

Package ‘colorfulVennPlot’

February 19, 2015

Type Package

Title Plot and add custom coloring to Venn diagrams for 2-dimensional, 3-dimensional and 4-dimensional data.

Version 2.4

Date 2013-11-11

Author Elliot Noma, Aliona Manvae

Maintainer Elliot Noma <noma@garrettassetmanagement.com>

Description Given 2-,3- or 4-dimensional data, plots a Venn diagram, i.e. 'crossing circles'. The user can specify values, labels for each circle-group and unique colors for each plotted part. Here is what it would look like for a 3-dimensional plot: <http://elliotnoma.files.wordpress.com/2011/02/venndiagram.png>. To see what the 4-dimensional plot looks like, go to <http://elliotnoma.files.wordpress.com/2013/03/4dplot.png>.

Depends grid

Suggests MASS

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-12 18:52:29

R topics documented:

colorfulVennPlot-package	2
createVennData	3
plotVenn	5
plotVenn2d	6
plotVenn3d	7
plotVenn4d	8
resizeCircles	9

Index	11
--------------	-----------

colorfulVennPlot-package

Venn diagram with user-specified colors for 2-dimensional, 3-dimensional and 4-dimensional data

Description

Plot Venn diagram of 2-dimensional, 3-dimensional and 4-dimensional data with user-specified custom colors.

Details

Package: plotVenn
 Type: Package
 Version: 2.4
 Date: 2013-11-11
 License: GPL-2

The package is designed to be able to plot Venn diagrams for 2-d, 3-d and 4-d data. The uniqueness of it is in being able to specify custom colors for each part of the plot - including the crossovers.

If data needs to be shaped into appropriate format for it, `createVennData()` can be used. Given 3 columns of data and values by which to split the data, `createVennData()` splits it into 7 groups and counts values in each group or percent allocations. Notice: currently, it only can be used to generate 3-dimensional data.

Author(s)

Elliot Noma, Aliona Manvae

Maintainer: Elliot Noma <noma@garrettassetmanagement.com>

Examples

```
data(survey, package = "MASS")
# For more info: ?MASS::survey

# Grouping students by Pulse, Height and Age
vennData <- createVennData(survey,
  Cols = c("Pulse", "Height", "Age"),
  Splits = c(75, 170, 20), ## Split Pulse at 75, split Height at 170, split Age at 20
  Labels = NULL, ## Keep Labels NULL, as we wish to stick with Pulse, Height and Age titles.
  type = c("percent")) ## generate data as percents

# Plot Venn diagram
# Notice:
# plotVenn is a generic function, which can be used to call plotVenn2d, plotVenn3d or plotVenn4d
plot.new()
```

```

plotVenn(vennData$x, vennData$labels)

#####
# Example of 4-dimensional data plotting, with default data
plot.new()
plotVenn4d(Title = "Example of 4-d Venn diagram")

```

createVennData *Generates data for plotVenn3d()*

Description

Given data with 3 columns, it will generate a vector of 7 numerical values, which describe grouping of the data across 3 specified columns. The output should be used for plotVenn3d() function.

Usage

```

createVennData(x, Cols = NULL, Splits = c(0, 0, 0), Labels = NULL,
  type = c("count", "percent"), ToSkip = "000")

```

Arguments

x	a table of numerical values of at least 3 columns.
Cols	a numeric or character vector of length 3, specifying column names or numbers. If not specified, it defaults to 3 first columns of input data 'x'.
Splits	a numeric vector of length 3, specifying splitting value for each column. See below for more information.
Labels	a character vector of length 3, clarifying names of each group/column. If not specified, it defaults to value of variable 'Cols'.
type	specifies the type of output data. 'count' counts a number of each grouping 'percent' computes the percents of each grouping
ToSkip	defaults to '000'. Do not change it, unless you know exactly what you are doing.

Details

NOTICE: This only works for 3-dimensional data. Data 'x' should be a table of numerical values of at least 3 columns.

Each column's data is split by corresponding 'Splits' value into 'above' and 'below'. Then, it is gathered together in the form:

1 - how many rows of data 'x' had all 3 columns 'above' ("111")

2 - how many rows of data 'x' had 1st column 'below' Splits[1] and columns 2 & 3 'above' Splits[2:3] ("011")

3 - how many rows of data 'x' had 1st column 'above' Splits[1], 2nd 'below' Splits[2] and 3rd 'above' Splits[3] ("101")

4 - how many rows of data 'x' had 1st column 'below' Splits[1], 2nd 'below' Splits[2] and 3rd 'above' Splits[3] ("001")

5 - how many rows of data 'x' had 1st column 'above' Splits[1], 2nd 'above' Splits[2] and 3rd 'below' Splits[3] ("110")

6 - how many rows of data 'x' had 1st column 'below' Splits[1], 2nd 'above' Splits[2] and 3rd 'below' Splits[3] ("010")

7 - how many rows of data 'x' had 1st column 'above' Splits[1], 2nd 'below' Splits[2] and 3rd 'below' Splits[3] ("100")

8 - how many rows of data 'x' had all 3 columns 'below' ("000")

Once this vector is generated, it transforms it into percents, if specified in variable 'type'. At this point, the sum will add up to 100 percent.

Then, one value has to be dropped - by definition, a 3-dimensional Venn diagram plots 7 out of 8 values, the space around the plot being the last value. Generally, value '000' must be dropped - it is set as default, and we do not recommend changing it.

The output will be then a vector of length 7 rather than 8, missing the 'ToSkip' part.

Value

a list containing:

x a numeric vector of length 7 describing the counts/percents of split data. See 'Details' for more information.

labels a character vector of 3, specifying the names of the groups.

Author(s)

Aliona Manvae

Examples

```
data(survey, package = "MASS")
# For more info: ?MASS::survey

# Grouping students by Pulse, Height and Age
plot.new()
vennData <- createVennData(survey,
  Cols = c("Pulse", "Height", "Age"),
  Splits = c(75, 170, 20), ## Split Pulse at 75, split Height at 170, split Age at 20
  Labels = NULL, ## Keep Labels NULL, as we wish to stick with Pulse, Height and Age titles.
  type = c("percent")) ## generate data as percents

# Plot
plot.new()
plotVenn3d(vennData$x, vennData$labels)
# OR:
plot.new()
plotVenn(vennData$x, vennData$labels)
```

`plotVenn`*Plot Venn diagram for 2-, 3-, 4- dimensional data.*

Description

Plots a 2-, 3-, or 4-dimensional Venn plot depending on the number of input values. The user can specify values, labels for each circle-group and colors. If the vector is of length 3, a 2-d plot is produced. If the vector is of length 7, a 3-d plot is produced. If the vector is length 15, a 4-d plot is produced.

Usage

```
plotVenn(x, ...)
```

Arguments

`x` for a 3-d plot, a numeric vector of length 7, with a permutation of the names `c("001", "010", "011", "100", "101", "110", "111")`.

`...` Additional arguments accepted by `plotVenn2d`, `plotVenn3d` or `plotVenn4d`.

Details

The `'00'`, `'000'`, `'0000'` groups are not plotted, so `'percent'` data will not add up to 100 percent on the graph.

Note

It requires package `grid` to be installed, and can be plotted according to specified grid parameters.

Author(s)

Elliot Noma, Aliona Manvae

References

Original code: <http://elliottnoma.wordpress.com/2011/02/09/venn-diagram/>

Output file: <http://elliottnoma.files.wordpress.com/2011/02/venndiagram.png>

Examples

```
y <- c(37,29,6,232,121,77,25)
names(y) <- c("001", "010", "011", "100", "101", "110", "111")
labels <- c("A", "B", "C")
plot.new()
plotVenn(y, labels, Colors=rainbow(7))
```

plotVenn2d

Plot Venn diagram for 2-dimensional data.

Description

Given a vector of 3 values, which describe 2-dimensional data, it plots a Venn diagram, i.e. 'crossing circles'. The user can specify values, labels for each circle-group and colors.

Usage

```
plotVenn2d(x, labels = c('A', 'B'),
           Colors = c("red", "yellow", "green"),
           Title = NULL, shrink = 1, rot=0, radius= c(1,1), resizePlot = 1,
           reverseLabelOrdering=TRUE)
```

Arguments

x	a numeric vector of length 3, where first value corresponds to only group 1, second is only group 2, and third is a crossover of the two groups. Can also be a character vector of length 3.
labels	a character vector of length 2, providing names for the 2 groups/dimensions.
shrink	a numeric value, specifying zooming effect of the plot, defaults to 1.
Colors	a vector of color names for the backgrounds of each part of the diagram.
Title	optional: a character vector of length 1, specifying title for the whole plot.
rot	a numeric value, specifying the number of degrees to rotate the graph.
radius	a 2-item numeric vector containing the relative sizes of the two circles
resizePlot	a numeric value indicating the amount to increase or decrease the size of the plot
reverseLabelOrdering	boolean value indicating whether labels are in the normal or reverse ordering. default=TRUE for backward compatability with the original version.

Details

NOTICE: This only works for 2-dimensional data.

By definition, value '00' lies outside the plotted diagram. Therefore, it will not be plotted. Because of that, if plotting 'percent' data, all values will not add up to 100 percent, missing 00's allocation.

Note

It requires package grid to be installed, and can be plotted according to specified grid parameters.

Author(s)

Elliot Noma, Aliona Manvae

Examples

```
# plot generic X, Y and intersection
plot.new()
plotVenn2d(rep("",3), radius=c(1.25,1.25), labels=c("Y","X"), Colors=c("yellow","orange","pink"))
grid.text(expression(paste(X,intersect(Y))),0.5,0.5)
grid.text("X",0.25,0.5)
grid.text("Y",0.75,0.5)

# rotated plot
y <- c(37,29,6)
labels <- c("A","B")
plot.new()
plotVenn2d(y, labels, Colors=rainbow(3), Title = "This is an example of a 2D Venn plot",
  radius=c(0.85, 1.0), rot=45)
```

plotVenn3d

Plot Venn diagram for 3-dimensional data.

Description

Given a vector of 7 values, which describe 3-dimensional data, it plots up Venn diagram, i.e. 'crossing circles'. The user is able to specify values, labels for each circle-group and colors.

Usage

```
plotVenn3d(x, labels = c('A','B','C'),
  Colors = c("red","yellow","orange","lightblue","purple","green","grey"),
  Title = NULL, shrink=1, rot=0)
```

Arguments

x	a numeric vector of length 7, with names c("001","010","011","100","101","110","111") in suitable order.
labels	a character vector of length 3, providing names for the 3 dimensions.
shrink	a numeric value, specifying zooming effect of the plot.
Colors	a vector of color names for the backgrounds of each part of the diagram.
Title	a string containing the graph title.
rot	a numeric value for the number of degrees to rotate the graph.

Details

NOTICE: This only works for 3-dimensional data.

By definition, value '000' lays outside the plotted diagram. Therefore, it will not be plotted. Because of that, if plotting 'percent' data, all values will not add up to 100 percent, missing 000's allocation.

Note

It requires package grid to be installed, and can be plotted according to specified grid parameters.

Author(s)

Elliot Noma, Aliona Manvae

References

Original code: <http://elliotnoma.wordpress.com/2011/02/09/venn-diagram/>

Output file: <http://elliotnoma.files.wordpress.com/2011/02/venndiagram.png>

Examples

```
y <- c(37,29,6,232,121,77,25)
names(y) <- c("001", "010", "011", "100", "101", "110", "111")
labels <- c("A", "B", "C")
plot.new()
plotVenn3d(y, labels, Colors=rainbow(7), Title = "This is an example of a 3D Venn plot")
```

plotVenn4d

Plot Venn diagram for 4-dimensional data.

Description

Given a vector of 15 values, which describe 4-dimensional data, it plots a Venn diagram, i.e. 'crossing circles'. The user is able to specify values, labels for each circle-group and colors.

Usage

```
plotVenn4d(x, labels = c('A', 'B', 'C', 'D'),
  Colors = c('red', 'yellow', 'green', 'pink', 'darkgreen', 'blue', 'lightblue', 'tan',
  'yellowgreen', 'orange', 'purple', 'white', 'grey', 'plum', 'brown'),
  Title = NULL, shrink = 1, rot=45)
```

Arguments

x	a numeric vector of length 15, with names c('1000', '0100', '1100', '0010', '1010', '0110', '1110', '0001', '1001', '0101', '1101', '0011', '1011', '0111', '1111')) in suitable order.
labels	a character vector of length 4, providing names for the 4 dimensions.
shrink	a numeric value specifying zooming effect of the plot.
Colors	a vector of color names for the backgrounds of each part of the diagram.
Title	a string containing the graph title.
rot	a numeric value for the number of degrees to rotate the graph.

Details

NOTICE: This only works for 4-dimensional data.

By definition, value '0000' lays outside the plotted diagram. Therefore, it will not be plotted. Because of that, if plotting 'percent' data, all values will not add up to 100 percent, missing 0000's allocation.

Note

It requires package grid to be installed, and can be plotted according to specified grid parameters.

Author(s)

Elliot Noma, Aliona Manvae

Examples

```
plot.new()
Colors <- c('red', 'yellow', 'green', 'pink', 'darkgreen', 'blue', 'lightblue', 'tan',
  'yellowgreen', 'orange', 'purple', 'white', 'grey', 'plum', 'brown')
regions <- seq(15)
names(regions) <- c('1000', '0100', '1100', '0010', '1010', '0110', '1110', '0001',
  '1001', '0101', '1101', '0011', '1011', '0111', '1111')
plotVenn4d(regions, Colors=Colors, Title = '4-dimensional')
```

resizeCircles	<i>Match the areas of the 2 circles and their overlap to the input values for X, Y and the intersection of X and Y.</i>
---------------	---

Description

Calculates the radii of the two circles relative to a fixed center-to-center distance. The areas of the circles and their overlap will be proportionate the three input values of X alone, Y alone and the intersection of X and Y. The method used is described in <http://elliottnoma.wordpress.com/2013/03/17/spacing-circles-in-a-2-item-venn-diagram/>.

Usage

```
resizeCircles(x_only, y_only, overlap, standardDistance = sqrt(0.5))
```

Arguments

x_only	a numeric value representing the relative size of X excluding its intersection with Y.
y_only	a numeric value representing the relative size of Y excluding its intersection with X.

`overlap` a numeric value representing the relative size of the intersection of X and Y.
`standardDistance` a numeric value specifying the fixed center-to-center distance to which the radii are matched. The default value is that used by `plotVenn2d`

Details

This function is designed to be used in conjunction with the `plotVenn2d` function. `plotVenn2d` plots the circles with a fixed center-to-center distance as set by the `standardDistance` argument. This routine specifies the radii of the circles to control their overlap relative to the fixed center-to-center distance. To increase or decrease the center-to-center distance in `plotVenn2d`, use the `resizePlot` parameter.

The function assumes that the first two arguments are for X-excluding the intersection and Y-excluding the intersection. The total X amount equals this first and third arguments. The total Y amount equals the second and third arguments.

Author(s)

Elliot Noma

Examples

```
plot.new()  
plotVenn2d(rep("",3), radius=resizeCircles(20,.5, 3), Title=NULL,  
  resizePlot=0.7, labels=c("", ""))
```

Index

*Topic **hplot**

- plotVenn, [5](#)
- plotVenn2d, [6](#)
- plotVenn3d, [7](#)
- plotVenn4d, [8](#)

*Topic **manip**

- createVennData, [3](#)
- resizeCircles, [9](#)

*Topic **package**

- colorfulVennPlot-package, [2](#)

colorfulVennPlot

- (colorfulVennPlot-package), [2](#)

colorfulVennPlot-package, [2](#)

createVennData, [3](#)

plotVenn, [5](#)

plotVenn2d, [6](#)

plotVenn3d, [7](#)

plotVenn4d, [8](#)

resizeCircles, [9](#)