

Package: tastypie (via r-universe)

September 1, 2024

Title Easy Pie Charts

Version 0.1.1

Description You only need to type 'why pie charts are bad' on Google to find thousands of articles full of (valid) reasons why other types of charts should be preferred over this one. Therefore, because of the little use due to the reasons already mentioned, making pie charts (and related) in R is not straightforward, so other functions are needed to simplify things. In this R package there are useful functions to make 'tasty' pie charts immediately by exploiting the many cool templates provided.

License GPL-3

URL <https://paolodalena.github.io/tastypie/>

BugReports <https://github.com/PaoloDalena/tastypie/issues>

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports ggplot2, dplyr, scales, RColorBrewer, shadowtext, tibble, packcircles, fmsb

Depends R (>= 2.10)

Suggests knitr, markdown, rmarkdown, testthat (>= 3.0.0), jpeg, patternplot

VignetteBuilder knitr

Config/testthat/edition 3

Repository <https://paolodalena.r-universe.dev>

RemoteUrl <https://github.com/paolodalena/tastypie>

RemoteRef HEAD

RemoteSha 6293890c8c35b3b24d06633b3b55a395eacabf38

Contents

bubble_blow	2
pie_addimages	3
pie_bake	5
pie_bake_pro	6
pie_datacheck	8
pie_discover	9
pie_templates	10
pie_template_list	10
pie_template_list_pro	11
tastypie	11
Index	13

bubble_blow	<i>Easily create circular packing charts</i>
-------------	--

Description

This function allows you to create cool circular packing charts easily by providing just the data and a template among the available ones. Moreover, you can choose whether and where to display percentages and also optionally specify a title for the chart.

Usage

```
bubble_blow(data, template, perc = "no", title = "")
```

Arguments

data	A data frame (or a tibble) with two variables (columns): <ul style="list-style-type: none"> • in the first one there must be the vector of labels; • in the second one there must be the vector of values. You can use pie_datacheck to understand if the data is suitable.
template	The chosen template. The available templates for this function are 'bub1', 'bub2', 'bub3', 'bub4', 'bub5'.
perc	You can choose among: <ul style="list-style-type: none"> • 'no' : the percentages won't be displayed (default choice); • 'below' : the percentages will be displayed below the group labels; • 'right' : the percentages will be displayed next to the group labels.
title	A string. If you want, you can specify the title of the graph.

Value

A ggplot object. In particular, this function returns a circular packing chart according to the data, the choice of the template, and the other specifications provided.

See Also

See all the available templates displayed [here!](#)

Examples

```
example <- data.frame(
  c("This", "Is", "Just", "An", "Example"),
  c(2.9, 6.9, 4.20, 13.12, 6.66)
)
bubble_blow(
  data = example,
  template = "bub1",
  perc = "below",
  title = "Example1"
)

example2 <- tibble::tibble(
  c("cat 1", "cat2", "cat3", "cat4", "cat5", "cat6", "cat7", "cat8", "cat9"),
  c(324, 432, 499, 291, 750, 836, 314, 133, 372)
)
bubble_blow(
  data = example2,
  template = "bub2",
  perc = "right",
  title = "Ex2"
)
```

pie_addimages

Add images to pie charts

Description

This function allows to easily add some figures to you pie chart, taking information from the output of the [pie_bake](#) function. You can also customize the chart by specifying the colors of borders and labels, the title of the pie chart and where (and if) you want to print the percentages for each group.

Usage

```
pie_addimages(
  mypie,
  imglst,
  perc = "no",
  lbl_col = "black",
```

```
border_col = "black",
title = ""
)
```

Arguments

mypie	A ggplot object. The output from <code>pie_bake</code> , NOT from <code>pie_bake_pro</code> . Don't focus on the template, only labels and values will be inherited.
imglist	A list of objects returned by and <code>readJPEG</code> used to fill slices. For optimal and fast results, we recommend the use of small files (5-10 kB).
perc	You can choose among: <ul style="list-style-type: none"> • 'no' : the percentages won't be displayed (default choice); • 'below' : the percentages will be displayed below the group labels; • 'right' : the percentages will be displayed next to the group labels.
lbl_col	A string containing the chosen color for the labels, default is "black".
border_col	A string containing the chosen color for the border of the pie chart, default is "black".
title	A string. If you want, you can specify the title of the graph.

Value

A ggplot object.

See Also

Please note that this function is based on the `imagepie` function.

Examples

```
img5 <- jpeg::readJPEG(system.file("img", "pie.jpeg", package = "tastypie"))
imgs2 <- list(img5, img5, img5)

df2 <- data.frame(
  c("A", "B", "C"),
  c(300, 250, 600)
)

mypie2 <- pie_bake(df2, template = "rainbow1")

pie_addimages(
  mypie = mypie2,
  imglist = imgs2,
  perc = "right",
  lbl_col = "darkcyan",
  border_col = "orangered",
  title = "Example"
)
```

`pie_bake`*Easily create pie charts*

Description

This function allows you to create cool pie charts easily by providing just the data and a template among the available ones. Moreover, you can choose whether to display percentages or not and also optionally specify a title for the chart and a name for the categories.

Using this function you can create the *classical* pie charts (including the donut charts). If you are looking for something more *complex and extravagant* (but probably less understandable), check out the [pie_bake_pro](#) function.

Usage

```
pie_bake(data, template, perc = FALSE, group_name = "group", title = "")
```

Arguments

<code>data</code>	A data frame (or a tibble) with two variables (columns): <ul style="list-style-type: none">• in the first one there must be the vector of labels;• in the second one there must be the vector of values. Please note that the labels are automatically sorted in alphabetical order. If you want to specify a particular order, it is recommended to type numbers or letters before the category names (e.g. "a. category1", "b. category2", ...). You can use pie_datacheck to understand if the data is suitable.
<code>template</code>	The chosen template. Type <code>pie_template_list</code> to display all the available ones for this function.
<code>perc</code>	A logical value. Should the proportions be displayed?
<code>group_name</code>	A string. If you want, you can specify a name for the categories.
<code>title</code>	A string. If you want, you can specify the title of the graph.

Value

A ggplot object. In particular, this function returns a pie (or donut) chart according to the data, the choice of template, and the other specifications provided.

See Also

See all the available templates displayed [here!](#)

Examples

```
example <- data.frame(
  c("a. This", "b. Is", "c. Just", "d. An", "e. Example"),
  c(2.9, 6.9, 4.20, 13.12, 6.66)
)
pie_bake(
  data = example,
  template = "basic3",
  perc = TRUE,
  group_name = "groups",
  title = "Example1"
)

pie_bake(
  data = example,
  template = "red1",
  title = "Ex2"
)

pie_bake(
  data = example,
  template = "donut2",
  perc = TRUE,
  group_name = "CAT:"
)
```

pie_bake_pro

Easily create (more complex) pie charts

Description

This function allows you to create cool pie charts easily by providing just the data and a template among the available ones. Moreover, you can optionally specify a title for the chart and a name for the categories.

Using this function you can create some *complex and extravagant* pie charts. If you are looking for something more *classical* (and probably more understandable), check out the [pie_bake](#) function.

Usage

```
pie_bake_pro(data, template, group_name = "group", title = "")
```

Arguments

data A data frame (or a tibble) with two variables (columns):

- in the first one there must be the vector of labels;
- in the second one there must be the vector of values.

Please note that the labels are automatically sorted in alphabetical order. If you want to specify a particular order, it is recommended to type numbers or letters before the category names (e.g. "a. category1", "b. category2", ...). You can use [pie_datacheck](#) to understand if the data is suitable.

template	The chosen template. Type <code>pie_template_list_pro</code> to display all the available ones for this function.
group_name	A string. If you want, you can specify a name for the categories.
title	A string. If you want, you can specify the title of the graph.

Value

A ggplot object (or no value if the chosen template is among the spider chart ones). In particular, this function returns a pie (or similar) chart according to the data, the choice of template, and the other specifications provided.

See Also

See all the available templates displayed [here!](#)

Examples

```
example <- data.frame(
  c("a. This", "b. Is", "c. Just", "d. An", "e. Example"),
  c(2.9, 6.9, 4.20, 13.12, 6.66)
)
pie_bake_pro(
  data = example,
  template = "eaten3",
  group_name = "cat:",
  title = "Example1"
)

pie_bake_pro(
  data = example,
  template = "dart1",
  title = "Example2!"
)

pie_bake_pro(
  data = example,
  template = "eye5",
  group_name = "GROUPS:"
)

pie_bake_pro(
  data = example,
```

```
template = "spider2"  
)
```

pie_datacheck

Check if the data is good for making tasty pies

Description

pie_datacheck checks if the provided data are suitable for creating pie charts using the useful functions in the tastypie package.

Usage

```
pie_datacheck(data, check = FALSE)
```

Arguments

data	The data that you want to use for creating pie charts.
check	Logical, set equal to TRUE if you need a message to know if there are no problems.

Value

If the provided data are a dataframe with only two variables (columns) with the vector of labels in the first one and the vector of values in the second one, nothing will happen. Otherwise, an error that tells you what's wrong occurs.

Examples

```
wrong <- c(1, 2, 3)  
# Would return an Error  
  
wrong2 <- data.frame("a" = c(1, 2, 3), "b" = c("ex", "am", "ple"))  
# Would return an Error  
  
right <- data.frame("a" = c("ex", "am", "ple"), "b" = c(1, 2, 3))  
pie_datacheck(right) # No Error ==> OK!  
pie_datacheck(right, check = TRUE) # Positive message
```

pie_discover	<i>Discover templates for tastypie</i>
--------------	--

Description

This function allows to user to find out a random combination of templates, number of groups and features in order to get an idea of the many available plots in the package `tastypie`.

Usage

```
pie_discover(which = c("all", "bake", "pro"))
```

Arguments

<code>which</code>	Allows to select a subset of the available templates. If "all", the template is randomly chosen among all the available ones. If "bake", the template is randomly chosen among the ones that can be used through pie_bake . If "pro", the template is randomly chosen among the (more complex) ones that can be used through pie_bake_pro . The default value is "all".
--------------------	---

Value

A `ggplot` object. In particular, this function returns a pie (or similar) chart according to a random choice of template and the other features.

See Also

See all the available templates displayed [here!](#)

Examples

```
pie_discover()  
pie_discover("pro")
```

pie_templates	<i>Display an example of a particular template</i>
---------------	--

Description

This function allows to user to display a pie chart by selecting the template, the number of groups of interest and whether to display the proportions or not, in order to make it easier to choose between the many templates available in the package `tastypie`.

Usage

```
pie_templates(template, n_groups = 4, perc = FALSE)
```

Arguments

template	The chosen template.
n_groups	A number from 2 to 9.
perc	A logical value. Should the proportions be displayed? Note that if the selected template is one of those to be used with <code>pie_bake_pro</code> (listed in the <code>pie_template_list_pro</code> vector), this argument is useless.

Value

A ggplot object. In particular, this function returns a pie (or similar) chart according to the choice of template and the other specifications provided.

See Also

See all the available templates displayed [here!](#)

Examples

```
pie_templates(template = "bw1", n_groups = 3, perc = TRUE)
pie_templates(template = "watermelon2", n_groups = 8)
```

pie_template_list	<i>Available templates in tastypie for pie_bake.</i>
-------------------	--

Description

The vector containing all the available templates for pie charts for the `pie_bake` function. Other more complex templates are available in `pie_template_list_pro` that can be used through the `pie_bake_pro` function.

Usage

pie_template_list

Format

An object of class character of length 30.

See Also

See all the available templates displayed [here!](#)

pie_template_list_pro *Available templates in tastypie for pie_bake_pro.*

Description

The vector containing all the available templates for pie charts for the [pie_bake_pro](#) function. Other less complex templates are available in [pie_template_list](#) that can be used through the [pie_bake](#) function.

Usage

pie_template_list_pro

Format

An object of class character of length 30.

See Also

See all the available templates displayed [here!](#)

tastypie *Easy pie charts with tastypie*

Description

You only need to type [why pie charts are bad](#) on Google to find thousands of articles full of (*valid*) reasons why other types of charts should be preferred over this one. Therefore, because of the little use due to the reasons already mentioned, making pie charts (and related) in R is not straightforward, so other functions are needed to simplify things. In this R package there are useful functions to make *tasty* pie charts immediately by exploiting the many cool templates provided.

Author(s)

Author and Maintainer: Paolo Dalena <paolodalena97@gmail.com>

See Also

Find more information on the package web site [here](#), or on the GitHub package page [here](#).

Index

* datasets

pie_template_list, [10](#)

pie_template_list_pro, [11](#)

bubble_blow, [2](#)

imagepie, [4](#)

pie_addimages, [3](#)

pie_bake, [3](#), [4](#), [5](#), [6](#), [9-11](#)

pie_bake_pro, [4](#), [5](#), [6](#), [9-11](#)

pie_datacheck, [2](#), [5](#), [7](#), [8](#)

pie_discover, [9](#)

pie_template_list, [10](#), [11](#)

pie_template_list_pro, [10](#), [11](#)

pie_templates, [10](#)

readJPEG, [4](#)

tastypie, [11](#)

tastypie-package (tastypie), [11](#)