Package: nVennR2 (via r-universe)

November 16, 2026

Type Package

Title An Interface to 'nVenn2'

Version 2.0.4.9000

Date 2026-07-15

Description Creates quasi-proportional Venn diagrams with an arbitrary number of sets. It is related to the old 'nVennR' package, but the algorithm and use have been reworked.

License MIT + file LICENSE

Imports Rcpp (>= 1.0.14)

LinkingTo Rcpp

Encoding UTF-8

Suggests knitr, rmarkdown, rsvg, graphics, grImport2, testthat (>= 3.0.0)

Depends R (>= 4.1)

LazyData true

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

URL https://github.com/vqf/nVennR2

BugReports https://github.com/vqf/nVennR2/issues

Config/roxygen2/version 8.0.0

Repository https://vqf.r-universe.dev

Date/Publication 2026-07-16 09:59:58 UTC

RemoteUrl https://github.com/vqf/nvennr2

RemoteRef HEAD

RemoteSha 865ee39cae8a95b34162ef8d56515d67c24b6cef

Contents

estimateExhaustiveRunTime . . . 2
exampledf . . . 3
getVennRegion . . . 3
getVennSetNames . . . 4
getVennSvg . . . 5
listVennRegions . . . 5
nVennDiagram . . . 6
nVennR2 . . . 7
plotVenn . . . 8
readVennSVG . . . 9
rotateVenn . . . 10
setVennColor . . . 10
setVennColors . . . 11
setVennOpts . . . 12
setVennPalette . . . 13
setVennSkin . . . 14

Index **15**

`estimateExhaustiveRunTime`

Estimates the time taken by minimization steps when preparing a diagram with the exhaustive method at a given depth

---

Description

Estimates the time taken by minimization steps when preparing a diagram with the exhaustive method at a given depth

Usage

```
estimateExhaustiveRunTime(desc, maxlevel = 0L, byCol = 0L)
```

Arguments

`desc`	Description of sets, either as a file path, a list of lists, text or a previously created nVenn object (see `nVennDiagram()`).
`maxlevel`	Depth of the exhaustive search. See the `maxlevel` parameter of `nVennDiagram()` for details.
`byCol`	If the input is a text, this parameter indicates whether each set is a column (1) or a row (2). Defaults to 0, which means that the package will try to guess which possibility makes more sense.

Value

float Estimated time for steps 3 and 4 in seconds. In case of error, the returned value is -1.

Examples

```
estimateExhaustiveRunTime(exampledf, 4)
```

`exampledf` *Example data frame.*

Description

A dataset containing programming preferences from 18 employees. This data set was provided by user Krantz to inquire about nVennR2

Usage

```
exampledf
```

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

https://stackoverflow.com/questions/49471565/transforming-data-to-create-generalized-quasi-proporti

---

`getVennRegion` *Gets a list of the elements in a region of the diagram*

Description

Gets a list of the elements in a region of the diagram

Usage

```
getVennRegion(nVennObj, n)
```

Arguments

`nVennObj`	nVennR2 object generated with `nVennDiagram()`.
`n`	Region, either as an integer or as a vector of set names. See Details

Details

An integer expresses a region by considering its binary representation in reverse. A 1 in a position means "belongs to" and a 0 or empty position means "does not belong to". Thus, the binary representation of 19 (10011), read from right to left, means "region that belongs to sets 1, 2 and 5 and does not belong to any other set".

A vector of set names expresses a region by giving the sets the region belongs to. It is understood that the region does not belong to any other set.

Value

List of set names.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
#Both commands are equivalent:
getVennRegion(myv, 3)
getVennRegion(myv, c("Set1", "Set2"))
```

---

getVennSetNames *Gets the names of the sets*

Description

Gets the names of the sets

Usage

```
getVennSetNames(nVennObj)
```

Arguments

nVennObj	nVennR2 object generated with nVennDiagram().

Value

List of set names.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
getVennSetNames(myv)
```

getVennSvg *Get the svg code of an nVenn diagram*

Description

Get the svg code of an nVenn diagram

Usage

```
getVennSvg(nVennObj)
```

Arguments

nVennObj	nVennR2 object generated with nVennDiagram().

Details

The code returned by the funciton can be saved to a file and then edited with vectorial image software. This can be done directly with nVennDiagram() or plotVenn() by using the outFile param.

Value

String with svg code.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
getVennSvg(myv)
```

listVennRegions *Lists the elements in every region of the diagram*

Description

Lists the elements in every region of the diagram

Usage

```
listVennRegions(nVennObj, showEmpty = FALSE)
```

Arguments

nVennObj	nVennR2 object generated with nVennDiagram().
showEmpty	If true, lists every region, even if empty. If false (default), only lists regions containing elements.

Value

List of non-empty regions with the elements contained

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
mylist <- listVennRegions(myv)
mylist
```

`nVennDiagram` *Creates nVenn plot*

Description

Creates nVenn plot

Usage

```
nVennDiagram(
  desc,
  plot = TRUE,
  outFile = "",
  systemShow = FALSE,
  verbose = TRUE,
  maxlevel = 0L,
  byCol = 0L
)
```

Arguments

`desc`	Description of sets, either as a file path, a list of lists, text or a previously created nVenn object (see Details).
`plot`	If true (default), the resulting diagram is plotted. If false, only the object is returned.
`outFile`	If it contains a valid file path and `plot` is also true, the svg code of the plot will be saved in that path.
`systemShow`	If true, and `plot` is true, the function will attempt to open the resulting svg figure in the default editor. Defaults to false.
`verbose`	If true, shows messages as the nVenn plot is created.
`maxlevel`	If higher than zero, the simulation uses an exhaustive algorithm for the minimization steps. The number will represent the depth of the search. With a dept of one, every exchange of two regions will be explored. With a depth of two, every exchange of four regions will be explored. The computing resources necessary for this exploration increase extremely fast with the depth, so users are advised to use `estimateExhaustiveRunTime()` to determine whether it is feasible to use this procedure. The process will be unresponsive during the simulation and no messages will be shown, regardless of `verbose`.

byCol — If the input is a text, this parameter indicates whether each set is a column (1) or a row (2). Defaults to 0, which means that the package will try to guess which possibility makes more sense.

Details

A list of lists contains inner lists with a name, which will be the corresponding set name. A dataframe can be used in the same way.

The input can also be a text containing a table, possibly with missing values. If a text is provided, the package will try to guess if each set is encoded in columns or rows (use `byCol` to force) and which character separates fields (usually tab, space or comma). If the text describes a valid text file path, the contents of the file will be used.

Value

nVenn object. As a side effect, shows the nVenn plot. In case of error, returns null object.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
```

---

nVennR2 *nVennR2*

Description

An interface to nVenn2 to create n-dimensional, quasi-proportional Venn diagrams.

Details

The input for nVennR2 is a list of lists or a text table. The dataset `exampledf` contains an example of a list of lists, where each internal list has a name for the set and some set elements. The result will be a Venn diagram where each region is represented by a circle whose area is approximately proportional to the number of elements in that region. Each circle has two labels. The number in a larger font size represents the number of elements in the region. The smaller numbers in parentheses show which sets the region belongs to.

A diagram is created with `nVennDiagram()` and plotted and/or saved with `plotVenn()`. The appearance of the plot can be changed with `setVennOpts()` and `setVennColors()`. If a plot is saved, the resulting svg file can be loaded again with `readVennSVG()`.

Value

nVennObj. An nVenn object with the information needed to represent and edit the diagram, and to explore its regions.

Author(s)

Victor Quesada, Universidad de Oviedo (Spain).

Examples

```
library(nVennR2)
myv <- nVennDiagram(exampledf, verbose=FALSE)
myv <- setVennOpts(myv, opacity=0.2, fontSize=14)
```

---

plotVenn *Plot nVenn diagram*

Description

The nVenn diagram is generated in the plot window by default. If this is not possible, parameters outFile and systemShow may be used to export the SVG figure.

Usage

```
plotVenn(nVennObj, outFile = "", systemShow = FALSE)
```

Arguments

nVennObj	nVenn object.
outFile	Path to export the SVG figure. If empty, the figure is not exported.
systemShow	If true, and the system has a default SVG-editing program, opens the figure in the default editor.

Value

Nothing. If nVennObj is correct and the system supports it, the diagram is plotted in the plot window.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
plotVenn(myv)
```

readVennSVG *Creates nVenn plot*

Description

Creates nVenn plot

Usage

```
readVennSVG(svgFile, plot = TRUE, outFile = "", systemShow = FALSE)
```

Arguments

svgFile	File name of an SVG or HTML file created by nVenn.
plot	If true (default), the resulting diagram is plotted. If false, only the object is returned.
outFile	If it contains a valid file path and plot is also true, the svg code of the plot will be saved in that path.
systemShow	If true, and plot is true, the function will attempt to open the resulting svg figure in the default editor. Defaults to false.

Details

In principle, this function should work with any SVG or HTML file created by nVenn, with either nVennR2, a web interface or nVennPy.

Value

nVenn object.

Examples

```
if (file.exists('example.svg')){
  readVennSVG('example.svg')
}
```

rotateVenn *Rotate a diagram*

Description

Rotate a diagram

Usage

```
rotateVenn(nVennObj, angle, plot = TRUE)
```

Arguments

nVennObj	nVennR2 object generated with nVennDiagram().
angle	Angle of rotation in degrees. Positive values rotate counterclockwise.
plot	If true (default), prints the diagram after the rotation.

Value

nVenn object.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
myv <- rotateVenn(myv, 45)
```

setVennColor *Change a set color*

Description

Change a set color

Usage

```
setVennColor(nVennObj, setName, color, plot = TRUE)
```

Arguments

nVennObj	nVenn object created with nVennDiagram().
setName	Name of the set.
color	New color. It is important to notice that this parameter must be a valid color in SVG format. The value is not checked, and therefore an incorrect value may break the plot in plotVenn() or lead to unexpected results.
plot	If true (default), plots the diagram after setting the color.

Value

nVenn object with the change in color for the set.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
myv <- setVennColor(myv, "Set2", "black")
myv <- setVennColor(myv, "Set1", "#ffff00")
```

---

setVennColors *Change set colors*

Description

Change set colors

Usage

```
setVennColors(nVennObj, colorList, plot = TRUE)
```

Arguments

nVennObj	nVenn object created with nVennDiagram().
colorList	Vector or list of colors for the sets (see Details).
plot	If true (default), plots the diagram after setting the colors.

Details

If a vector of svg-formatted colors is provided, they will be used in the same order. This is a good way to create and use a custom color palette. If a list is used, the function will call setVennColor() with the names in the list.

It is important to notice that each color must be a valid color in SVG format. The value is not checked, and therefore an incorrect value may break the plot in plotVenn() or lead to unexpected results.

Value

nVenn object with changed set colors.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
mypalette <- c("black", "#ffff00", "red")
myv <- setVennColors(myv, mypalette)
```

setVennOpts *Set graphical parameters of an nVenn object diagram.*

Description

The function returns an object whose parameters will change the way the diagram is plotted.

Usage

```
setVennOpts(
  nVennObj,
  opacity = 0.4,
  fontSize = 12,
  lineWidth = 1,
  palette = 0,
  showRegions = TRUE,
  showWeights = TRUE,
  plot = TRUE
)
```

Arguments

nVennObj	nVenn object generated with nVennDiagram().
opacity	Opacity of sets, between 0 (completely transparent) and 1 (completely opaque). Defaults to 0.4.
fontSize	Size of the font for the numbers of elements (weights) in each region. It also indirectly controls the font sizes of the region descriptions, if shown (showRegions=TRUE). Defaults to 10.
lineWidth	Width of the lines defining each set. The value will be rounded to the closest integer. Defaults to 1.
palette	Color palette to use to fill the sets. Integer from 0 to 3. Defaults to 0.
showRegions	If true (default), show region descriptions.
showWeights	If true (default), show number of elements in each region.
plot	If true (default), plots the diagram after the operation.

Value

nVennObj with graphical options set.

Examples

```
myv <- nVennDiagram(exampledf)
myv <- setVennOpts(myv, showRegions=FALSE, opacity=0.2, lineWidth=2)
```

`setVennPalette` *Change the color palette for a diagram*

Description

Change the color palette for a diagram

Usage

```
setVennPalette(nVennObj, palette = 0, plot = TRUE)
```

Arguments

`nVennObj`	nVenn object created with `nVennDiagram()`.
`palette`	Either 0 (default), 1, 2 or 3. Each number defines a color palette that will be applied to the results
`plot`	If true (default), plots the diagram after applying the palette.

Details

The color palette for a diagram can also be set with `setVennOpts()`, but with a small difference. If a set color has been set with `setVennColor()`, `setVennPalette()` will override and delete that setting, while `setVennOpts()` will not. This way, one can have a color defined for a set and change the colors of the rest of sets with `setVennOpts()` or reset every color to the pre-defined palette with `setVennPalette()`.

Value

nVenn object with the palette set to the requested value.

See Also

`setVennOpts()` for an alternative way to change the color palette.

Examples

```
myv <- nVennDiagram(list(Set1=c("a", "b", "c"), Set2=c("a", "c", "d")), verbose=FALSE)
myv <- setVennPalette(myv, 2)
myv <- setVennPalette(myv, 3)
```

setVennSkin *Set nVenn diagram appearance*

Description

Set nVenn diagram appearance

Usage

```
setVennSkin(nVennObj, params, plot = TRUE)
```

Arguments

nVennObj	nVenn object generated with nVennDiagram().
params	List of graphical options, as in setVennOpts(), plus two special parameters: palette to set a color palette as in setVennPalette() and colors to change set colors as in setVennColors().
plot	If true (default), plots the diagram after the operation.

Details

Running this function is equivalent to running setVennOpts() repeatedly. The advantage is that setVennSkin() can set multiple parameters at the same time. The same params list can be used on multiple nVenn objects as a theme.

The only valid parameters for params are those in setVennOpts(), palette and colors.

Value

nVenn object with skin applied

Examples

```
theme <- list(opacity=0.2, lineWidth=2, fontSize=14, showRegions=FALSE,
              palette=2, colors=c("black"))
myv <- nVennDiagram(exampledf)
myv <- setVennSkin(myv, theme)
```

Index

∗ **datasets**
exampledf, 3

estimateExhaustiveRunTime, 2
estimateExhaustiveRunTime(), *6*
exampledf, 3

getVennRegion, 3
getVennSetNames, 4
getVennSvg, 5

listVennRegions, 5

nVennDiagram, 6
nVennDiagram(), *2–5*, *7*, *10–14*
nVennR2, 7

plotVenn, 8
plotVenn(), *5*, *7*, *10, 11*

readVennSVG, 9
readVennSVG(), *7*
rotateVenn, 10

setVennColor, 10
setVennColor(), *11*, *13*
setVennColors, 11
setVennColors(), *7*, *14*
setVennOpts, 12
setVennOpts(), *7*, *13, 14*
setVennPalette, 13
setVennPalette(), *13, 14*
setVennSkin, 14
setVennSkin(), *14*