Package 'makeFlow'

July 22, 2025

Type Package

Title Visualizing Sequential Classifications

Version 1.0.2

Date 2016-08-22

Author Alex J. Krebs

Maintainer Alex J. Krebs <Krebs.AlexJ@gmail.com>

Description A user-friendly tool for visualizing categorical or group movement.

License GPL (>= 2)

Imports dplyr, RColorBrewer

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-08-24 18:02:38

Contents

`makeFlow-package` *Visualizing Sequential Classifications*

Description

A user-friendly tool for visualizing categorical or group movement.

Details

The DESCRIPTION file:

Package:	makeFlow
Type:	Package
Title:	Visualizing Sequential Classifications
Version:	1.0.2
Date:	2016-08-22
Author:	Alex J. Krebs
Maintainer:	Alex J. Krebs <Krebs.AlexJ@gmail.com>
Description:	A user-friendly tool for visualizing categorical or group movement.
License:	GPL (>= 2)
Imports:	dplyr, RColorBrewer
Copyright:	Copyright Notice GENERAL DISCLAIMER This program is free software; you can redistribute it and/or m
RoxygenNote:	5.0.1

Index of help topics:

```
FlowSummaries           FlowSummaries()
GateSummaries           GateSummaries()
addAlpha                addAlpha()
colorCount              colorCount()
makeFlow                makeFlow()
makeFlow-package        Visualizing Sequential Classifications
shelters                shelters
```

Users should ensure all classFields (columns) are explicitly defined in the same dataset. colorCount(), FlowSummaries(), GateSummaries(), and makeFlow() can all operate with the same two basic inputs: data and classFields. Graphical parameters can be defined with additional makeFlow() arguments.

Author(s)

Alex J. Krebs

Maintainer: Alex J. Krebs <Krebs.AlexJ@gmail.com>

Examples

```
##  Data:
##  carData <- mtcars
##  carData$car <- "All Cars"
##  carData$speedclass <- ifelse(carData$qsec < 15, "Fast",
##                          ifelse(carData$qsec < 18, "Mid-Speed","Slow"))
## carData$speedclass <- factor(x = carData$speedclass, levels = c("Slow","Mid-Speed","Fast"))
##
##  Create Diagram:
##  makeFlow(data = carData, classFields = c("car","cyl","speedclass"),
##    gateWidth = 20, minVerticalBtwnGates = .15, distanceBtwnGates = 70,
##    fieldLabels = c("", "Cylinders","Speed"), gateBorder = "black")

##  Generate underlying tables using GateSummaries() and FlowSummaries()
```

---

addAlpha *addAlpha()*

Description

Adds a specified opacity (between 0 and 1) to any color(s) listed.

Usage

```
addAlpha(col, alpha = 1)
```

Arguments

col	A vector of one or many colors.
alpha	A value between 0 and 1. 0 indicates complete transparency. 1 indicates complete opacity.

Value

Returns the Hexadecimal representation of the modified color(s).

Examples

```
## The function is currently defined as
function (col, alpha = 1)
{
    if (missing(col))
        stop("Please provide a vector of colors.")
    apply(sapply(col, col2rgb)/255, 2, function(x) {
        rgb(x[1], x[2], x[3], alpha = alpha)
    })
  }
```

`colorCount` *colorCount()*

Description

Returns an integer representing the number of unique categories from all specified fields. This value should serve as a guide for users specifying colors in the makeFlow() function.

Usage

```
colorCount(data, classFields)
```

Arguments

`data`	An object of class data.frame in which all specified classFields (or column names) can be found.
`classFields`	A vector of the column names intended to be represented in a makeFlow() diagram. Each element must be a string.

Note

Relies on Hadley Wickham's dplyr package.

Author(s)

Alex J. Krebs

FlowSummaries *FlowSummaries()*

Description

Outputs a list of objects of class tbl_df containing values presented in a makeFlow() diagram utilizing the same data and classFields arguments.

Usage

```
FlowSummaries(data, classFields)
```

Arguments

data	An object of class data.frame in which all specified classFields (column names) can be found.
classFields	A vector of the column names intended to be represented in the makeFlow() diagram. Each element must be a string.

Details

Outputs tables with standard naming convention of Flow_Summary_x. For example, Flow_Summary_1 provides the counts and frequencies of each category (gate) from the first column specified in classFields moving to each of the categories (gates) in the second column listed in classFields.

Note

Relies on Hadley Wickham's dplyr package.

Author(s)

Alex J. Krebs

Examples

```
## myFlows <- FlowSummaries(data= shelters, classFields= c("loc","Jan","Feb","Mar"))
## Flow_Summary_2 will show the counts and percentages from
## "Jan" categories that move to the categories in "Feb"
```

GateSummaries *GateSummaries()*

Description

Outputs a list of objects of class tbl_df containing the values in a makeFlow() diagram utilizing the same data and classFields arguments.

Usage

```
GateSummaries(data, classFields)
```

Arguments

`data`	An object of class data.frame in which all specified classFields (column names) can be found.
`classFields`	A vector of the column names intended to be represented in a makeFlow() diagram. Each element must be a string.

Details

Outputs tables with standard naming convention of Gate_Summary_x. For example, Gate_Summary_1 provides the counts and frequencies of each category within the first column specified in the classFields argument.

Note

Relies on Hadley Wickham's dplyr package.

Author(s)

Alex J. Krebs

Examples

```
## myGates <- GateSummaries(data = shelters, classFields = c("loc","Jan","Feb","Mar"))
## Gate_Summary_2 will show the count and percentage of observations
##  within each category of the column "Jan"
```

makeFlow *makeFlow()*

Description

Uses the selected dataset and specified order of columns to produce a left-to-right flow diagram. This function assumes the use of a single dataset and categorical variables resembling observations' movement from one state to another, such that at every state, every observation can be mapped. Colors bridging states assume the color of the gate from which they originate.

Usage

```
makeFlow(data, classFields, rotate = F, gatecolors = NA, minVerticalBtwnGates = 0.1,
connectingAlpha = 0.5, bg = "white", plotTitle = "", titleAdj = 0.5, txtColor = "black",
distanceBtwnGates = 50, gateWidth = 7, gateBorder = NA, labels = T, fieldLabels = NA,
showPercentages = T, showConnectPercentages = F, percentTextColor = "black",
showCounts = T, countTextColor = "black")
```

Arguments

data	An object of class data.frame in which all specified classFields (column names) can be found.
classFields	A vector of the column names intended to be represented in a makeFlow() diagram. Each element in this vector must be a string.
rotate	Set to TRUE to rotate all text. This is a work-around to allow for a vertical (top-down) flow. Manual manipulation after exporting a diagram is needed to adjust the orientation of the output.
gatecolors	A vector of colors to apply to gates. For greater control, the alphabetized order of categories from all selected classFields will match the corresponding color in this list. If the length of this vector is shorter than the number of unique categories in the full diagram, a predefined palette "Set3" is substituted in its place, and a warning message will alert the user of how many colors are required for manual input. If more colors are supplied than necessary, only the required number will be taken from the beginning of the vector. (default is NA, which will assign colors from palette "Set3")
minVerticalBtwnGates	A single value (likely between 0 and 1) used to determine the minumum plotted gap between categories (gates) within the same classField. For reference, the default plot height is 1. (default is 0.1)
connectingAlpha	A single decimal value [0,1] setting the opacity of the "flows" connecting gates. (default is 0.5)
bg	A single color from colors() specifying the diagram's background color. (default is "white")
plotTitle	The title of the diagram. (default is "")

`titleAdj` A single value [0,1] adjusting the title's horizontal placement. 0 implies left-align; 1 impllies right-align. (default is 0.5)

`txtColor` A single color from colors() specifying the text color in the diagram. (default is "black")

`distanceBtwnGates` A single numeric value representing the horizontal distance between classFields. (default is 50)

`gateWidth` A single numeric value representing the horizontal width of gates. (default is 7)

`gateBorder` A single color from colors() specifying the color of all gates' borders. (default is NA)

`labels` Set to FALSE to not see the category labels above each gate.

`fieldLabels` A character vector of names to identify each classField in the diagram. These labels will be placed below each set of gates. If labels is FALSE, fieldLabels will not be displayed. (default is NA)

`showPercentages` Set to FALSE to hide the percentage values within gates.

`showConnectPercentages` Set to FALSE to hide the percentage values within flows.

`percentTextColor` A single color from colors() for the gate percentage and flow percentage texts. (default is "black")

`showCounts` Set to FALSE to hide the counts of observations within each gate.

`countTextColor` A single color from colors() for the gate count text. (default is "black")

Note

Relies on Hadley Wickham's dplyr package to generate summaries. Gates within each field are ordered alphabetically, numerically, or (if applicable) in the order of a factor's levels.

Author(s)

Alex J. Krebs

Examples

```
## makeFlow(data = shelters, classFields = c("loc","Jan","Feb","Mar"),
##          fieldLabels = c("","Jan","Feb","Mar"), gateWidth = 20)
```

shelters *shelters*

Description

Pseudodata depicting the outcomes of 10,000 animals in shelters over three months generated to demonstrate the makeFlow() diagram. Aside: consider adopting your next pet :)

Usage

```
data("shelters")
```

Format

A data frame with 10000 observations on the following 6 variables.

`loc` a factor with levels `shelter`

`name` a character vector

`id` a numeric vector

`Jan` a factor with levels `Remaining` `Adopted` `Transfered` `Euthanized`

`Feb` a factor with levels `Remaining` `Adopted` `Transfered` `Euthanized`

`Mar` a factor with levels `Remaining` `Adopted` `Transfered` `Euthanized`

Index