Package ‘nVennR’

January 24, 2021

Type Package
Title Create n-Dimensional, Quasi-Proportional Venn Diagrams
Version 0.2.3
Author Victor Quesada [aut, cre, cph]
Maintainer Victor Quesada <quesadavictor@uniovi.es>
Description Provides an interface for the ”nVenn” algorithm (Perez-Silva, Araujo-Voces and Quesada (2018), <doi:10.1093/bioinformatics/bty109>). This algorithm works for any number of sets, and usually yields pleasing and informative Venn diagrams with proportionality information. However, representing more than six sets takes a long time and is hard to interpret, unless many of the regions are empty. If you cannot make sense of the result, you may want to consider ’UpSetR’.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Depends R (>= 2.10)
LinkingTo Rcpp
Imports Rcpp
Suggests knitr, rmarkdown, rsvg, grImport2
VignetteBuilder knitr
NeedsCompilation yes
Repository CRAN
Date/Publication 2021-01-24 17:00:05 UTC

R topics documented:

createVennObj .............................................................. 2
exampledf ................................................................. 3
getVennRegion ............................................................. 3
createVennObj

Create nVennObj from scratch

Description

Create nVennObj from scratch

Usage

createVennObj(nSets = 1, sNames = NULL, sSizes = NULL)

Arguments

nSets  Number of sets.
sNames List of names.
sSizes List of sizes for all the regions (from '0' to '2**nSets - 1'). To understand the order of the regions, one can think of a region as a binary number. Each bit tells whether the region belongs (1) or not (0) to a given set. For instance, with 4 sets we have 4 bits. The number 7 with 4 bits is 0111, which describes a region belonging to sets 2, 3, and 4 and not to set 1. To pass the values of the regions, those values are sorted according to the number describing the region. Thus, with four sets, the first element corresponds to region 0 (0000), the second to region 1 (0001), the third to region 2 (0010), ... The last corresponds to region 15 (1111), which belongs to all the sets.

Value

nVennObj with set information. To plot, it must be sent to `toVenn`. Sending it to `showSVG` will render the diagram before simulation.
**exampledf**  

*Example data frame.*

**Description**

A dataset containing programming preferences from 18 employees. This data set was provided by user Krantz to inquire about nVennR

**Usage**

```r
df
```

**Format**

A data frame with 18 rows and 3 variables:

- **Employee** Employee ID
- **SAS** Employee uses SAS
- **Python** Employee uses Python
- **R** Employee uses R

**Source**


---

**getVennRegion**  

*Get elements in a region*

**Description**

Get elements in a region

**Usage**

```r
gv <- getVennRegion(nVennObj, region)
```

**Arguments**

- **nVennObj** Object describing an nVenn job.
- **region** Description of the region. This can be a vector with the names of the groups the region belongs to or a vector describing whether the region belongs to each set in order (i.e., `c(1, 0, 0)` means the region belongs to set 1 and does not belong to sets 2 and 3).

**Value**

list of the elements belonging to the specified region
listVennRegions  List elements in every region

Description

List elements in every region

Usage

listVennRegions(nVennObj, na.rm = T)

Arguments

nVennObj  Object to list.
na.rm  If true, empty regions are not listed.

plotVenn  Create Venn diagram using the nVenn algorithm.

Description

This algorithm is based on a simulation that compacts the figure. If the resulting diagram is not compact enough, the simulation can be tweaked in two ways: changing the number of simulation cycles (‘nCycles’) and executing this function repeatedly.

Usage

plotVenn(
  sets,
  nVennObj = NULL,
  nCycles = 7000,
  sNames = NULL,
  showPlot = T,
  ...
)

Arguments

sets  List of lists with the input sets.
nVennObj  Object returned from previous run. If provided, the function will improve the diagram by running more cycles on the previous result. If nVennObj is provided, do not feed additional input lists, as they will be ignored
nCycles  Number of cycles for the simulation. For up to 4 sets, the default number of 7000 should be enough. Even for more complex scenarios, it may be better to run the function repeatedly, as a large number of cycles may take up too many resources.
setVennRegion

sNames
List of set names, in the same order as the input lists. If the input has tables or data frames and the name exists, it will select the corresponding column.

showPlot
Show the result in the graphic device.

... Options for 'showSVG' If input lists have names, those names will be used for the legend. If not, names can be provided with sNames.

Value
nVennObj with the result of the simulation. As a side effect, The result can be drawn in the graphical device.

Examples
set1 <- list(set1 = c('a', 'b', 'c'))
set2 <- list(set2 = c('e', 'f', 'c'))
set3 <- list(set3 = c('c', 'b', 'e'))
myNV <- plotVenn(list(set1, set2, set3), sNames=c("One", "Two", "Three"))
showSVG(myNV, opacity=0.2)

---

Description
Set number of elements in a region

Usage
setVennRegion(nVennObj, region, value)

Arguments
nVennObj Object describing an nVenn job.
region Description of the region. This can be a vector with the names of the groups the region belongs to or a vector describing whether the region belongs to each set in order (i.e., c(1, 0, 0) means the region belongs to set 1 and does not belong to sets 2 and 3).
value Size of the region.

Value
Modified nVennObj
showSVG  

Show Venn diagram. Automatically called from plotVenn.

Description
Show Venn diagram. Automatically called from plotVenn.

Usage

```r
showSVG(
  nVennObj,
  opacity = 0.4,
  borderWidth = 1,
  showLegend = T,
  outFile = "",
  systemShow = FALSE,
  labelRegions = T,
  showNumbers = T,
  setColors = NULL,
  fontScale = 1
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nVennObj</td>
<td>Object with nVennR information. Can be obtained from a plotVenn call.</td>
</tr>
<tr>
<td>opacity</td>
<td>Fill opacity for the sets. Defaults to 0.4.</td>
</tr>
<tr>
<td>borderWidth</td>
<td>Width of set borders. Defaults to 1.</td>
</tr>
<tr>
<td>showLegend</td>
<td>Boolean stating whether the resulting figure should contain a legend. Defaults to true.</td>
</tr>
<tr>
<td>outFile</td>
<td>File name to save SVG figure. If empty, a temp file will be created and sent to the graphic device.</td>
</tr>
<tr>
<td>systemShow</td>
<td>Show the result in the system SVG viewer (i.e., Inkscape).</td>
</tr>
<tr>
<td>labelRegions</td>
<td>Show region identifiers. These are numbers in parentheses inside each region indicating which sets that region belongs to. Defaults to true</td>
</tr>
<tr>
<td>showNumbers</td>
<td>Show how many elements belong to each region (large numbers in the figure). Defaults to true</td>
</tr>
<tr>
<td>setColors</td>
<td>Vector with the color of each set in order. Color names must be CSS-compatible.</td>
</tr>
<tr>
<td>fontScale</td>
<td>Multiplier for font sizes. The font size of both numbers and region labels will be multiplied by this factor. Values larger than 2 will probably make labels clash.</td>
</tr>
</tbody>
</table>

Value
Processed nVenn object. Creates a Venn diagram in svg as a side effect.
Index

* datasets
  exampledf, 3
createVennObj, 2
exampledf, 3
getVennRegion, 3
listVennRegions, 4
plotVenn, 4
setVennRegion, 5
showSVG, 6