Package ‘ggfittext’

September 5, 2023

Title Fit Text Inside a Box in 'ggplot2'
Version 0.10.1
Description A 'ggplot2' extension to fit text into a box by growing, shrinking or wrapping the text.
Depends R (>= 3.6)
License GPL-2
LazyData true
Imports grid (>= 3.1), stringi (>= 1.1.2), shades (>= 1.3.1), gridtext
       (>= 0.1.4), ggplot2 (>= 2.2.1)
RoxygenNote 7.2.3
URL https://wilkox.org/ggfittext/
BugReports https://github.com/wilkox/ggfittext/issues/
Suggests knitr, rmarkdown, testthat (>= 3.0.0), vdiffr, spelling
VignetteBuilder knitr
Encoding UTF-8
Language en-GB
Config/testthat/edition 3
NeedsCompilation no
Author David Wilkins [aut, cre],
    Brady Johnston [ctb]
Maintainer David Wilkins <david@wilkox.org>
Repository CRAN
Date/Publication 2023-09-05 11:50:14 UTC

R topics documented:

     altitudes .......................................................... 2
     animals ............................................................ 2
     animals_rich ....................................................... 3
     beverages .......................................................... 3
altitudes

Some craft and their altitudes

Description
Some craft and their altitudes

Usage
altitudes

Format
A data frame with 4 rows and 2 variables:
craft the name of the craft
altitude the craft’s normal cruising altitude in metres

animals

Some animals and their attributes

Description
Some animals and their attributes

Usage
animals

Format
A data frame with 6 rows and 4 variables:
aminal the name of the animal
type the type of animal
flies whether the animal can fly
mass the average mass of the animal in grams
animals_rich

Some animals and their attributes, in rich text

Description

Some animals and their attributes, in rich text

Usage

animals_rich

Format

A data frame with 6 rows and 4 variables:

- animal  the name of the animal, in rich text
- type    the type of animal
- flies    whether the animal can fly
- mass     the average mass of the animal in grams

beverages

The compositions of some espresso-based beverages

Description

The compositions of some espresso-based beverages

Usage

beverages

Format

A data frame with 6 rows and 3 variables:

- beverage   the name of the beverage
- ingredient the ingredient
- proportion the proportion of a cup to fill with the ingredient
### Description

The compositions of some espresso-based beverages, in rich text

### Usage

```r
beverages_rich
```

### Format

A data frame with 6 rows and 3 variables:

- **beverage**: the name of the beverage
- **ingredient**: the ingredient, in rich text
- **proportion**: the proportion of a cup to fill with the ingredient

---

### Description

`geom_fit_text()` shrinks, grows and wraps text to fit inside a defined box. `geom_bar_text()` is a convenience wrapper around `geom_fit_text()` for labelling bar plots generated with `geom_col()` and `geom_bar()`.

### Usage

```r
gem_bar_text(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  padding.x = grid::unit(1, "mm"),
  padding.y = grid::unit(1, "mm"),
  min.size = 8,
  place = NULL,
  outside = NULL,
  grow = FALSE,
  reflow = FALSE,
)```
Arguments

mapping | ggplot2::aes() object as standard in `ggplot2`. Note that aesthetics specifying the box must be provided. See Details.

data, stat, position, na.rm, show.legend, inherit.aes, ...  
Standard geom arguments as for ggplot2::geom_text().

padding.x, padding.y  
Horizontal and vertical padding around the text, expressed in grid::unit() objects. Both default to 1 mm.

min.size  
Minimum font size, in points. Text that would need to be shrunk below this size
to fit the box will be hidden. Defaults to 4 pt (8 pt for `geom_bar_text()`) for `geom_bar_text()` will be set heuristically if not specified.

**place**

Where inside the box to place the text. Default is 'centre'; other options are 'topleft', 'top', 'topright', 'right', 'bottomright', 'bottom', 'bottomleft', 'left', and 'center'/'middle' which are both synonyms for 'centre'. For `geom_bar_text()`, will be set heuristically if not specified.

**outside**

If TRUE, text placed in one of 'top', 'right', 'bottom' or 'left' that would need to be shrunk smaller than min.size to fit the box will be drawn outside the box if possible. This is mostly useful for drawing text inside bar/column geoms. Defaults to TRUE for position = "identity" when using `geom_bar_text()`, otherwise FALSE.

**grow**

If TRUE, text will be grown as well as shrunk to fill the box. Defaults to FALSE.

**reflow**

If TRUE, text will be reflowed (wrapped) to better fit the box. Defaults to FALSE.

**hjust, vjust**

Horizontal and vertical justification of the text. By default, these are automatically set to appropriate values based on place.

**fullheight**

If TRUE, descenders will be counted when resizing and placing text; if FALSE, only the x-height and ascenders will be counted. The main use for this option is for aligning text at the baseline (FALSE) or preventing descenders from spilling outside the box (TRUE). By default this is set automatically depending on place and grow.

**width, height**

When using x and/or y aesthetics, these set the width and/or height of the box. These should be either `grid::unit()` objects or numeric values on the x and y scales.

**formatter**

A function that will be applied to the text before it is drawn. This is useful when using `geom_fit_text()` in context involving interpolated variables, such as with the 'gganimate' package. formatter will be applied serially to each element in the label column, so it does not need to be a vectorised function.

**contrast**

If TRUE and in combination with a fill aesthetic, the colour of the text will be inverted for better contrast against dark background fills. FALSE by default for `geom_fit_text()`, set heuristically for `geom_bar_text()`.

**rich**

If TRUE, text will be formatted with markdown and HTML markup as implemented by `gridtext::richtext_grob()`. FALSE by default. Rich text cannot be drawn in polar coordinates. Please note that rich text support is experimental and breaking changes are likely.

**flip**

If TRUE, when in polar coordinates 'upside-down' text will be flipped the 'right way up', to enhance readability.

**Details**

Except where noted, `geom_fit_text()` behaves more or less like `ggplot2::geom_text()`.

There are three ways to define the box in which you want the text to be drawn. The extents of the box on the x and y axes are independent, so any combination of these methods can be used:

1. If the x and/or y aesthetics are used to set the location of the box, the width or height will be set automatically based on the number of discrete values in x and/or y.
2. Alternatively, if x and/or y aesthetics are used, the width and/or height of the box can be overridden with a 'width' and/or 'height' argument. These should be grid::unit() objects; if not, they will be assumed to use the native axis scale.

3. The boundaries of the box can be set using the aesthetics 'xmin' and 'xmax', and/or 'ymin' and 'ymax'.

If the text is too big for the box, it will be shrunk to fit the box. With \texttt{grow = TRUE}, the text will be made to fill the box completely whether that requires shrinking or growing. \texttt{reflow = TRUE} will cause the text to be reflowed (wrapped) to better fit in the box. If the text cannot be made to fit by reflowing alone, it will be reflowed then shrunk to fit the box. Existing line breaks in the text will be respected when reflowing.

\texttt{geom_fit_text()} includes experimental support for drawing text in polar coordinates (by adding \texttt{coord_polar()} to the plot), however not all features are available when doing so.

\textbf{Aesthetics}

- \texttt{label} (required)
- \texttt{xmin AND xmax} OR \texttt{x} (required)
- \texttt{ymin AND ymax} OR \texttt{y} (required)
- \texttt{alpha}
- \texttt{angle}
- \texttt{colour}
- \texttt{family}
- \texttt{fontface}
- \texttt{lineheight}
- \texttt{size}

\textbf{Examples}

\begin{verbatim}
ggplot2::ggplot(ggplot2::presidential, ggplot2::aes(ymin = start, ymax = end, label = name, x = party)) + geom_fit_text(grow = TRUE)
\end{verbatim}

---

\textbf{gold} \hspace{2cm} \textit{Robert Frost’s poem Nothing Gold Can Stay (1923)}

\textbf{Description}

Robert Frost’s poem \textit{Nothing Gold Can Stay} (1923)

\textbf{Usage}

gold
Format

A data frame with 8 rows and 5 variables:

- **line**  a line from the poem
- **xmin**  the xmin coordinate with which to draw the line
- **xmax**  the xmax coordinate with which to draw the line
- **ymin**  the ymin coordinate with which to draw the line
- **ymax**  the ymax coordinate with which to draw the line
- **linenumber**  the line number
Index

* datasets
  altitudes, 2
  animals, 2
  animals_rich, 3
  beverages, 3
  beverages_rich, 4
  gold, 7

altitudes, 2
animals, 2
animals_rich, 3
beverages, 3
beverages_rich, 4
geom_bar_text, 4
geom_fit_text (geom_bar_text), 4
gold, 7