly by the end user.

geom_textabline *Labelled reference lines: horizontal, vertical, and diagonal*

Description

These geoms add labelled reference lines to a plot, either horizontal, vertical, or diagonal (specified by slope and intercept). These are useful for annotating plots. They are the labelled equivalent of the `geom_vline`, `geom_hline` and `geom_abline` from ggplot2.

Usage

```
geom_textabline(
  mapping = NULL,
  data = NULL,
  slope,
  intercept,
  ...,
  na.rm = FALSE,
  show.legend = NA
)

geom_labelabline(
  mapping = NULL,
  data = NULL,
  slope,
  intercept,
  ...,
  straight = NULL,
  label.r = unit(0.15, "lines"),
  label.padding = unit(0.25, "lines"),
  na.rm = FALSE,
  show.legend = NA
)

geom_texthline(
  mapping = NULL,
  data = NULL,
  yintercept,
  stat = "identity",
  position = "identity",
  ...,
  arrow = NULL,
  lineend = "butt",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

```
geom_labelhline(
  mapping = NULL,
  data = NULL,
  yintercept,
  stat = "identity",
  position = "identity",
  ...,
  arrow = NULL,
  lineend = "butt",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  straight = NULL,
  label.r = unit(0.15, "lines"),
  label.padding = unit(0.25, "lines")
)

geom_textvline(
  mapping = NULL,
  data = NULL,
  xintercept,
  stat = "identity",
  position = "identity",
  ...,
  arrow = NULL,
  lineend = "butt",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_labelvline(
  mapping = NULL,
  data = NULL,
  xintercept,
  stat = "identity",
  position = "identity",
  ...,
  arrow = NULL,
  lineend = "butt",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  straight = NULL,
  label.r = unit(0.15, "lines"),
  label.padding = unit(0.25, "lines")
)
```

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. A function can be created from a formula (e.g. ~ head(.x,10)).
slope	The slope of the abline
intercept	the point on the y axis at which the abline crosses it.
...	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. These can also be the following text-path parameters: text_only A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored. gap A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA. upright A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text. halign A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right". offset A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting. parse A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used. straight A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths. padding A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path. text_smoothing a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.

rich A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().

remove_long if TRUE, labels that are longer than their associated path will be removed.

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.
straight	A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.
label.r	Radius of rounded corners. Defaults to 0.15 lines.
label.padding	Amount of padding around label. Defaults to 0.25 lines.
yintercept	The value at which the line should intercept the y axis
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.
arrow	Arrow specification, as created by grid::arrow().
lineend	Line end style (round, butt, square).
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().
xintercept	The value at which the line should intercept the y axis

Details

Although reference lines are straight, and therefore don't lend themselves to curved text, these geom layers are included in this package because they make labelling reference lines easier, allow automatic line breaking if desired, and will translate nicely into polar co-ordinates.

These geoms act slightly differently from other geoms. You can supply the parameters in two ways: either as arguments to the layer function, or via aesthetics. If you use arguments, e.g. geom_textabline(label = "my label",intercept = 0,slope = 1), then behind the scenes the geom makes a new data frame containing just the data you've supplied. That means that the lines will be the same in all facets; if you want them to vary across facets, construct the data frame yourself and use aesthetics.

Unlike most other geoms, these geoms do not inherit aesthetics from the plot default, because they do not understand x and y aesthetics which are commonly set in the plot. They also do not affect the x and y scales.

Value

A Layer ggproto object that can be added to a plot.

Aesthetics

The geom_textabline(), geom_texthline() and geom_textvline() understand the following aesthetics (required aesthetics are in bold):

- label
- slope (geom_textabline() and geom_labelabline())
- intercept (geom_textabline() and geom_labelabline())
- yintercept (geom_texthline() and geom_labelhline())
- xintercept (geom_textvline() and geom_labelvline())
- alpha
- angle
- colour
- family
- fontface
- group
- hjust
- vjust
- linecolour
- lineheight
- linetype
- linewidth
- size
- spacing
- textcolour

In addition to aforementioned aesthetics, geom_labelabline(), geom_labelhline() and geom_labelvline() also understand:

- boxcolour
- boxlinetype
- boxlinewidth
- fill

The spacing aesthetic allows fine control of spacing of text, which is called 'tracking' in typography. The default is 0 and units are measured in 1/1000 em. Numbers greater than zero increase the spacing, whereas negative numbers decrease the spacing.

See Also

Other geom layers that place text on paths.

Examples

```
ggplot(mtcars, aes(mpg, disp)) +
geom_point() +
geom_texthline(yintercept = 200, label = "displacement threshold",
               hjust = 0.8, color = "red4") +
geom_textvline(xintercept = 20, label = "consumption threshold", hjust = 0.8,
               linetype = 2, vjust = 1.3, color = "blue4") +
geom_textabline(slope = 15, intercept = -100, label = "partition line",
                color = "green4", hjust = 0.6, vjust = -0.2)
```

`geom_textcontour` *Produce labelled contour lines in* **ggplot2**

Description

Contour lines are available already in **ggplot2**, but the native `geom_contour` does not allow the lines to be labelled with the level of each contour. `geom_textcontour` adds this ability.

Usage

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

geom_labelcontour(
  mapping = NULL,
  data = NULL,
  stat = "textcontour",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...,
  lineend = "butt",
```

```
  linejoin = "round",
  linemitre = 10,
  bins = NULL,
  binwidth = NULL,
  breaks = NULL,
  label.padding = unit(0.25, "lines"),
  label.r = unit(0.15, "lines"),
  arrow = NULL
)

stat_textcontour(
  mapping = NULL,
  data = NULL,
  geom = "textcontour",
  position = "identity",
  ...,
  bins = NULL,
  binwidth = NULL,
  breaks = NULL,
  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. 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, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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().

lineend — Line end style (round, butt, square).

linejoin — Line join style (round, mitre, bevel).

linemitre — Line mitre limit (number greater than 1).

bins — Number of contour bins. Overridden by binwidth.

binwidth — The width of the contour bins. Overridden by breaks.

breaks — Numeric vector to set the contour breaks. Overrides binwidth and bins. By default, this is a vector of length ten with pretty() breaks.

... — 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. These can also be the following text-path parameters:

- text_only A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored.
- gap A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA.
- upright A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text.
- halign A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right".
- offset A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting.
- parse A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used.
- straight A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.
- padding A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.
- text_smoothing a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.
- rich A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().
- remove_long if TRUE, labels that are longer than their associated path will be removed.

label.padding	Amount of padding around label. Defaults to 0.25 lines.
label.r	Radius of rounded corners. Defaults to 0.15 lines.
arrow	Arrow specification, as created by grid::arrow().
geom	The geometric object to use display the data

Value

A Layer ggproto object that can be added to a plot.

Aesthetics

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

- x
- y
- alpha
- angle
- colour
- family
- fontface
- group
- hjust
- linecolour
- lineheight
- linetype
- linewidth
- size
- spacing
- textcolour
- vjust

In addition to aforementioned aesthetics, geom_labelcontour() also understands:

- boxcolour
- boxlinetype
- boxlinewidth
- fill

The spacing aesthetic allows fine control of spacing of text, which is called 'tracking' in typography. The default is 0 and units are measured in 1/1000 em. Numbers greater than zero increase the spacing, whereas negative numbers decrease the spacing.

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

Computed variables

The variable ‘level‘ is a numeric or a factor depending on whether lines or bands are calculated.

‘level‘ Height of contour. This is a numeric vector that represents bin boundaries.

See Also

Other geom layers that place text on paths.

Examples

```
df <- expand.grid(x = seq(nrow(volcano)), y = seq(ncol(volcano)))
df$z <- as.vector(volcano)

ggplot(df, aes(x, y, z = z)) +
  geom_contour_filled(bins = 6, alpha = 0.6) +
  geom_textcontour(bins = 6, size = 2.5, padding = unit(0.05, "in")) +
  scale_fill_manual(values = terrain.colors(11)) +
  theme_classic() +
  theme(legend.position = "none")
```

---

geom_textcurve *Text on a curve*

Description

`geom_textcurve()` and `geom_labelcurve()` draw text on curved lines. See the underlying `grid::curveGrob()` for the parameters that control the curve.

Usage

```
geom_textcurve(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  curvature = 0.5,
  angle = 90,
  ncp = 5,
  arrow = NULL,
  lineend = "butt",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_labelcurve(
```

```
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  curvature = 0.5,
  angle = 90,
  ncp = 5,
  arrow = NULL,
  lineend = "butt",
  label.r = unit(0.15, "lines"),
  label.padding = unit(0.25, "lines"),
  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. 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, 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 colour = "red" or size = 3. These can also be the following text-path parameters: text_only A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored. gap A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA.

upright A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text.

halign A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right".

offset A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting.

parse A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used.

straight A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.

padding A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.

rich A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().

remove_long if TRUE, labels that are longer than their associated path will be removed.

curvature A numeric value giving the amount of curvature. Negative values produce left-hand curves, positive values produce right-hand curves, and zero produces a straight line.

angle A numeric value between 0 and 180, giving an amount to skew the control points of the curve. Values less than 90 skew the curve towards the start point and values greater than 90 skew the curve towards the end point.

ncp The number of control points used to draw the curve. More control points creates a smoother curve.

arrow Arrow specification, as created by grid::arrow().

lineend Line end style (round, butt, square).

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().

label.r Radius of rounded corners. Defaults to 0.15 lines.

label.padding Amount of padding around label. Defaults to 0.25 lines.

Value

A Layer ggproto object that can be added to a plot.

Aesthetics

`geom_textcurve()` understands the following aesthetics (required aesthetics are in bold):

- `x`
- `xend`
- `y`
- `yend`
- `label`
- `alpha`
- `angle`
- `colour`
- `family`
- `fontface`
- `group`
- `hjust`
- `linecolour`
- `lineheight`
- `linetype`
- `linewidth`
- `size`
- `spacing`
- `textcolour`
- `vjust`

In addition to aforementioned aesthetics, `geom_labelcurve()` also understands:

- `boxcolour`
- `boxlinetype`
- `boxlinewidth`
- `fill`

The `spacing` aesthetic allows fine control of spacing of text, which is called 'tracking' in typography. The default is 0 and units are measured in 1/1000 em. Numbers greater than zero increase the spacing, whereas negative numbers decrease the spacing.

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

Examples

```
t <- seq(0, 2 * pi, length.out = 4)[-1]

df <- data.frame(
  x = cos(t),
  y = sin(t),
  xend = cos(t + 1.8),
  yend = sin(t + 1.8)
)

ggplot(df, aes(x, y, xend = xend, yend = yend)) +
  geom_textcurve(
    label = c(
      "A chicken lays an egg",
      "A chick becomes a chicken",
      "An egg hatches into a chick"
    ),
    curvature = 0.5, vjust = 2,
    arrow = arrow(ends = "first")
  ) +
  coord_equal(xlim = c(-1.1, 1.1), ylim = c(-1.1, 1.1))
```

---

geom_textdensity *Produce smoothly labelled density plots in* **ggplot2**

Description

Line plots of smoothed kernel density estimates are available in **ggplot2** via geom_density. This geom layer simply adds a text label to each curve that follow the contour of the density line when used as a drop-in replacement for geom_density

Usage

```
geom_textdensity(
  mapping = NULL,
  data = NULL,
  stat = "density",
  position = "identity",
  ...,
  bw = "nrd0",
  adjust = 1,
  kernel = "gaussian",
  n = 512,
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  na.rm = FALSE,
  show.legend = NA,
```

```
  inherit.aes = TRUE
)

geom_labeldensity(
  mapping = NULL,
  data = NULL,
  stat = "density",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...,
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  bw = "nrd0",
  adjust = 1,
  kernel = "gaussian",
  n = 512,
  label.padding = unit(0.25, "lines"),
  label.r = unit(0.15, "lines"),
  arrow = NULL
)
```

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. 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, 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 colour = "red" or size = 3. These can also be the following text-path parameters: text_only A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored.

gap A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA.

upright A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text.

halign A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right".

offset A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting.

parse A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used.

straight A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.

padding A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.

text_smoothing a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.

rich A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().

remove_long if TRUE, labels that are longer than their associated path will be removed.

bw The smoothing bandwidth to be used. If numeric, the standard deviation of the smoothing kernel. If character, a rule to choose the bandwidth, as listed in stats::bw.nrd().

adjust A multiplicate bandwidth adjustment. This makes it possible to adjust the bandwidth while still using the a bandwidth estimator. For example, adjust = 1/2 means use half of the default bandwidth.

kernel Kernel. See list of available kernels in density().

n number of equally spaced points at which the density is to be estimated, should be a power of two, see density() for details

lineend Line end style (round, butt, square).

linejoin Line join style (round, mitre, bevel).

linemitre Line mitre limit (number greater than 1).

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().
label.padding	Amount of padding around label. Defaults to 0.25 lines.
label.r	Radius of rounded corners. Defaults to 0.15 lines.
arrow	Arrow specification, as created by grid::arrow().

Value

A Layer ggproto object that can be added to a plot.

Aesthetics

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

- x
- label
- alpha
- angle
- colour
- family
- fontface
- group
- hjust
- linecolour
- lineheight
- linetype
- linewidth
- size
- spacing
- textcolour
- vjust

In addition to aforementioned aesthetics, geom_labeldensity() also understands:

- boxcolour
- boxlinetype
- boxlinewidth
- fill

The spacing aesthetic allows fine control of spacing of text, which is called 'tracking' in typography. The default is 0 and units are measured in 1/1000 em. Numbers greater than zero increase the spacing, whereas negative numbers decrease the spacing.

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

See Also

Other geom layers that place text on paths.

Examples

```
ggplot(iris, aes(Sepal.Length, label = Species, color = Species)) +
  geom_textdensity()
```

`geom_textdensity2d` *Produce labelled contour lines of 2D density in* **ggplot2**

Description

Contour lines representing 2D density are available already in **ggplot2**, but the native `geom_density_2d` does not allow the lines to be labelled with the level of each contour. `geom_textdensity2d` adds this ability.

Usage

```
geom_textdensity2d(
  mapping = NULL,
  data = NULL,
  stat = "density_2d",
  position = "identity",
  ...,
  contour_var = "density",
  n = 100,
  h = NULL,
  adjust = c(1, 1),
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_labeldensity2d(
  mapping = NULL,
  data = NULL,
  stat = "density_2d",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...,
  contour_var = "density",
```

```
  n = 100,
  h = NULL,
  adjust = c(1, 1),
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  label.padding = unit(0.25, "lines"),
  label.r = unit(0.15, "lines"),
  arrow = NULL
)
```

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. 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, 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 colour = "red" or size = 3. These can also be the following text-path parameters: text_only A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored. gap A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA. upright A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text. halign A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right". offset A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting.

parse A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used.

straight A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.

padding A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.

text_smoothing a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.

rich A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().

remove_long if TRUE, labels that are longer than their associated path will be removed.

contour_var Character string identifying the variable to contour by. Can be one of "density", "ndensity", or "count". See the section on computed variables for details.

n Number of grid points in each direction.

h Bandwidth (vector of length two). If NULL, estimated using MASS::bandwidth.nrd().

adjust A multiplicative bandwidth adjustment to be used if 'h' is 'NULL'. This makes it possible to adjust the bandwidth while still using the a bandwidth estimator. For example, adjust = 1/2 means use half of the default bandwidth.

lineend Line end style (round, butt, square).

linejoin Line join style (round, mitre, bevel).

linemitre Line mitre limit (number greater than 1).

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().

label.padding Amount of padding around label. Defaults to 0.25 lines.

label.r Radius of rounded corners. Defaults to 0.15 lines.

arrow Arrow specification, as created by grid::arrow().

Value

A Layer ggproto object that can be added to a plot.

Aesthetics

`geom_textdensity2d()` understands the following aesthetics (required aesthetics are in bold):

- `x`
- `y`
- `alpha`
- `angle`
- `colour`
- `family`
- `fontface`
- `group`
- `hjust`
- `linecolour`
- `lineheight`
- `linetype`
- `linewidth`
- `size`
- `spacing`
- `textcolour`
- `vjust`

In addition to aforementioned aesthetics, `geom_labeldensity2d()` also understands:

- `boxcolour`
- `boxlinetype`
- `boxlinewidth`
- `fill`

The `spacing` aesthetic allows fine control of spacing of text, which is called 'tracking' in typography. The default is 0 and units are measured in 1/1000 em. Numbers greater than zero increase the spacing, whereas negative numbers decrease the spacing.

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

See Also

Other geom layers that place text on paths.

Examples

```
set.seed(1)

df  <- data.frame(x = rnorm(100), y = rnorm(100))

ggplot(df, aes(x, y)) +
  geom_textdensity2d() +
  theme_classic()
```

geom_textpath *Add Curved Text Along Paths in* **ggplot2**

Description

The existing text-based geom layers in **ggplot2** (geom_text() and geom_label()) are ideal for the majority of plots, since typically textual annotations are short, straight and in line with the axes of the plot. However, there are some occasions when it is useful to have text follow a curved path. This may be to create or recreate a specific visual effect, or it may be to label a circular / polar plot in a more "natural" way.

Usage

```
geom_textpath(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...,
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  text_only = FALSE,
  gap = NA,
  upright = TRUE,
  halign = "center",
  offset = NULL,
  parse = FALSE,
  straight = FALSE,
  padding = unit(0.05, "inch"),
  text_smoothing = 0,
  rich = FALSE,
  arrow = NULL,
  remove_long = FALSE
)

geom_textline(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
```

```
  inherit.aes = TRUE,
  ...,
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  arrow = NULL
)

geom_labelpath(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...,
  lineend = "butt",
  linejoin = "round",
  linemitre = 10,
  text_only = FALSE,
  gap = FALSE,
  upright = TRUE,
  halign = "center",
  offset = NULL,
  parse = FALSE,
  straight = FALSE,
  padding = unit(0.05, "inch"),
  text_smoothing = 0,
  rich = FALSE,
  label.padding = unit(0.25, "lines"),
  label.r = unit(0.15, "lines"),
  arrow = NULL,
  remove_long = FALSE
)

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

```
  text_only = FALSE,
  gap = FALSE,
  upright = TRUE,
  halign = "center",
  offset = NULL,
  parse = FALSE,
  straight = FALSE,
  padding = unit(0.05, "inch"),
  label.padding = unit(0.25, "lines"),
  label.r = unit(0.15, "lines"),
  arrow = NULL,
  remove_long = 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. 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, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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 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).

text_only	A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored.
gap	A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA.
upright	A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text.
halign	A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right".
offset	A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting.
parse	A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used.
straight	A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.
padding	A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.
text_smoothing	a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.
rich	A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().
arrow	Arrow specification, as created by grid::arrow().
remove_long	if TRUE, labels that are longer than their associated path will be removed.
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.
label.padding	Amount of padding around label. Defaults to 0.25 lines.
label.r	Radius of rounded corners. Defaults to 0.15 lines.

Details

Limitations:

There are limitations inherent in the plotting of text elements in **ggplot2** due to the way that the underlying **grid** graphics handles text. A text string is dealt with as a zero-width object, and therefore the rotation and spacing of the letters making up the string can only be dealt with by treating each letter separately.

It is important to realise that the letters are only rotated, and do not undergo any change in shape. Thus, for example, large text appearing on convex curves will not be deformed so that individual letters are narrower at the bottom and wider at the top. Doing so would require reinterpreting the letters as polygons.

Another issue is that we may wish to use a short curved label on a much longer path. Spacing the letters equally along the path would mean there is too much space between the letters for the label to remain legible. A single text string is therefore kept "together" according to the point size of the text in `geom_textpath()`. This then leaves the problem of where on the path the text should be placed. This can be dealt with by the aesthetic mapping `hjust`, which allows the user to place the labels at the desired position along the path, including separate positions for each label.

A final point to note is that a path is usually a group-based geom (i.e. a path typically comprises x, y points from two columns over several rows of a data frame), whereas text labels can come from single rows in a data frame. This means that if we have a data frame with an x column, a y column and a grouping variable column, there can only be a single label for the group. Typically, this will be the grouping variable itself (see the examples, particularly those using the built-in `iris` data set.)

Rich text:

The rich text option (`rich = TRUE`) relies heavily on rich-text parsers copied from the {**gridtext**} package. We thank Claus O. Wilke for developing {**gridtext**} and allowing us to re-use his code under the MIT licence. Currently, the supported HTML tags are <p>, , <b>, , <i>, , <sub>, <sup> and
.

Value

A `Layer` ggproto object that can be added to a plot.

Aesthetics

`geom_textpath()` understands the following aesthetics (required aesthetics are in bold):

- `x`
- `y`
- `label`
- `alpha`
- `angle`
- `colour`
- `family`
- `fontface`
- `group`
- `hjust`
- `linecolour`
- `lineheight`
- `linetype`
- `linewidth`

- size
- spacing
- textcolour
- vjust

In addition to aforementioned aesthetics, geom_labelpath() also understands:

- boxcolour
- boxlinetype
- boxlinewidth
- fill

The spacing aesthetic allows fine control of spacing of text, which is called 'tracking' in typography. The default is 0 and units are measured in 1/1000 em. Numbers greater than zero increase the spacing, whereas negative numbers decrease the spacing.

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

See Also

Other geom layers that place text on paths.

Examples

```
# Plot text along an arbitrary path
 t <- seq(-1, 5, length.out = 1000) * pi
 spiral <- data.frame(
   x = rev(sin(t) * 1000:1),
   y = rev(cos(t) * 1000:1),
   s = seq(1, 10, length.out = 100),
   text = paste(
     "Like a circle in a spiral, like a wheel within a wheel,",
     "never ending or beginning on an ever spinning reel"
  )
)

 ggplot(spiral, aes(x, y, label = text)) +
   geom_textpath(size = 7, vjust = 2, linewidth = 0) +
   coord_equal(xlim = c(-1500, 1500), ylim = c(-1500, 1500))

# Use geom_textline as a drop-in for geom_line

 df <- data.frame(x = rep(1:100, 3),
                  y = sin(c(seq(0, pi, len = 100),
                            seq(pi, 2*pi, len = 100),
                            rep(0, 100))),
                  label = rep(c("y is increasing",
                                "y is falling",
                                "y is flat"), each = 100))

ggplot(df, aes(x, y, label = label, color = label)) +
```

```
   geom_textline(size = 6) + theme(legend.position = "none")

# Rich text labels can contain a subset of HTML tags
label <- paste0(
  "Indometacin (",
  "C<sub>19</sub>H<sub>16</sub>",
  "<span style='color:limegreen'>Cl</span>",
  "<span style='color:blue'>N</span>",
  "<span style='color:red'>O</span><sub>4</sub>",
  ") concentration"
)

# These are interpreted when `rich = TRUE`
ggplot(Indometh, aes(time, conc)) +
  geom_point() +
  geom_labelpath(
    label = label,
    stat = "smooth", formula = y ~ x, method = "loess",
    vjust = -3, size = 8, rich = TRUE
  ) +
  scale_x_log10()
```

`geom_textsegment` *Add text to line segments*

Description

`geom_textsegment` draws a line between two points defined by (x, y) and (xend, yend) and places a text label on that line. It is the text-placement equivalent of `geom_segment()`.

Usage

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

geom_labelsegment(
  mapping = NULL,
```

```
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  arrow = NULL,
  lineend = "butt",
  linejoin = "round",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  straight = NULL,
  label.r = unit(0.15, "lines"),
  label.padding = unit(0.25, "lines")
)
```

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. 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, 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 colour = "red" or size = 3. These can also be the following text-path parameters: text_only A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored. gap A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA. upright A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text.

halign A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right".

offset A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting.

parse A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used.

padding A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.

text_smoothing a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.

rich A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().

remove_long if TRUE, labels that are longer than their associated path will be removed.

arrow	Arrow specification, as created by grid::arrow().
lineend	Line end style (round, butt, square).
linejoin	Line join style (round, mitre, bevel).
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().
straight	A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.
label.r	Radius of rounded corners. Defaults to 0.15 lines.
label.padding	Amount of padding around label. Defaults to 0.25 lines.

Value

A Layer ggproto object that can be added to a plot.

Aesthetics

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

- x
- y
- xend

- yend
- label
- alpha
- angle
- colour
- family
- fontface
- group
- hjust
- linecolour
- lineheight
- linetype
- linewidth
- size
- spacing
- textcolour
- vjust

In addition to aforementioned aesthetics, geom_labelsegment() also understands:

- boxcolour
- boxlinetype
- boxlinewidth
- fill

The spacing aesthetic allows fine control of spacing of text, which is called 'tracking' in typography. The default is 0 and units are measured in 1/1000 em. Numbers greater than zero increase the spacing, whereas negative numbers decrease the spacing.

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

See Also

Other geom layers that place text on paths.

Examples

```
# The convenience here is that the position and angle
# are in sync automatically with the data
sleep2 <- reshape(sleep, direction = "wide",
                  idvar = "ID", timevar = "group")

ggplot(sleep2, aes(x = "Drug 1", y = extra.1)) +
  geom_textsegment(
    aes(xend = "Drug 2", yend = extra.2,
        label = paste0("Patient #", ID))
```

```
  )

# As an annotation
ggplot(mapping = aes(x, y)) +
  geom_col(
    data = data.frame(x = c(1, 2), y = c(1, 10))
  ) +
  annotate(
    "textsegment",
    x = 1, xend = 2, y = 1, yend = 10,
    label = "10x increase", arrow = arrow()
  )
```

`geom_textsf` *Visualise sf objects with labels*

Description

This set of geom, stat, and coord are used to visualise simple feature (sf) objects. For simple plots, you will only need `geom_sf()` as it uses `stat_sf()` and adds `coord_sf()` for you. `geom_textsf()` is an unusual geom because it will draw different geometric objects depending on what simple features are present in the data: you can get points, lines, or polygons.

Usage

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

geom_labelsf(
  mapping = aes(),
  data = NULL,
  stat = "sf",
  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. 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, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
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. You can also set this to one of "polygon", "line", and "point" to override the default legend.
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().
...	Arguments passed on to geom_textpath, geom_labelpath text_only A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored. gap A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA. upright A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text. halign A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right". offset A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting. parse A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used.

straight A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.

padding A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.

text_smoothing a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.

rich A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().

remove_long if TRUE, labels that are longer than their associated path will be removed.

label.padding Amount of padding around label. Defaults to 0.25 lines.

label.r Radius of rounded corners. Defaults to 0.15 lines.

Value

A Layer ggproto object that can be added to a plot.

Geometry aesthetic

geom_textsf() uses a unique aesthetic: geometry, giving an column of class sfc containing simple features data. There are three ways to supply the geometry aesthetic:

- Do nothing: by default geom_textsf() assumes it is stored in the geometry column.
- Explicitly pass an sf object to the data argument. This will use the primary geometry column, no matter what it's called.
- Supply your own using aes(geometry = my_column)

Unlike other aesthetics, geometry will never be inherited from the plot.

CRS

coord_sf() ensures that all layers use a common CRS. You can either specify it using the crs param, or coord_sf() will take it from the first layer that defines a CRS.

Combining sf layers and regular geoms

Most regular geoms, such as geom_point(), geom_path(), geom_text(), geom_polygon() etc. will work fine with coord_sf(). However when using these geoms, two problems arise. First, what CRS should be used for the x and y coordinates used by these non-sf geoms? The CRS applied to non-sf geoms is set by the default_crs parameter, and it defaults to NULL, which means positions for non-sf geoms are interpreted as projected coordinates in the coordinate system set by the crs parameter. This setting allows you complete control over where exactly items are placed on the plot canvas, but it may require some understanding of how projections work and how to generate data in projected coordinates. As an alternative, you can set default_crs = sf::st_crs(4326), the World Geodetic System 1984 (WGS84). This means that x and y positions are interpreted as

longitude and latitude, respectively. You can also specify any other valid CRS as the default CRS for non-sf geoms.

The second problem that arises for non-sf geoms is how straight lines should be interpreted in projected space when `default_crs` is not set to `NULL`. The approach `coord_sf()` takes is to break straight lines into small pieces (i.e., segmentize them) and then transform the pieces into projected coordinates. For the default setting where x and y are interpreted as longitude and latitude, this approach means that horizontal lines follow the parallels and vertical lines follow the meridians. If you need a different approach to handling straight lines, then you should manually segmentize and project coordinates and generate the plot in projected coordinates.

See Also

`stat_sf_coordinates()`. Other geom layers that place text on paths.

Examples

```
ggplot(waterways) +
 geom_textsf(label = "Forth and Clyde Canal",
             hjust = 0.62, vjust = -0.3, fill = "#E4E0A3") +
 lims(x = c(-4.2, -3.9), y = c(55.9, 56))
```

---

geom_textsmooth *Labelled conditional means in* **ggplot2**

Description

Smoothed conditional means are available in **ggplot2** via `geom_smooth`. This geom layer simply adds a text label to each curve that follow the contour of this line when used as a drop-in replacement for `geom_smooth`

Usage

```
geom_textsmooth(
  mapping = NULL,
  data = NULL,
  stat = "smooth",
  position = "identity",
  ...,
  method = NULL,
  formula = NULL,
  na.rm = FALSE,
  method.args = list(),
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE
)
```

```
geom_labelsmooth(
  mapping = NULL,
  data = NULL,
  stat = "smooth",
  position = "identity",
  method = NULL,
  formula = NULL,
  na.rm = FALSE,
  method.args = list(),
  orientation = NA,
  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. A function can be created from a formula (e.g. ~ head(.x,10)).
stat	Use to override the default connection between geom_smooth() and stat_smooth().
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 colour = "red" or size = 3. These can also be the following text-path parameters: text_only A logical(1) indicating whether the path part should be plotted along with the text (FALSE, the default). If TRUE, any parameters or aesthetics relating to the drawing of the path will be ignored. gap A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively, if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA. upright A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text.

halign A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right".

offset A unit object of length 1 to determine the offset of the text from the path. If this is NULL (default), the vjust parameter decides the offset. If not NULL, the offset argument overrules the vjust setting.

parse A logical(1) which if TRUE, will coerce the labels into expressions, allowing for plotmath syntax to be used.

straight A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.

padding A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.

text_smoothing a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.

rich A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().

remove_long if TRUE, labels that are longer than their associated path will be removed.

method Smoothing method (function) to use, accepts either NULL or a character vector, e.g. "lm", "glm", "gam", "loess" or a function, e.g. MASS::rlm or mgcv::gam, stats::lm, or stats::loess. "auto" is also accepted for backwards compatibility. It is equivalent to NULL.

For method = NULL the smoothing method is chosen based on the size of the largest group (across all panels). stats::loess() is used for less than 1,000 observations; otherwise mgcv::gam() is used with formula = y ~ s(x,bs = "cs") with method = "REML". Somewhat anecdotally, loess gives a better appearance, but is $O(N^2)$ in memory, so does not work for larger datasets.

If you have fewer than 1,000 observations but want to use the same gam() model that method = NULL would use, then set method = "gam", formula = y ~ s(x, bs = "cs").

formula Formula to use in smoothing function, eg. y ~ x, y ~ poly(x,2), y ~ log(x). NULL by default, in which case method = NULL implies formula = y ~ x when there are fewer than 1,000 observations and formula = y ~ s(x,bs = "cs") otherwise.

na.rm If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

method.args List of additional arguments passed on to the modelling function defined by method.

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.

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

A `Layer` ggproto object that can be added to a plot.

Aesthetics

`geom_textdensity()` understands the following aesthetics (required aesthetics are in bold):

- `x`
- `label`
- `alpha`
- `angle`
- `colour`
- `family`
- `fontface`
- `group`
- `hjust`
- `linecolour`
- `lineheight`
- `linetype`
- `linewidth`
- `size`
- `spacing`
- `textcolour`
- `vjust`

In addition to aforementioned aesthetics, `geom_labeldensity()` also understands:

- `boxcolour`
- `boxlinetype`
- `boxlinewidth`
- `fill`

The `spacing` aesthetic allows fine control of spacing of text, which is called 'tracking' in typography. The default is 0 and units are measured in 1/1000 em. Numbers greater than zero increase the spacing, whereas negative numbers decrease the spacing.

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

See Also

Other geom layers that place text on paths.

Examples

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_point(alpha = 0.1) +
  geom_textsmooth(aes(label = Species, colour = Species),
                method = "loess", formula = y ~ x,
                size = 7, linetype = 3, fontface = 2, linewidth = 1) +
  scale_colour_manual(values = c("forestgreen", "deepskyblue4", "tomato4")) +
  theme_bw() +
  theme(legend.position = "none")
```

`scale_hjust_discrete` *justification scales*

Description

Sometimes text labels on adjacent lines can clash if the lines are not well separated vertically. One option for controlling this is to use an hjust or vjust scale that will place each label on a different position on each path, either vertically (vjust) or horizontally (hjust).

Usage

```
scale_hjust_discrete(..., range = c(0, 1), guide = "none")

scale_hjust_manual(
  ...,
  values,
  breaks = waiver(),
  guide = "none",
  na.value = NA
)

scale_hjust_identity(..., guide = "none")

scale_vjust_discrete(..., guide = "none", range = c(-0.5, 1.5))

scale_vjust_manual(
  ...,
  values,
  breaks = waiver(),
  guide = "none",
  na.value = NA
)

scale_vjust_identity(..., guide = "none")
```

Arguments

...	Other arguments passed on to [continuous_scale()], [binned_scale], or [discrete_scale()] as appropriate, to control name, limits, breaks, labels and so forth.
range	Output range of hjust and vjust. Must lie between 0 and 1 for hjust.
guide	A function used to create a guide or its name. See [guides()] for more information.
values	a set of aesthetic values to map data values to. The values will be matched in order (usually alphabetical) with the limits of the scale, or with breaks if provided. If this is a named vector, then the values will be matched based on the names instead. Data values that don't match will be given na.value.
breaks	One of: - 'NULL' for no breaks - 'waiver()' for the default breaks computed by the [transformation object][scales::trans_new()] - A numeric vector of positions - A function that takes the limits as input and returns breaks as output (e.g., a function returned by [scales::extended_breaks()]). Also accepts rlang [lambda][rlang::as_function()] function notation.
na.value	Missing values will be replaced with this value.

Details

The simplest way to separate labels is by adding 'scale_hjust_discrete()' or 'scale_vjust_discrete()' to your plot, but you can get more control with 'scale_hjust_manual' and 'scale_vjust_manual'.

Value

A 'Scale' ggproto object that can be added to a plot.

Examples

```
ggplot(iris, aes(Sepal.Length, color = Species)) +
  geom_textdensity(aes(label = Species, hjust = Species), size = 6) +
  scale_hjust_discrete()
```

---

sibling_layers *Sibling layers*

Description

The goal of **geomtextpath** is to label (curved) lines in a plot. The **ggplot2** package has various ways to construct lines. For several of the **ggplot2** line functions, there is a plain 'text' sibling and a 'label' sibling that includes a text box. Below is an overview of how function in **geomtextpath** relate to those in **ggplot2**.

Details

ggplot2 geom	Text equivalent	Label equivalent
geom_path()	geom_textpath()	geom_labelpath()
geom_line()	geom_textline()	geom_labelline()
geom_segment()	geom_textsegment()	geom_labelsegment()
geom_curve()	geom_textcurve	geom_labelcurve()
geom_abline()	geom_textabline()	geom_labelabline()
geom_hline()	geom_texthline()	geom_labelhline()
geom_vline()	geom_textvline()	geom_labelvline()
geom_density()	geom_textdensity()	geom_labeldensity()
geom_smooth()	geom_textsmooth()	geom_labelsmooth()
geom_contour()	geom_textcontour()	geom_labelcontour()
geom_density2d()	geom_textdensity2d()	geom_labeldensity2d()
geom_sf()	geom_textsf()	geom_labelsf()

`textpathGrob` *Draw text on a path.*

Description

This function creates (curved) text on a path.

Usage

```
textpathGrob(
  label,
  x = 0.5,
  y = 0.5,
  id = 1L,
  just = "centre",
  hjust = NULL,
  vjust = NULL,
  halign = "left",
  angle = 0,
  straight = FALSE,
  rich = FALSE,
  gp_text = gpar(),
  gp_path = gpar(),
  gp_box = gpar(),
  gap = NA,
  upright = TRUE,
  text_smoothing = 0,
```

```
  polar_params = NULL,
  padding = unit(0.05, "inch"),
  label.padding = unit(0.25, "lines"),
  label.r = unit(0.15, "lines"),
  remove_long = FALSE,
  arrow = NULL,
  default.units = "npc",
  name = NULL,
  vp = NULL,
  as_label = FALSE
)
```

Arguments

label	A character vector.
x	A numeric vector.
y	A numeric vector.
id	A numeric vector used to separate locations in x and y into multiple lines. All locations with the same id belong to the same line.
just	The justification of the text relative to its (x, y) location. If there are two values, the first value specifies horizontal justification and the second value specifies vertical justification. Possible string values are: "left", "right", "centre", "center", "bottom", and "top". For numeric values, 0 means left (bottom) alignment and 1 means right (top) alignment.
hjust	A numeric vector specifying horizontal justification. If specified, overrides the just setting.
vjust	A numeric vector specifying justification orthogonal to the direction of the text. Alternatively a unit() object to directly set the offset from the path.
halign	A character(1) describing how multi-line text should be justified. Can either be "center" (default), "left" or "right".
angle	a numeric vector either length 1 or the same length as id describing the angle in degrees at which text should be rotated.
straight	A logical(1) which if TRUE, keeps the letters of a label on a straight baseline and if FALSE (default), lets individual letters follow the curve. This might be helpful for noisy paths.
rich	A logical(1) whether to interpret the text as html/markdown formatted rich text. Default: FALSE. See also the rich text section of the details in geom_textpath().
gp_text, gp_path	An object of class "gpar", typically the output from a call from the gpar() function. These are basically lists of graphical parameters for the text and path respectively.
gp_box	(Optional) an object of class "gpar", typically the output from a call to the gpar() function. If this is an empty list, no text box will be drawn.
gap	A logical(1) which if TRUE, breaks the path into two sections with a gap on either side of the label. If FALSE, the path is plotted as a whole. Alternatively,

	if NA, the path will be broken if the string has a vjust between 0 and 1, and not otherwise. The default for the label variant is FALSE and for the text variant is NA.
upright	A logical(1) which if TRUE (default), inverts any text where the majority of letters would upside down along the path, to improve legibility. If FALSE, the path decides the orientation of text.
text_smoothing	a numeric(1) value between 0 and 100 that smooths the text without affecting the line portion of the geom. The default value of 0 means no smoothing is applied.
polar_params	a list consisting of an x, y, and r component that specifies the central point and radius of a circle around which single-point labels will be wrapped.
padding	A unit object of length 1 to determine the padding between the text and the path when the gap parameter trims the path.
label.padding	Amount of padding around label. Defaults to 0.25 lines.
label.r	Radius of rounded corners. Defaults to 0.15 lines.
remove_long	if TRUE, labels that are longer than their associated path will be removed.
arrow	Arrow specification, as created by arrow().
default.units	A string indicating the default units to use if x or y are only given as numeric vectors.
name	A character identifier.
vp	A Grid viewport object (or NULL).
as_label	a logical TRUE or FALSE indicating whether the text should be drawn inside a text box. If FALSE, the parameters label.padding, label.r and gp_box will be ignored.

Value

An object of class gTree, containing grobs.

Examples

```
require(grid)

t <- seq(0, 2 * pi, length.out = 100)
grob <- textpathGrob(
  label = c(
    "Why I am making trigonometry jokes? Cos I can!",
    "I was never any good at sine language."
  ),
  x = c(t, t) / (2 * pi),
  y = c(cos(t), sin(t)) * 0.25 + 0.5,
  id = rep(1:2, each = length(t)),
  vjust = rep(0.5, 2 * length(t)),
  gp_text = gpar(lineheight = c(1.2, 1.2), fontsize = c(10, 10)),
  gp_path = gpar(lty = c(1, 2))
)

grid.newpage(); grid.draw(grob)
```

`waterways` *A simple features data frame of three Scottish waterways*

Description

An 'sf' object showing the River Clyde, the River Forth, and the canal joining the two.

Usage

```
waterways
```

Format

A data frame with 4 rows and 3 variables

name name of geographic object

type type of geographic object

geometry sfc objects

Details

Index