

Package ‘patternplot’

June 24, 2018

Type Package

Title Versatile Pie Charts, Bar Charts and Box Plots using Patterns,
Colors and Images

Version 0.2.1

Maintainer Chunqiao Luo <chunqiaoluo@gmail.com>

Imports Rcpp (>= 0.11.0), R6 (>= 2.1.2), ggplot2 (>= 2.1.0), jpeg (>= 0.1-8), png (>= 0.1-7), grDevices, utils, RcppParallel, dplyr, gtable

Depends R (>= 3.0.2)

Suggests knitr, Cairo, RCurl, rmarkdown

Date 2018-6-14

License GPL

Description Creates aesthetically pleasing and informative pie charts, bar charts and box plots with colors, patterns, and images.

LinkingTo Rcpp, RcppParallel

SystemRequirements GNU make

RoxygenNote 6.0.1

NeedsCompilation yes

VignetteBuilder knitr

Author Chunqiao Luo [aut, cre],
Shasha Bai [aut]

Repository CRAN

Date/Publication 2018-06-24 05:12:24 UTC

R topics documented:

imagebar	2
imageboxplot	3
imagepie	5
patternbar	6
patternboxplot	8
patternpie	10

imagebar	<i>Plot a bar chart with bars filled with png and jpeg images.</i>
----------	--

Description

The imagebar function is a tool for creating versatile bar charts by filling the bars with external png and jpeg images.

Usage

```
imagebar(data, x, y, group = NULL, xlab = "", ylab = "",
  label.size = 3.5, vjust = -1, pattern.type, frame.color = "black",
  frame.size = 1, legend.type = "h", legend.h = 6, legend.y.pos = 0.5,
  legend.pixel = 5)
```

Arguments

data	the data to be used.
x	the variable used on x axis.
y	the variable used on y axis.
group	the variable used as the second grouping variable on x axis.
xlab	a character string to give x axis label.
ylab	a character string to give y axis label.
label.size	the font size of labels shown above the bars.
vjust	the distance of labels from the top border of each bar.
pattern.type	a list of objects returned by readPNG and readJPEG used to fill the bars.
frame.color	the color of the borders of bars.
frame.size	a numeric value, the line size for the borders of bars.
legend.type	if legend.type='h', the layout of legends is horizontal; if legend.type='v', the layout of legends is vertical.
legend.h	a numeric value to change the height of legend boxes.
legend.y.pos	a numeric value to change the position of legends on y axis.
legend.pixel	a numeric value to change the pixel of legends.

Details

imagebar function offers flexible ways in doing bar charts.

Value

A ggplot object.

Author(s)

Chunqiao Luo (chunqiaoluo@gmail.com)

See Also

Function patternbar

Examples

```
library(patternplot)
library(jpeg)

#http://all-free-download.com/free-vector/modern-seamless-pattern.html
childcare<-readJPEG(system.file("img", "childcare.jpg", package="patternplot"))
food<-readJPEG(system.file("img", "food.jpg", package="patternplot"))
housing <-readJPEG(system.file("img", "housing.jpg", package="patternplot"))

#Example 1
data <- read.csv(system.file("extdata", "monthlyexp.csv", package="patternplot"))
data<-data[which(data$Location=='City 1'),]
x<-factor(data$Type, c('Housing', 'Food', 'Childcare'))
y<-data$Amount
pattern.type<-list(housing, food, childcare)
imagebar(data,x, y,group=NULL,pattern.type=pattern.type,
         frame.color=c('darkred', 'darkgreen', 'orange'),
         ylab='Monthly Expenses, Dollars')
```

imageboxplot

Plot a boxplot with boxes filled with png and jpeg images.

Description

The imageboxplot function is a tool for creating versatile boxplots by filling the boxplots with external png and jpeg images.

Usage

```
imageboxplot(data, x, y, group = NULL, xlab = "", ylab = "", pattern.type,
             frame.color = "black", linetype = "solid", frame.size = 1,
             outlier.shape = 21, outlier.color = "black", outlier.size = 1,
             legend.type = "h", legend.h = 6, legend.x.pos = 0.5,
             legend.y.pos = 0.5, legend.pixel = 0.3, legend.label.size = 3,
             legend.ratio1 = 0.1)
```

Arguments

<code>data</code>	the data to be used.
<code>x</code>	the variable used on x axis.
<code>y</code>	the variable used on y axis.
<code>group</code>	the variable used as the second grouping variable on x axis.
<code>xlab</code>	a character string to give x axis label.
<code>ylab</code>	a character string to give y axis label.
<code>pattern.type</code>	a list of objects returned by <code>readPNG</code> and <code>readJPEG</code> used to fill boxplots.
<code>frame.color</code>	the color for the borders of boxplots.
<code>linetype</code>	the linetype for the borders of boxplots.
<code>frame.size</code>	a numeric value, the line size for the borders of boxplots.
<code>outlier.shape</code>	the shape of outlier dots.
<code>outlier.color</code>	the color of outlier dots.
<code>outlier.size</code>	the size of outlier dots.
<code>legend.type</code>	if <code>legend.type='h'</code> , the layout of legends is horizontal; if <code>legend.type='v'</code> , the layout of legends is vertical.
<code>legend.h</code>	a numeric value to fine-tune the width of legend boxes on y axis.
<code>legend.x.pos</code>	a numeric value to change the position of legend text on x axis.
<code>legend.y.pos</code>	a numeric value to change the position of legend text on y axis.
<code>legend.pixel</code>	a numeric value to change the pixel of legend boxes.
<code>legend.label.size</code>	the font size of legend labels.
<code>legend.ratio1</code>	a numeric value to fine-tune the position of legend boxes on y axis.

Details

`imageboxplot` function offers flexible ways in doing boxplots.

Value

A `ggplot` object.

Author(s)

Chunqiao Luo (chunqiaoluo@gmail.com)

See Also

Function `patternboxplot`

Examples

```

library(patternplot)
library(jpeg)

Orange<-readJPEG(system.file("img", "oranges.jpg", package="patternplot"))
Strawberry <-readJPEG(system.file("img", "strawberries.jpg", package="patternplot"))
Watermelon<-readJPEG(system.file("img", "watermelons.jpg", package="patternplot"))

#Example 1
data <- read.csv(system.file("extdata", "fruits.csv", package="patternplot"))
x<-data$Fruit
y<-data$Weight
group<-data$Store
pattern.type<-list(Orange, Strawberry, Watermelon)
imageboxplot(data,x, y,group=NULL,pattern.type=pattern.type,
              frame.color=c('orange', 'darkred', 'darkgreen'),ylab='Weight, Pounds')

```

imagepie

Plot a pie chart with slices filled with png and jpeg images.

Description

The imagepie function is a tool for creating versatile pie charts by filling the slices with external png and jpeg images.

Usage

```

imagepie(group, pct, label, label.size = 4, label.color = "black",
         label.distance = 1.35, pattern.type, frame.color = "black",
         frame.size = 1)

```

Arguments

group	a vector of strings, containing the names of each slice.
pct	a vector of non-negative numbers, containing percentages of each group. The numbers must sum up to 100.
label	a vector of strings, giving the names for the slices shown in the pie chart.
label.size	the font size of labels shown in the pie chart.
label.color	the color of labels shown in the pie chart.
label.distance	the distance of labels from the border of the pie chart.
pattern.type	a list of objects returned by readPNG and readJPEG used to fill slices.
frame.color	the color for the borders of slices.
frame.size	a numeric value, the line size for the borders of slices.

Details

imagepie function offers flexible ways in doing pie charts.

Value

A ggplot object.

Author(s)

Chunqiao Luo (chunqiaoluo@gmail.com)

See Also

Function patternpie

Examples

```
library(patternplot)
library(jpeg)
Tomatoes <- readJPEG(system.file("img", "tomatoes.jpg", package="patternplot"))
Peas <- readJPEG(system.file("img", "peas.jpg", package="patternplot"))
Potatoes <- readJPEG(system.file("img", "potatoes.jpg", package="patternplot"))

#Example 1
data <- read.csv(system.file("extdata", "vegetables.csv", package="patternplot"))
pattern.type<-list(Tomatoes,Peas,Potatoes)
imagepie(group=data$group,pct=data$pct,label=data$label,pattern.type=pattern.type,
          label.distance=1.25,frame.color='burlywood4', frame.size=0.8, label.size=6,
          label.color='forestgreen')
```

patternbar

Plot a bar chart using patterns and colors to fill the bars.

Description

The patternbar function is a tool for creating versatile bar charts by filling the bars with colors and patterns.

Usage

```
patternbar(data, x, y, group = NULL, xlab = "", ylab = "",
           label.size = 3.5, vjust = -1, pattern.type, pattern.line.size = 1,
           frame.size = 1, pattern.color = rep("black", ifelse(is.null(group),
           length(x), length(unique(group))))), background.color = rep("white",
           ifelse(is.null(group), length(x), length(unique(group))))), pixel = 1,
           density = rep(7, ifelse(is.null(group), length(x), length(unique(group))))),
```

```

frame.color = rep("black", ifelse(is.null(group), length(x),
length(unique(group))))), legend.type = "h", legend.h = 6,
legend.y.pos = 0.5, legend.pixel = 0.1)

```

Arguments

<code>data</code>	the data to be used.
<code>x</code>	the variable used on x axis.
<code>y</code>	the variable used on y axis.
<code>group</code>	the variable used as the second grouping variable on x axis.
<code>xlab</code>	a character string to give x axis label.
<code>ylab</code>	a character string to give y axis label.
<code>label.size</code>	the font size of labels shown above the bars.
<code>vjust</code>	the distance of labels from the top border of each bar.
<code>pattern.type</code>	a vector of patterns to be filled in the bars There are 15 pattern types: 'blank', 'bricks', 'circles1', 'circles2', 'vdashes', 'hdashes', 'crosshatch', 'dots', 'grid', 'hlines', 'nelines', 'nwlines', 'shells', 'vlines', 'waves'.
<code>pattern.line.size</code>	a numeric value, the line size for the lines/dots of patterns.
<code>frame.size</code>	a numeric value, the line size for the borders of bars.
<code>pattern.color</code>	a vector of colors for the lines/dots of patterns.
<code>background.color</code>	a vector of colors to be filled in the bars.
<code>pixel</code>	a numeric value, the pixel resolution of bar chart.
<code>density</code>	a numeric vector, the density for the lines/dots of patterns.
<code>frame.color</code>	the color for the borders of bars.
<code>legend.type</code>	if <code>legend.type='h'</code> , the layout of legends is horizontal; if <code>legend.type='v'</code> , the layout of legends is vertical.
<code>legend.h</code>	a numeric value to change the height of legend boxes.
<code>legend.y.pos</code>	a numeric value to change the position of legends on y axis.
<code>legend.pixel</code>	a numeric value to change the pixel of legends.

Details

`patternbar` function offers flexible ways in doing bar charts.

Value

A `ggplot` object.

Author(s)

Chunqiao Luo (chunqiaoluo@gmail.com)

See Also

Function `patternbar`

Examples

```
#Example 1
data <- read.csv(system.file("extdata", "monthlyexp.csv", package="patternplot"))
data<-data[which(data$Location=='City 1'),]
x<-factor(data$Type, c('Housing', 'Food', 'Childcare'))
y<-data$Amount
pattern.type<-c('hdashes', 'blank', 'crosshatch')
pattern.color=c('black','black', 'black')
background.color=c('white','white', 'white')
density<-c(20, 20, 10)
patternbar(data,x, y,group=NULL,ylab='Monthly Expenses, Dollars', pattern.type=pattern.type,
           pattern.color=pattern.color, background.color=background.color,
           pattern.line.size=0.5,frame.color=c('black', 'black', 'black'), density=density)

#Example 2
pattern.color=c('black','white', 'grey20')
background.color=c('lightgreen','lightgreen', 'lightgreen')
patternbar(data,x, y,group=NULL,ylab='Monthly Expenses, Dollars', pattern.type=pattern.type,
           pattern.color=pattern.color, background.color=background.color,
           pattern.line.size=0.5,frame.color=c('black', 'black', 'black'), density=density)
```

`patternboxplot`

Plot a boxplot using patterns and colors to fill the boxes.

Description

The `patternboxplot` function is a tool for creating versatile boxplots by filling the boxplots with colors and patterns.

Usage

```
patternboxplot(data, x, y, group = NULL, xlab = "", ylab = "",
              outlier.shape = 21, outlier.color = "black", outlier.size = 1,
              linetype = rep("solid", ifelse(is.null(group), length(unique(x)),
              length(unique(group))))) , pattern.type, pattern.color = rep("black",
              ifelse(is.null(group), length(unique(x)), length(unique(group))))) ,
              pattern.line.size = 1, background.color = rep("white",
              ifelse(is.null(group), length(unique(x)), length(unique(group))))) ,
              frame.color = rep("black", ifelse(is.null(group), length(unique(x)),
              length(unique(group))))) , frame.size = 1, pixel = 1, density = rep(7,
              ifelse(is.null(group), length(unique(x)), length(unique(group))))) ,
              legend.type = "h", legend.h = 6, legend.x.pos = 0.5,
              legend.y.pos = 0.5, legend.pixel = 0.3, label.size = 3.5,
              legend.ratio1 = 0.1)
```


Arguments

<code>data</code>	the data to be used.
<code>x</code>	the variable used on x axis.
<code>y</code>	the variable used on y axis.
<code>group</code>	the variable used as the second grouping variable on x axis.
<code>xlab</code>	a character string to give x axis label.
<code>ylab</code>	a character string to give y axis label.
<code>outlier.shape</code>	the shape of outlier dots.
<code>outlier.color</code>	the color of outlier dots.
<code>outlier.size</code>	the size of outlier dots.
<code>linetype</code>	the linetype for the borders of boxplots.
<code>pattern.type</code>	a vector of patterns to be filled in the boxes There are 15 pattern types: 'blank', 'bricks', 'circles1', 'circles2', 'vdashes', 'hdashes', 'crosshatch', 'dots', 'grid', 'hlines', 'nelines', 'nwlines', 'shells', 'vlines', 'waves'.
<code>pattern.color</code>	a vector of colors for the lines/dots of patterns.
<code>pattern.line.size</code>	a numeric value, the line size for the lines/dots of patterns.
<code>background.color</code>	a vector of colors to be filled in the boxes.
<code>frame.color</code>	the color for the borders of boxes.
<code>frame.size</code>	a numeric value, the line size for the borders of boxes.
<code>pixel</code>	a numeric value, the pixel resolution of boxplot.
<code>density</code>	a numeric vector, the density for the lines/dots of patterns.
<code>legend.type</code>	if <code>legend.type='h'</code> , the layout of legends is horizontal; if <code>legend.type='v'</code> , the layout of legends is vertical.
<code>legend.h</code>	a numeric value to change the height of legend boxes.
<code>legend.x.pos</code>	a numeric value to change the position of legend text on x axis.
<code>legend.y.pos</code>	a numeric value to change the position of legend text on y axis.
<code>legend.pixel</code>	a numeric value to change the pixel of legends.
<code>label.size</code>	the font size of labels shown above the boxplots.
<code>legend.ratio1</code>	a numeric value to fine-tune the position of legend boxes on y axis.

Details

`patternboxplot` function offers flexible ways in doing boxplots.

Value

A `ggplot` object.

Author(s)

Chunqiao Luo (chunqiaoluo@gmail.com)

See Also

Function imageboxplot

Examples

```
library(patternplot)
library(jpeg)

Orange<-readJPEG(system.file("img", "oranges.jpg", package="patternplot"))
Strawberry <-readJPEG(system.file("img", "strawberries.jpg", package="patternplot"))
Watermelon<-readJPEG(system.file("img", "watermelons.jpg", package="patternplot"))

#Example 1
data <- read.csv(system.file("extdata", "fruits.csv", package="patternplot"))
x<-data$Fruit
y<-data$Weight
group<-data$Store
pattern.type<-list(Orange, Strawberry, Watermelon)
imageboxplot(data,x, y,group=NULL,pattern.type=pattern.type,
              frame.color=c('orange', 'darkred', 'darkgreen'),ylab='Weight, Pounds')
```

patternpie

Plot a pie chart using patterns and colors to fill the slices.

Description

The patternpie function is a tool for creating versatile pie charts by filling the slices with colors and patterns.

Usage

```
patternpie(group, pct, label, label.size = 4, label.color = "black",
           label.distance = 1.2, pattern.type, pattern.color = rep("black",
           length(group)), pattern.line.size = 1, background.color = rep("white",
           length(group)), frame.color = "black", frame.size = 1, pixel = 0.3,
           density = rep(10, length(group)))
```

Arguments

group	a vector of strings, containing the names of each slice.
pct	a vector of non-negative numbers, containing percentages of each group. The numbers must sum up to 100.
label	a vector of strings, giving the names for the slices shown in the pie chart.

label.size	the font size of labels shown in the pie chart.
label.color	the color of labels shown in the pie chart.
label.distance	the distance of labels from the border of the pie chart.
pattern.type	a vector of patterns to be filled in the slices. There are 15 pattern types: 'blank', 'bricks', 'circles1', 'circles2', 'vdashes', 'hdashes', 'crosshatch', 'dots', 'grid', 'hlines', 'nelines', 'nwlines', 'shells', 'vlines', 'waves'.
pattern.color	a vector of colors for the lines/dots of patterns.
pattern.line.size	a numeric value, the line size for the lines/dots of patterns.
background.color	a vector of colors to be filled in the slices.
frame.color	the color for the borders of slices.
frame.size	a numeric value, the line size for the borders of slices.
pixel	a numeric value, the pixel resolution of pie chart.
density	a numeric vector, the density for the lines/dots of patterns.

Details

patternpie function offers flexible ways in doing pie charts.

Value

A ggplot object.

Author(s)

Chunqiao Luo (chunqiaoluo@gmail.com)

See Also

Function imagepie

Examples

```
#Example 1
library(patternplot)
data <- read.csv(system.file("extdata", "vegetables.csv", package="patternplot"))
pattern.type<-c('hdashes', 'vdashes', 'bricks')
patternpie(group=data$group,pct=data$pct,label=data$label, pattern.type=pattern.type,
           pixel=0.3, pattern.line.size=0.2, frame.size=1.5)

#Example 2
pattern.color<-c('red3','green3', 'white' )
background.color<-c('dodgerblue', 'lightpink', 'orange')
patternpie(group=data$group,pct=data$pct,label=data$label, pattern.type=pattern.type,
           pattern.color=pattern.color,background.color=background.color,
           frame.color='grey40', pixel=0.3, pattern.line.size=0.3, frame.size=1.5)
```

Index

imagebar, [2](#)

imageboxplot, [3](#)

imagepie, [5](#)

patternbar, [6](#)

patternboxplot, [8](#)

patternpie, [10](#)