Package ‘Radviz’

February 4, 2020

Type Package
Title Project Multidimensional Data in 2D Space
Version 0.8.2
Depends R (>= 3.0)
Imports ggplot2, dplyr, rlang, stats, utils
Suggests knitr, rmarkdown, bodenmiller, tidyr
Description An implementation of the radviz projection in R. It enables the visualization of multidimensional data while maintaining the relation to the original dimensions. This package provides functions to create and plot radviz projections, and a number of summary plots that enable comparison and analysis. For reference see Ankerst *et al.* (1996) (<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.68.1811>) for original implementation, see Di Caro *et al* 2012 (<http://link.springer.com/chapter/10.1007/978-3-642-13672-6_13>) for the original method for dimensional anchor arrangements.
License CC BY-NC-SA 4.0
URL http://github.com/yannabraham/Radviz
BugReports http://github.com/yannabraham/Radviz/issues
RoxygenNote 7.0.1
VignetteBuilder knitr
Encoding UTF-8
NeedsCompilation no
Author Yann Abraham [aut, cre]
Maintainer Yann Abraham <yann.abraham@gmail.com>
Repository CRAN
Date/Publication 2020-02-04 09:30:03 UTC
bubbleRadviz

A Plotting Function for the Radviz Object

Description

Plots the Dimensional Anchors and projected data points in a 2D space.

Usage

```r
bubbleRadviz(
  x,
  main = NULL,
  group = NULL,
  color = NULL,
  size = c(3, 16),
  label.color = NULL,
  label.size = NULL,
  bubble.color,
  bubble.fg,
  bubble.size,
  scale,
  decreasing,
  add
)
```
**bubbleRadviz**

**Arguments**

- **x**
  A radviz object as produced by do.radviz
- **main**
  [Optional] a title to the graph, displayed on top
- **group**
  The name of the grouping variable used to aggregate the data
- **color**
  [Optional] the name of the variable used to color the points
- **size**
  The size range for the plot
- **label.color**
  The color of springs for visualization
- **label.size**
  The size of labels
- **bubble.color**
  Deprecated, use `geom_point` instead
- **bubble.fg**
  Deprecated, use `geom_point` instead
- **bubble.size**
  Deprecated, use `geom_point` instead
- **scale**
  Deprecated, use `geom_point` instead
- **decreasing**
  Deprecated, use `geom_point` instead
- **add**
  Deprecated, use `geom_point` instead

**Details**

This function allows for the projection of clusters in Radviz (for example results of the SPADE algorithm), where the cluster size is derived from the number of events that fall into a specific cluster. If color is not specified the grouping variable is used.

**Value**

The internal ggplot2 object plus added layers, allowing for extra geoms to be added.

**Author(s)**

Yann Abraham

**Examples**

```r
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
bubbleRadviz(rv, group='Species')
```
contour.radviz

Plots the Dimensional Anchors and density lines for projected data points in a 2D space.

Description

Plots the Dimensional Anchors and density lines for projected data points in a 2D space.

Usage

## S3 method for class 'radviz'
contour(
  x,
  ..., 
  main = NULL,
  color = NULL,
  size = 0.5,
  label.color = NULL,
  label.size = NULL,
  contour.color,
  contour.size,
  point.color,
  point.shape,
  point.size,
  n,
  drawlabels,
  drawpoints,
  add
)

Arguments

- **x**: a radviz object as produced by do.radviz
- **...**: further arguments to be passed to or from other methods (not implemented)
- **main**: [Optional] a title to the graph, displayed on top
- **color**: the variable in the Radviz projection used to color the contours
- **size**: The thickness of contour lines
- **label.color**: the color of springs for visualization
- **label.size**: the size of labels
- **contour.color**: deprecated, see `geom_density2d` instead
- **contour.size**: deprecated, see `geom_density2d` instead
- **point.color**: deprecated, see `geom_density2d` instead
- **point.shape**: deprecated, see `geom_density2d` instead
- **point.size**: deprecated, see `geom_density2d` instead
cosine

Description
Given a dataset, compute the cosine similarity between to columns for use in optimization of Dimensional Anchors

Usage
cosine(mat)

Arguments
mat A matrix or data.frame

Details
implementation by David Ruau (see https://gist.github.com/bobthecat/2903031 for details)

Value
A symmetrical matrix with as many rows as there are columns in input
do.L  

Perform L-Normalization on a Vector

Description
Standardizes all values in a vector to the unit vector ([0,1]) using local min and max.

Usage
`do.L(v, fun = range, na.rm = T)`

Arguments
- `v`: a vector of values
- `fun`: a function that will return the minimum and maximum values to use to scale `v`; defaults to `range`
- `na.rm`: Logical: should NA be removed? defaults to TRUE

Details
This is an alternative to performing a L normalization over the full matrix. if the minimum and the maximum values returned after applying `fun` are the same, `do.L` will return 0.

Value
A vector of values of the same length as `x`, scaled to the unit vector.

Author(s)
Yann Abraham
do.optim

**Examples**

```r
data(iris)
scaled <- apply(mat, 2, do.L)
summary(scaled) # all values are between [0,1]
scaled2 <- apply(mat, 2, do.L, fun=function(x) quantile(x, c(0.025, 0.975)))
summary(scaled2) # all values are between [0,1]
plot(scaled, scaled2,
col=rep(seq(1, ncol(scaled)), each=nrow(scaled)),
pch=16)
legend('topleft', legend=dimnames(scaled)[[2]], col=seq(1, ncol(scaled)), pch=16, bty='n')
```

---

**do.optim**  
*Optimize the Dimensional Anchors Position using a Genetic Algorithm*

**Description**

Allows to compute the best arrangement of Dimensional Anchors so that visualization efficiency is maximized.

**Usage**

```r
do.optim(
  springs,
  similarity,
  iter = 100,
  n = 1000,
  top = round(n * 0.1),
  lambda = 0.01,
  nlast = 5,
  optim = "in.da"
)
```

**Arguments**

- `springs`: A matrix of 2D dimensional anchor coordinates, as returned by `make.S`
- `similarity`: A similarity matrix measuring the correlation between Dimensional Anchors
- `iter`: The maximum number of iterations (defaults to 100)
- `n`: The number of permutations of Dimensional Anchors to be created at each generation
- `top`: The number of permutations to keep to create the next generation
- `lambda`: The threshold for the optimization process
- `nlast`: The number of generations to wait before lambda is applied
- `optim`: The optimization function (in or rv)
Details

The first generation is a random sampling of all Dimensional Anchors. For every generation afterwards, only the best solutions (as specified by top) are kept; the solutions are normalized around the unit circle (i.e., $c(1,2,3,4)$ is equivalent to $c(4,1,2,3)$ for Radviz projection) before the next generation is created. The next generation consists of

- all unique best solutions from the previous generation (after circular normalization)
- a permutation of all previous solutions.

Briefly, for every Dimensional Anchor position the previous generation is sampled to give a mixture of identical and slightly shifted (mutated) solutions. The algorithm will stop when the maximum number of iterations (as defined by \texttt{iter}) is reached, or when a number of generations (defined by \texttt{nlast}) as not improved over the best solution by more than a given threshold (specified by \texttt{lambda}).

Value

a list containing 3 sets of values:

- \texttt{perfs} the list of the best performances by generation
- \texttt{best} the best performing arrangement by generation
- \texttt{last} the top performing arrangements of the last generation

Author(s)

Yann Abraham

Examples

data(iris)
das <- c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width")
S <- make.S(das)
rv <- do.radviz(iris,S)
plot(rv,anchors.only=FALSE)
sim.mat <- cosine(iris[,das])
in.da(S,sim.mat) # the starting value
new <- do.optim(S,sim.mat,iter=10,n=100)
new.S <- make.S(get.optim(new))
new.rv <- do.radviz(iris,new.S)
plot(new.rv,anchors.only=FALSE)
do.radviz

Projects a Matrix or a Data Frame to a 2D space defined by Dimensional Anchors

Description

do.radviz will return a projection of a multidimensional dataset onto a 2D space defined by dimensional anchors that have been projected on the unit circle using \texttt{make.S}

Usage

do.radviz(
  x,
  springs,
  trans = do.L,
  label.color = "orangered4",
  label.size = NA
)

Arguments

\begin{itemize}
  \item \texttt{x} a data.frame or matrix to be projected, with column names matching row names in \texttt{springs}
  \item \texttt{springs} a matrix of 2D dimensional anchor coordinates, as returned by \texttt{make.S}
  \item \texttt{trans} a transformation to be applied to the data before projection
  \item \texttt{label.color} the color of springs for visualization
  \item \texttt{label.size} the size of labels
\end{itemize}

Details

The function expects that at least some of the column names in df will be matched by row names in \texttt{springs}

Value

a ggplot2 object of class radviz with a single geom_text layer corresponding to \texttt{springs}. the data slot of the ggplot2 corresponds to the input parameter \texttt{x} with the following extra columns

\begin{itemize}
  \item \texttt{rx} and \texttt{ry} the X and Y coordinates of the radviz projection of \texttt{x} over \texttt{springs}
  \item \texttt{rvalid} an index of points corresponding to an invalid projection (any \texttt{rx} or \texttt{ry} is NA)
\end{itemize}

Author(s)

Yann Abraham
get.optim

Get the Result of the Optimization Operation

Description

Once the order of anchors has been optimized using do.optim this function can be used to recover the optimized anchors or any intermediate step.

Usage

get.optim(opt, n = NULL)

Arguments

opt    the result of the optimization operation performed by do.optim
n      the optimized order of anchors to return; defaults to NULL, which returns the best identified combination

Value

a character vector of the anchor names, ordered as in the n^th^ step of the optimization

Author(s)

Yann Abraham
Examples

data(iris)  
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')  
S <- make.S(das)  
sim.mat <- cosine(iris[,das])  
in.da(S,sim.mat) # the starting value  
new <- do.optim(S,sim.mat,iter=10,n=100)  
get.optim(new) # the optimal order  
get.optim(new,2) # the second step of the optimization

hexplot

A hexplot function for Radviz objects

Description

Plots the Dimensional Anchors and a hexplot-based density representation of projected data points in a 2D space.

Usage

hexplot(
  x,  
  main = NULL,  
  nbins = 30,  
  color = NULL,  
  label.color = NULL,  
  label.size = NULL,  
  mincnt,  
  style
)

Arguments

x a radviz object as produced by do.radviz  
main [Optional] a title to the graph, displayed on top  
nbins the number of equally spaced bins for the binning computation (see geom_hex for details)  
color if color is not NULL and corresponds to one of the channels in the hexcols slot of the Radviz object, cells will be colored using colors in the hexcols slot  
label.color the color of springs for visualization  
label.size the size of labels  
mincnt deprecated, see stat_summary_hex instead  
style deprecated, see stat_summary_hex instead
in.da

Optimization functions for Dimensional Anchors in Radviz

### Value

the internal ggplot2 object plus added layers, allowing for extra geoms to be added

### Author(s)

Yann Abraham

### Examples

```r
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
hexplot(rv,color='Sepal.Length')
```

### Description

Visual efficiency of Radviz plots depends heavily on the correct arrangement of Dimensional Anchors. These functions implement the optimization strategies described in Di Caro et al 2012

### Usage

```r
in.da(springs, similarity)
rv.da(springs, similarity)
```

### Arguments

- `springs` A matrix of 2D dimensional anchor coordinates, as returned by `make.S`
- `similarity` A similarity matrix measuring the correlation between Dimensional Anchors

### Details

Following the recommendation of Di Caro *et al.* we used a cosine function to calculate the similarity between Dimensional Anchors (see `cosine` for details). The `in.da` function implements the independent similarity measure, where the value increases as the Radviz projection improves. The `rv.da` function implements the radviz-dependent similarity measure, where the value decreases as the Radviz projection improves.

### Value

A measure of the efficiency of the Radviz projection of the similarity matrix onto a set of springs
is.radviz

Author(s)
Yann Abraham

Examples

```r
data(iris)
S <- make.S(das)
mat <- iris[, das]
sim.mat <- cosine(mat)
in.da(S, sim.mat)
rv.da(S, sim.mat)
```

---

is.radviz

Test if the object is a Radviz object

Description

The function will return TRUE if the object is a Radviz object

Usage

```r
is.radviz(x)
```

Arguments

- `x` an object of class Radviz, as returned by `do.radviz`

Author(s)
Yann Abraham

Examples

```r
data(iris)
S <- make.S(das)
rv <- do.radviz(iris, S)
```

```r
is.radviz(rv) # should be true
```
is.valid

Identify the valid projections from a Radviz object

Description

The function will return a vector as long as the data in x where points that could not be projected are TRUE.

Usage

is.valid(x)

Arguments

x

an object of class Radviz, as returned by do.radviz

Author(s)

Yann Abraham

Examples

data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
iris0 <- rbind(iris,c(rep(0,length(das)),NA))
S <- make.S(das)
rv0 <- do.radviz(iris0,S)

sum(!is.valid(rv0)) # should be 1

# to find which points where invalid in the data
which(!is.valid(rv0))

# to review the original data points
rv1 <- subset(rv0,is.valid(rv0))

summary(rv1)

make.S

Define Dimensional Anchors around the Unit Circle

Description

make.S will return [x,y] coordinates for n dimensional anchors equally spaced around the unit circle.

Usage

make.S(x)
Arguments

- `x`: a vector of dimensional anchors, or a list of dimensional anchors for Class Discrimination Layout, or the number of anchors to put on the circle

Details

If `x` is a vector or a list, values will be used to set the row names of the matrix.

Value

A matrix with 2 columns (x and y coordinates of dimensional anchors) and 1 line per dimensional anchor (so called springs). If `x` is a vector, the row names of the matrix will be set to the syntactically correct version of values in the vector (through a call to `make.names`). Please note that some functions expect to match column names of data to row names of the spring matrix.

Author(s)

Yann Abraham

Examples

```r
data(iris)
das <- c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width")
make.S(length(das)) # without row names
make.S(das) # with row names
make.S(list(c("Sepal.Length","Sepal.Width"),c("Petal.Length","Petal.Width")))
```

Description

Plots the Dimensional Anchors and projected data points in a 2D space.

Usage

```r
## S3 method for class 'radviz'
plot(
  x,
  main = NULL,
  anchors.only = TRUE,
  label.color = NULL,
  label.size = NULL,
  point.color,
  point.shape,
  point.size,
  add,
```
Arguments

- **x**: a radviz object as produced by `do.radviz`
- **main**: [Optional] a title to the graph, displayed on top
- **anchors.only**: by default only plot the anchors so that other methods can easily be chained
- **label.color**: the color of springs for visualization
- **label.size**: the size of labels
- **point.color**: deprecated, use `geom_point` instead
- **point.shape**: deprecated, use `geom_point` instead
- **point.size**: deprecated, use `geom_point` instead
- **add**: deprecated, use `geom_point` instead
- **...**: further arguments to be passed to or from other methods (not implemented)

Details

by default the plot function only shows the anchors. Extra geoms are required to display the data

Value

the internal ggplot2 object, allowing for extra geoms to be added

Author(s)

Yann Abraham

Examples

data(iris)
das <- c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width")
S <- make.S(das)
rv <- do.radviz(iris,S)
plot(rv)
plot(rv,anchors.only=FALSE)

library(ggplot2)
## should look the same as before
plot(rv)+geom_point()
plot(rv)+geom_point(aes(color=Species))
Radviz Projection of Multidimensional Data

Description

Radviz uses Dimensional Anchors and the spring paradigm to project a multidimensional space in 2D. This allows for the quick visualization of large and complex datasets.

Examples

```r
data(iris)
das <- c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width")
S <- make.S(das)
rv <- do.radviz(iris,S)
plot(rv, anchors.only=FALSE)
```

recenter

Rotate Dimensional Anchors around the Unit Circle

Description

recenter will rotate the order of the dimensional anchors around the circle, to put a channel of reference to the top of the display.

Usage

```r
recenter(springs, newc)
```

Arguments

- `springs`: a spring object as created by `make.S`
- `newc`: a string specifying which dimensional anchor should be placed on top of the unit circle

Value

a spring object with rotated labels

Author(s)

Yann Abraham
### Examples

```r
data(iris)
iris.S <- make.S(das)
iris.S
recenter(iris.S, 'Petal.Length')
```

---

### smoothRadviz

**A smoothScatter function for Radviz objects**

#### Description

Plots the Dimensional Anchors and a smoothed color density representation of projected data points in a 2D space.

#### Usage

```r
smoothRadviz(
  x,
  main = NULL,
  color = "dodgerblue4",
  nbin = 200,
  label.color = NULL,
  label.size = NULL,
  smooth.color,
  max.dens,
  transformation,
  nrpoints,
  ncols,
  bandwidth
)
```

#### Arguments

- **x**: a radviz object as produced by `do.radviz`
- **main**: [Optional] a title to the graph, displayed on top
- **color**: the gradient will be generated from white to color
- **nbin**: the number of equally spaced grid points for the density estimation (see `geom_density_2d` for details)
- **label.color**: the color of springs for visualization
- **label.size**: the size of labels
- **smooth.color**: deprecated, see `stat_density2d` instead
- **max.dens**: deprecated, see `stat_density2d` instead
- **transformation**: deprecated, see `stat_density2d` instead
- **nrpoints**, **ncols**, **bandwidth**
Value

the internal ggplot2 object plus added layers, allowing for extra geoms to be added

Author(s)

Yann Abraham

Examples

data(iris)
S <- make.S(das)
rv <- do.radviz(iris, S)
smoothRadviz(rv)

---

subset.radviz

Subsetting a Radviz projection

Description

Subsetting a Radviz projection

Usage

## S3 method for class 'radviz'
subset(x, i, ...)

Arguments

x a radviz object
i A logical or indices vector of the same length as the original data used to create the Radviz object, that is used to subset each slots
... further arguments to be passed to or from other methods (not implemented)

Value

a new Radviz object containing only rows specified in i. Any density or hexbin analysis is dropped

Author(s)

Yann Abraham
Examples

data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
# subset rv
srv <- subset(rv,iris$Species=='setosa')
summary(srv)
sum(iris$Species=='setosa') # 50 objects in srv corresponding to setosa values

summary.radviz

Radviz Object Summary, head, print, dim and springs Methods

Description

Provides a summary for Radviz objects

Usage

## S3 method for class 'radviz'
summary(object, ..., n = 6)

## S3 method for class 'radviz'
head(x, n = 6, ...)

## S3 method for class 'radviz'
dim(x)

## S3 method for class 'radviz'
print(x, ...)

springs(x)

Arguments

object an object of class Radviz, as returned by do.radviz
...
... further arguments to be passed to or from other methods (not implemented)
n the number of lines from each slots in the Radviz object to display (defaults to 6)
x an object of class Radviz, as returned by do.radviz

Details

dim returns the number of points and the number of dimensions used for the projection
Author(s)

Yann Abraham

Examples

```r
data(iris)
S <- make.S(das)
rv <- do.radviz(iris, S)

summary(rv)
head(rv)
dim(rv)
print(rv)
```

Description

Text draws the strings given in the vector labels at the coordinates given by the radviz projection

Usage

```r
## S3 method for class 'radviz'
text(
x,
...,
main = NULL,
labels = NULL,
size = FALSE,
label.color,
label.size,
adj,
pos,
offset,
vfont,
cex,
col,
font,
add
)
```
theme_radviz

Arguments

- **x**
  - a radviz object as produced by do.radviz
- 
  - further arguments to be passed to or from other methods (not implemented)
- **main**
  - [Optional] a title to the graph, displayed on top if add is TRUE
- **labels**
  - the name of the variable used for labeling (see details)
- **size**
  - [Logical] if TRUE labels are sized after the number of points they correspond to
- **label.color**
  - deprecated, see do.radviz
- **label.size**
  - deprecated, see do.radviz
- **adj**
  - deprecated, see geom_text instead
- **pos**
  - deprecated, see geom_text instead
- **offset**
  - deprecated, see geom_text instead
- **vfont**
  - deprecated, see geom_text instead
- **cex**
  - deprecated, see geom_text instead
- **col**
  - deprecated, see geom_text instead
- **font**
  - deprecated, see geom_text instead
- **add**
  - deprecated, see geom_text instead

Author(s)

Yann Abraham

Examples

data(iris)
das <- c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width")
S <- make.S(das)
rv <- do.radviz(iris,S)
text(rv,labels="Species")

theme_radviz

Complete ggplot2 theme for Radviz projections

Description

A complete Radviz theme based on 'ggplot2::theme_light'

Usage

```
theme_radviz(
    base_size = 11,
    base_family = "",
    base_line_size = base_size/22,
    base_rect_size = base_size/22
)
```
**theme_radviz**

**Arguments**
- `base_size`  base font size
- `base_family`  base font family
- `base_line_size`  base size for line elements
- `base_rect_size`  base size for rect elements

**Details**
- On top of `ggplot2::theme_light` this theme removes axis title, text and ticks, as well as the reference grid. See [theme](https://www.ggplot2.org/) for details.

**Value**
- A complete `ggplot2` theme

**Author(s)**
- Yann Abraham

**Examples**
```r
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
plot(rv,main='Iris projection')
plot(rv,main='Iris projection')+theme_radviz(base_size=16)
```
Index

+Topic hplot
  contour.radviz, 4
+Topic multivariate
  contour.radviz, 4

bubbleRadviz, 2
contour.radviz, 4
cosine, 5, 12
dim.radviz(summary.radviz), 20
do.L, 6
do.optim, 7, 10
do.radviz, 9, 13, 14, 16, 18, 20, 22
gem_density2d, 4, 5
gem_density_2d, 18
gem_hex, 11
gem_point, 3, 16
gem_text, 22
get.optim, 10

head.radviz(summary.radviz), 20
hexplot, 11

in.da, 12
is.radviz, 13
is.valid, 14

make.names, 15
make.S, 7, 9, 12, 14, 17

plot.radviz, 15
print.radviz(summary.radviz), 20

Radviz, 17
range, 6
recenter, 17
rv.da(in.da), 12

smoothRadviz, 18

springs(summary.radviz), 20
stat_density2d, 18, 19
stat_summary_hex, 11
subset.radviz, 19
summary.radviz, 20
text.radviz, 21
theme, 23
theme_radviz, 22