

Package ‘bvenn’

February 19, 2015

Type Package

Title A Simple alternative to proportional Venn diagrams

Version 0.1

Date 2012-08-08

Author Raivo Kolde <rkolde@gmail.com>

Maintainer Raivo Kolde <rkolde@gmail.com>

Description This package implements a simple alternative to the traditional Venn diagram. It depicts each overlap as a separate bubble with area proportional to the overlap size. Relation of the bubbles to input sets is shown by their their arrangement.

License GPL (>= 2)

Depends grid

Collate 'bvenn-package.r' 'bvenn.r'

Repository CRAN

Date/Publication 2012-08-09 12:57:55

NeedsCompilation no

R topics documented:

bvenn-package	1
bvenn	2

Index	4
--------------	----------

bvenn-package	<i>A Simple alternative to proportional Venn diagrams</i>
---------------	---

Description

A Venn diagram is a useful tool to visualise the overlap between sets of elements. However, often the proportions of the overlaps are not accurately depicted by the areas on the figure. In fact, for some configurations of the 3 sets it is even impossible. This means that the proportions of the sets and overlaps often still have to be deduced from the numbers associated with the figure.

Details

This package implements a simple alternative to the traditional Venn diagram, where we depict each overlap as a separate bubble with area proportional to the overlap size. Relation of the bubbles to input sets becomes clear from their arrangement.

These figures are much easier to read than normal approximately proportional Venn diagrams. First, the sets are always in the same positions on the plot and not moving around like on normal Venn diagrams. This feature makes it easy to compare the figures if several diagrams are drawn side-by-side. Second, the proportions are easier to read, because they all are depicted as similar shapes not as circles and cuts of circles as on ordinary Venn diagrams.

The function for drawing the figures is [bvenn](#).

 bvenn

Simple alternative to Venn diagrams

Description

A simple alternative to the traditional Venn diagram. It depicts each overlap as a separate bubble with area proportional to the overlap size.

Usage

```
bvenn(sets, percentage = FALSE, colors = NULL,
      fontsize = 15, main = "", scale = 1, add = FALSE)
```

Arguments

sets	list of vectors to overlap. If list contains more than 3 elements only the first 3 are used.
percentage	logical showing if percentages or raw numbers are displayed
colors	vector of colors for the bubbles, see details on specifying that
fontsize	fontsize used for the numbers in the bubbles
main	title of the plot
scale	a scaling factor to adjust the base triangle size when the plot does not fit the window well.
add	logical determining if the figure is added to existing plot or if a new plot is initialized

Details

Colors can be specified as vector. For 2 set case a 3 element vector is required with colors for: Set1, Set2 and Set1 & Set2 correspondingly. For 3 set case a 7 element vector is required with colors for: Set1, Set2, Set3, Set1 & Set2, Set1 & Set3, Set2 & Set3 and Set1 & Set2 & Set3 correspondingly.

Author(s)

Raivo Kolde <rkolde@gmail.com>

Examples

```
bvenn(list(Set1 = sample(letters, 14), Set2 = sample(letters, 9)))
bvenn(list(Set1 = sample(letters, 16), Set2 = sample(letters, 12), Set3 = sample(letters, 7)))

# Adding colors
bvenn(list(Set1 = sample(letters, 14), Set2 = sample(letters, 9)), colors = c("red",
"green", "yellow"))
bvenn(list(Set1 = sample(letters, 16), Set2 = sample(letters, 12), Set3 =
sample(letters, 7)), colors = c("red", "blue", "yellow", "purple", "orange", "green",
"brown"))

# Adjust the triangle size
bvenn(list(Set1 = sample(letters, 16), Set2 = sample(letters, 12), Set3 =
sample(letters, 7)), colors = c("red", "blue", "yellow", "purple", "orange", "green",
"brown"), scale = 0.7)

# Combine several diagrams using grid graphics
vplayout = function(x, y){
  return(viewport(layout.pos.row = x, layout.pos.col = y))
}
grid.newpage()
pushViewport(viewport(layout = grid.layout(ncol = 2, nrow = 2)))
for(i in 1:2){
  for(j in 1:2){
    pushViewport(vplayout(i, j))
    bvenn(list(Set1 = sample(letters, 16), Set2 = sample(letters, 3+ 3*j), Set3 = sample(letters, 7)), add = TRUE, for
upViewport()
  }
}
```

Index

bvenn, [2](#), [2](#)

bvenn-package, [1](#)