Package 'nonLinearDotPlot'

May 9, 2026

Title Non Linear Dot Plots

Version 0.5.0

Description Non linear dot plots are diagrams that allow dots of varying size to be constructed, so that columns with a large number of samples are reduced in height. Implementation of algorithm described in: Nils Rodrigues and Daniel Weiskopf, ``Nonlinear Dot Plots", IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 616-625, 2018. <doi:10.1109/TVCG.2017.2744018>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Sören Döring [aut, cre] (ORCID:
<https://orcid.org/0000-0002-7229-6133>),
Manuel Kubica [aut],
Ines Fourati [aut]

Maintainer Sören Döring <doering.soeren@gmx.de>

Repository CRAN

Date/Publication 2023-05-19 17:30:02 UTC

Contents

dotscaling.linear 2
dotscaling.log 2
dotscaling.root 3
nonLinearDotPlot 3

Index **8**

dotscaling.linear *Predefined function to use as* dotscaling *in the* nonLinearDotPlot *function.*

Description

dotscaling(c) = 1 is a linear function

Usage

```
dotscaling.linear()
```

Value

Function that always returns 1.

References

N. Rodrigues and D. Weiskopf, "Nonlinear Dot Plots", IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 616-625, 2018. Available: doi:10.1109/TVCG.2017.2744018

dotscaling.log *Predefined function to use as* dotscaling *in the* nonLinearDotPlot *function.*

Description

dotscaling(c) = (log(c + base - 1) / log(base)) / c is a logarithmic function

Usage

```
dotscaling.log(base = 2)
```

Arguments

base	value of the base of the logarithm Default value of base equals 2

Value

Function to calculate dot size with (log(c + base - 1) / log(base)) / c.

References

N. Rodrigues and D. Weiskopf, "Nonlinear Dot Plots", IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 616-625, 2018. Available: doi:10.1109/TVCG.2017.2744018

dotscaling.root *Predefined function to use as* dotscaling *in the* nonLinearDotPlot *function.*

Description

dotscaling(c) = 1 / (c**e) is a root function

Usage

```
dotscaling.root(e = 0.3)
```

Arguments

e	determines which root should be used Default value of e equals 0.3

Value

Function to calculate dot size with 1 / (c ** e).

References

N. Rodrigues and D. Weiskopf, "Nonlinear Dot Plots", IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 616-625, 2018. Available: doi:10.1109/TVCG.2017.2744018

nonLinearDotPlot *Creates a non linear dot plot.*

Description

Non linear dot plots are diagrams that allow dots of varying size to be constructed, so that columns with a large number of samples are reduced in height. An efficient two-way sweep algorithm is used to obtain a dense and symmetrical layout.

Usage

```
nonLinearDotPlot(
  data,
  ...,
  xAttribute = 1,
  colorAttribute = NULL,
  main = NULL,
  sub = NULL,
  xlab = NULL,
  dSingle = 1,
  circlePadding = 0.05,
```

```
  useBlur = FALSE,
  blurEdge = 1,
  blurGapDistance = 2,
  colors = "black",
  colorPositions = NULL,
  xlim = NULL,
  xTicks = NULL,
  xAxisMargin = 0.5,
  asp = NULL,
  asplim = 10,
  useDeviceAsp = FALSE,
  colorAttributeMap = NULL,
  dotscaling = dotscaling.root(0.3),
  mar = NULL
)
```

Arguments

`data`	Input data frame.
`...`	Extra arguments are put into the R plot function. Using these may result in bad plots.
`xAttribute`	The name or the index of the header to use as X axis of the plot.
`colorAttribute`	The name or index of the header to use as reference for the colors of the plot.
`main`	Title of the plot.
`sub`	Subtitle for the plot.
`xlab`	Title for the x axis. If set to NULL, the name of the column is used.
`dSingle`	Constant start diameter used by the two-way sweep algorithm that facilitates an overall scaling of all dots. (see reference)
`circlePadding`	Value between 0 and 1 that scales the dots of the plot by the given percentage.
`useBlur`	Applies a simulated vertical Blur to all dots.
`blurEdge`	The number of dots at the top and bottom of every column in the plot that will be left unchanged when blur is activated.
`blurGapDistance`	Specifies how far away a column must be from the surrounding columns so that the blur will not be effective on it.
`colors`	Vector of the colors that will be used to color the dots. Colors can be specified either by name (e.g: "red") or as a hexadecimal RGB triplet (e.g: "#FFCC00"). See more.
`colorPositions`	Vector of values between 0 and 1 that defines the relative positions of each color used in the `colors` vector on the x axis of the plot. If `colorPositions` is equal NULL, the colors used in `colors` will be equidistant. The length of `colors` and `colorPositions` must be equal.
`xlim`	Vector (xMin,xMax) specifies the limits for the x axis, where xMin is the smallest value and xMax is the biggest value of the x axis.

xTicks	Vector of the values at which tick-marks are to be drawn. if set to NULL, tick-mark locations will be computed and set automatically.
xAxisMargin	Specifies the margin between the x Axis and the plot.
asp	Forces the plot to a specific aspect ratio. Set to NULL to disable it.
asplim	The number of iterations allowed for the aspect ratio approximation. The larger the value is, the more time is needed for the calculation.
useDeviceAsp	If it is set to TRUE, plot generation uses the aspect ratio of the available graphical output area. This will override the asp parameter.
colorAttributeMap	Function to map the color values of the plot. If set to NULL, the color values will be sorted by their default order and converted to numbers if not numeric.
dotscaling	Function responsible for the calculation of the circle diameter depending on the number of data values of the corresponding column and dSingle. These premade functions can be applied: dotscaling.root(e), dotscaling.linear(), and dotscaling.log(base). A custom scaling function can be used as well. For more details, please refer to the chapter 4.3 of the referenced paper doi:10.1109/TVCG.2017.2744018.
mar	Vector with margins around the plot in inches: (bottom, left, top, right). Set to NULL to use the standard margin of the R plot function.

Value

No return value. Plots directly to active device.

References

N. Rodrigues and D. Weiskopf, "Nonlinear Dot Plots", IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 616-625, 2018. Available: doi:10.1109/TVCG.2017.2744018

Examples

```
library(nonLinearDotPlot)

# creating data.frame with two columns
x <- c(0, 0, 1, 1, 2,
       2, 2, 2, 2, 3,
       3, 3, 3, 3, 4,
       4, 4, 5, 5, 5)
letter <- c("a", "b", "c", "c", "b",
            "b", "a", "b", "c", "a",
            "a", "a", "c", "a", "c",
            "a", "b", "c", "c", "a")
data <-  data.frame(x, letter)

# small margin around all plots
margin <- c(2.5,0.5,0.5,0.5)
# set the x axis limits for all plots
limits <- c(-0.5, 5.5)
```

```
# Simple plot with no axis description
nonLinearDotPlot(data = data, mar=margin,
                 xlim = limits, xlab ="",
                 xAttribute = "x")

# Dots are smaller
nonLinearDotPlot(data = data, mar=margin,
                 xlim = limits, xlab ="",
                 xAttribute = "x",
                 dSingle = 0.5)

# Plot adapts aspect ratio to device size
nonLinearDotPlot(data = data, mar=margin,
                 xlim = limits, xlab ="",
                 xAttribute = "x",
                 useDeviceAsp = TRUE)

# Logarithmic scaling function
nonLinearDotPlot(data = data, mar=margin,
                 xlim = limits, xlab ="",
                 xAttribute = "x",
                 dotscaling=dotscaling.log(3),
                 useDeviceAsp = TRUE)

# Linear scaling function
nonLinearDotPlot(data = data, mar=margin,
                 xlim = limits, xlab ="",
                 xAttribute = "x",
                 dotscaling=dotscaling.linear(),
                 useDeviceAsp = TRUE)

# Uniformly Changed color
nonLinearDotPlot(data = data, mar=margin,
                 xlim = limits, xlab ="",
                 xAttribute = "x",
                 colors = "Red")

# Simple coloring of dots depending on their table entry of "letter" in the data
colors <- c("#a6cee3", "#1f78b4", "#7570b3")

nonLinearDotPlot(data = data, mar=margin,
                 xlim = limits, xlab ="",
                 xAttribute = "x",
                 colorAttribute = "letter",
                 colors = colors)

# Change order of the stacked colors
colors <- c("#a6cee3", "#1f78b4", "#7570b3")
colorPositions <- c(1, 1/2, 0)

nonLinearDotPlot(data = data, mar=margin,
                 xlim = limits, xlab ="",
                 xAttribute = "x",
```

```
                  colorAttribute = "letter",
                  colors = colors,
                  colorPositions = colorPositions)

# Map specific color to each value in "letter"
colors <- c("#a6cee3", "#1f78b4", "#7570b3")
colorAttributeMapDict <- c('a' = 1, 'b' = 1/2, 'c' = 0)

colorAttributeMap <- function(names){
  return(colorAttributeMapDict[names])
}

nonLinearDotPlot(data = data, mar=margin,
                  xlim = limits, xlab ="",
                  xAttribute = "x",
                  colorAttribute = "letter",
                  colors = colors,
                  colorAttributeMap = colorAttributeMap)

# Use simulated blur
nonLinearDotPlot(data = data, mar=margin,
                  xlim = limits, xlab ="",
                  xAttribute = "x",
                  useBlur = TRUE,
                  blurEdge = 1,
                  blurGapDistance = 2)
```

Index

dotscaling.linear, 2
dotscaling.log, 2
dotscaling.root, 3

nonLinearDotPlot, 3