Package 'ggDoubleHeat'

July 22, 2025

Type Package

Title A Heatmap-Like Visualization Tool

Version 0.1.2

Description A data visualization design that provides comparison between two (Double) data sources (usually on a par with each other) on one reformed heatmap, while inheriting 'ggplot2' features.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports rlang, grid, ggplot2 (>= 3.0.0), ggnewscale (>= 0.4.5)

RoxygenNote 7.2.3

Suggests rmarkdown, knitr, dplyr, scales, tidyr

URL https://pursuitofdatascience.github.io/ggDoubleHeat/

BugReports https://github.com/PursuitOfDataScience/ggDoubleHeat/issues

NeedsCompilation no

Author Youzhi Yu [aut, cre],
Trent Buskirk [aut, ths]

Maintainer Youzhi Yu <yuyouzhi666@icloud.com>

Repository CRAN

Date/Publication 2023-08-24 21:00:04 UTC

Contents

`geom_heat_circle` *Heatcircle*

Description

The heatcircle geom is used to create the two concentric circles that use luminance to show the values from two sources on the same plot.

Usage

```
geom_heat_circle(
  outside,
  outside_name = NULL,
  outside_colors = c("#FED7D8", "#FE8C91", "#F5636B", "#E72D3F", "#C20824"),
  inside,
  inside_name = NULL,
  inside_colors = c("gray100", "gray85", "gray50", "gray35", "gray0"),
  r = 2 * sqrt(2),
  ...
)
```

Arguments

`outside`	The column name for the outside portion of heatcircle.
`outside_name`	The label name (in quotes) for the legend of the outside rendering. Default is `NULL`.
`outside_colors`	A color vector, usually as hex codes.
`inside`	The column name for the inside portion of heatcircle.
`inside_name`	The label name (in quotes) for the legend of the inside rendering. Default is `NULL`.
`inside_colors`	A color vector, usually as hex codes.
`r`	The value that controls how large of the inside portion with respect to the outside one. When `r` is larger, the inside get smaller. Default value is 2*sqrt(2) for achieving the equal inner and outer areas.
`...`	`...` accepts any arguments `scale_fill_gradientn()` has .

Value

A heatcircle comparing two data sources.

Examples

```
# heatcircle with categorical variables only

library(ggplot2)

data <- data.frame(x = rep(c("a", "b", "c"), 3),
                   y = rep(c("d", "e", "f"), 3),
                   outside_values = rep(c(1,5,7),3),
                   inside_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
geom_heat_circle(outside = outside_values,
                 inside = inside_values)


# Making the inside smaller by setting r to be larger.

ggplot(data, aes(x,y)) +
geom_heat_circle(outside = outside_values,
                 inside = inside_values,
                 r = 5)


# heatcircle with numeric variables only

data <- data.frame(x = rep(c(1, 2, 3), 3),
                   y = rep(c(1, 2, 3), 3),
                   outside_values = rep(c(1,5,7),3),
                   inside_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
geom_heat_circle(outside = outside_values,
                 inside = inside_values)


# heatcircle with a mixture of numeric and categorical variables

data <- data.frame(x = rep(c("a", "b", "c"), 3),
                   y = rep(c(1, 2, 3), 3),
                   outside_values = rep(c(1,5,7),3),
                   inside_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
geom_heat_circle(outside = outside_values,
                 inside = inside_values)
```

`geom_heat_grid` *Heatgrid*

Description

The heatgrid geom is used to create a modified heat map that uses luminance to show the values from two sources on the same plot.

Usage

```
geom_heat_grid(
  outside,
  outside_name = NULL,
  outside_colors = c("#FED7D8", "#FE8C91", "#F5636B", "#E72D3F", "#C20824"),
  inside,
  inside_name = NULL,
  inside_colors = c("gray100", "gray85", "gray50", "gray35", "gray0"),
  r = 2 * sqrt(2),
  ...
)
```

Arguments

`outside`	The column name for the outside portion of heatgrid.
`outside_name`	The label name (in quotes) for the legend of the outside rendering. Default is `NULL`.
`outside_colors`	A color vector, usually as hex codes.
`inside`	The column name for the inside portion of heatgrid.
`inside_name`	The label name (in quotes) for the legend of the inside rendering. Default is `NULL`.
`inside_colors`	A color vector, usually as hex codes.
`r`	The value that controls how large of the inside portion with respect to the outside one. When `r` is larger, the inside get smaller. Default value is 2*sqrt(2) for achieving the equal inner and outer areas.
`...`	`...` accepts any arguments `scale_fill_gradientn()` has .

Value

A heatgrid comparing two data sources.

Examples

```
# heatgrid with categorical variables only

library(ggplot2)

data <- data.frame(x = rep(c("a", "b", "c"), 3),
                   y = rep(c("d", "e", "f"), 3),
                   outside_values = rep(c(1,5,7),3),
                   inside_values = rep(c(2,3,4),3))
```

```
ggplot(data, aes(x,y)) +
geom_heat_grid(outside = outside_values,
               inside = inside_values)

# Making the inside smaller by setting r to be larger.

ggplot(data, aes(x,y)) +
geom_heat_grid(outside = outside_values,
               inside = inside_values,
               r = 5)

# heatgrid with numeric variables only

data <- data.frame(x = rep(c(1, 2, 3), 3),
                   y = rep(c(1, 2, 3), 3),
                   outside_values = rep(c(1,5,7),3),
                   inside_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
geom_heat_grid(outside = outside_values,
               inside = inside_values)


# heatgrid with a mixture of numeric and categorical variables

data <- data.frame(x = rep(c("a", "b", "c"), 3),
                   y = rep(c(1, 2, 3), 3),
                   outside_values = rep(c(1,5,7),3),
                   inside_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
geom_heat_grid(outside = outside_values,
               inside = inside_values)
```

`geom_heat_tri` *Heattriangle*

Description

The heattriangle geom is used to create the two triangles split by a diagonal line of a rectangle that use luminance to show the values from two sources on the same plot.

Usage

```
geom_heat_tri(
  lower,
```

```
  lower_name = NULL,
  lower_colors = c("#FED7D8", "#FE8C91", "#F5636B", "#E72D3F", "#C20824"),
  upper,
  upper_name = NULL,
  upper_colors = c("gray100", "gray85", "gray50", "gray35", "gray0"),
  ...
)
```

Arguments

lower	The column name for the lower portion of heattriangle.
lower_name	The label name (in quotes) for the legend of the lower rendering. Default is NULL.
lower_colors	A color vector, usually as hex codes.
upper	The column name for the upper portion of heattriangle.
upper_name	The label name (in quotes) for the legend of the upper rendering. Default is NULL.
upper_colors	A color vector, usually as hex codes.
...	... accepts any arguments scale_fill_gradientn() has .

Value

A heattriangle with the main diagonal split by a line within each unit.

Examples

```
# heattriangle with categorical variables only

library(ggplot2)

data <- data.frame(x = rep(c("a", "b", "c"), 3),
                   y = rep(c("d", "e", "f"), 3),
                   lower_values = rep(c(1,5,7),3),
                   upper_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
geom_heat_tri(lower = lower_values, upper = upper_values)


# heatcircle with numeric variables only

data <- data.frame(x = rep(c(1, 2, 3), 3),
                   y = rep(c(1, 2, 3), 3),
                   lower_values = rep(c(1,5,7),3),
                   upper_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
geom_heat_tri(lower = lower_values, upper = upper_values)
```

```
# heatcircle with a mixture of numeric and categorical variables

data <- data.frame(x = rep(c("a", "b", "c"), 3),
                   y = rep(c(1, 2, 3), 3),
                   lower_values = rep(c(1,5,7),3),
                   upper_values = rep(c(2,3,4),3))

ggplot(data, aes(x,y)) +
geom_heat_tri(lower = lower_values, upper = upper_values)
```

`ggDoubleHeat` *ggDoubleHeat: Data visualization for two sources*

Description

ggDoubleHeat, which is a ggplot2 extension, provides visualization for a reformed heatmap. Instead of facetting heatmaps by data sources, they can be combined together for making one single heatmap, generated by `geom_heat_*()` functions built in the package. Prior to using the package, users should load ggplot2.

ggDoubleHeat functions

All functions in the package are named as `geom_heat_*()`, making the naming convention consistent. Also, the arguments for functions are relatively similar, although with slight variations due to where a specific argument will connect to the position of the rendering plot. Users should reference the documentation and possibly run examples presented in the help file when trying to understand what each argument means visually.

`pitts_emojis` *Popular Emojis*

Description

The most popular Emoji of a given week in a given category from the Meltwater Tweet sample. They can be rendered by using `"richtext"` with `annotate()`.

Usage

```
pitts_emojis
```

Format

An object of class `character` of length 270.

pitts_tg *Pittsburgh COVID-related Google & Twitter incidence rates*

Description

A data set containing the 30-week incidence rates of COVID related categories from week 1 starting from June 1, 2020 to week 30 that ended in the last Sunday of the year in Pittsburgh Metropolitan Statistical Area (MSA). The data columns are introduced below. One quick note about the columns of the data set: `week_start` as a column is present in the data set for illustration purposes, reminding users what `week` column is. In other words, it does not participate any visualization.

Usage

```
pitts_tg
```

Format

A data frame with 270 rows and 6 columns:

msa Metropolitan statistical area (Pittsburgh only).

week week 1 to week 30.

week_start The Monday date of the week started.

category 9 Covid-related categories in total.

Twitter weekly tweets percentage (%) in the MSA falling into each category.

Google weekly Google search percentage (%) in the MSA falling into each category.

Source

Just like `states_tg`, Google is processed from Google Health API, and Twitter from Meltwater, a Twitter vendor. Both data sources are processed by the authors of the package.

remove_padding *Remove ggplot2 default padding*

Description

The default ggplot2 plots give certain amount of padding for both continuous and discrete variables. Due to this padding, it makes the plots generated from ‘geom_heat_*()‘ look like there is something missing. Depends on users’ preference, they can remove the "empty space" by using this function. The only thing users need to figure out is whether the ‘x‘ and ‘y‘ scales are continuous or discrete.

Usage

```
remove_padding(x = "c", y = "d", ...)
```

Arguments

x	x-axis scale, if it is continuous scale, input "c"; discrete, "d".
y	y-axis scale, if it is continuous scale, input "c"; discrete, "d".
...	...

Value

remove_padding

---

states_tg *States' COVID-related Google & Twitter incidence rates*

Description

A data set containing the 30-week incidence rates of COVID related categories from week 1 starting from June 1, 2020 to week 30 that ended in the last Sunday of the year in 4 states (Florida, Missouri, New York, and Texas). The data columns are introduced below. One quick note about the columns of the data set: `week_start` as a column is present in the data set for illustration purposes, reminding users what `week` column is. In other words, it does not participate any visualization.

Usage

```
states_tg
```

Format

A data frame with 1116 rows and 6 columns:

state state

week week 1 to week 30.

week_start The Monday date of the week started.

category 9 Covid-related categories in total.

Twitter weekly tweets percentage (%) in state falling into each category.

Google weekly Google search percentage (%) in state falling into each category.

Source

Just like `pitts_tg`, Google is processed from Google Health API, and Twitter from Meltwater, a Twitter vendor. Both data sources are processed by the authors of the package.

theme_heat *Plot Themes*

Description

Plot Themes

Usage

```
theme_heat(
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

Arguments

base_size	base font size
base_family	base font family
base_line_size	base size for line elements
base_rect_size	base size for rect elements

Value

Adding a heat theme to all plots generated by using the ggDoubleHeat package.

Index