Package 'Radviz'

December 8, 2016

Type Package

Title Project Multidimensional Data in 2D Space

Version 0.7.0

Depends R (>= 2.15)

Imports graphics, stats, utils, grid, gridBase, hexbin, MASS, KernSmooth

Suggests knitr, rmarkdown, bodenmiller, colorspace

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. (2010) <DOI: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 5.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Yann Abraham [aut, cre]

Maintainer Yann Abraham <yann.abraham@gmail.com>

Repository CRAN

Date/Publication 2016-12-08 15:29:53

R topics documented:

`bubbleRadviz` *A Plotting Function for the Radviz Object*

Description

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

Usage

```
bubbleRadviz(x, main = NULL, label.color = "orangered4", label.size = 1,
  bubble.color = "grey", bubble.fg = "white", bubble.size = 1,
  scale = 0.5, decreasing = NA, add = FALSE)
```

Arguments

`x`	a radviz object as produced by do.radviz
`main`	*Optional* a title to the graph, displayed on top
`label.color`	The color of the Dimensional Anchors (defaults to orangered4)
`label.size`	numeric character expansion factor for Dimensional Anchor labels; multiplied by `par("cex")` yields the final character size. NULL and NA are equivalent to 1.0

bubble.color	The color of the bubble, either a single color or a vector of colors (defaults to grey)
bubble.fg	The foreground color of the bubble, either a single color or a vector of colors (defaults to white)
bubble.size	the bubble size, either a single number of a vector of values (defaults to 1)
scale	A scaling factor that will be applied to bubble.size (see symbols for details)
decreasing	How the bubbles should be sorted: either NA for no sorting, TRUE or FALSE for sorting by decreasing or increasing bubble.size respecitvely, or a vector specifying the bubble order (see symbols for details)
add	Logical: if add is TRUE than only the projected points are plotted

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

Author(s)

Yann Abraham

Examples

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
species <- apply(iris[,c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')],
                 2,
                 tapply,iris$Species,median)
rv <- do.radviz(species,S)
bubbleRadviz(rv,
             bubble.color=c('red','green','blue')[seq(1,length(levels(iris$Species)))],
             bubble.size=table(iris$Species),
             decreasing=TRUE)
```

---

contour.radviz *Creates a contour plot, or add a contour plot to an existing plot for a Radviz Object*

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, label.color = "orangered4",
  label.size = 1, contour.color = par("fg"), contour.size = par("lwd"),
  point.color = "lightgrey", point.shape = ".", point.size = 1, add = F,
  drawlabels = FALSE, drawpoints = FALSE)
```

Arguments

x	a radviz object as produced by do.radviz
...	further arguments to be passed to or from other methods
main	[Optional] a title to the graph, displayed on top
label.color	The color of the Dimensional Anchors (defaults to orangered4)
label.size	numeric character expansion factor for Dimensional Anchor labels; multiplied by par("cex") yields the final character size. NULL and NA are equivalent to 1.0
contour.color	The color of contour lines (defaults to par('fg'))
contour.size	The thickness of contour lines (defaults to par('lwd'))
point.color	The point color (defaults to black)
point.shape	The point shape (defaults to '.')
point.size	the point size (defaults to 1)
add	Logical: if add is TRUE then only the contour lines are plotted
drawlabels	Logical. Contours are labelled if TRUE
drawpoints	Logical: if TRUE then the projected points are plotted

Details

The density lines will be calculated before plotting, if the Radviz object does not have one yet. The add allows plotting of contour lines over existing data, either the one used to generate the density or a different one (for context).

Value

Invisibly, the Radviz object that has been used as input; useful when do.density has not been called before so that results can be recovered

Author(s)

Yann Abraham

See Also

do.density for details about mapping projection to density

Examples

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
rv <- do.density(rv)
contour(rv,point.shape=1,point.color=c('red','green','blue')[as.integer(iris$Species)])
```

`cosine` *Compute the Cosine Similarity between the Columns of a Data Set*

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

Author(s)

Yann Abraham

David Ruau

Examples

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
mat <- iris[,das]
scaled <- apply(mat,2,do.L)
sim.mat <- cosine(scaled)
ncol(mat)
dim(sim.mat)
```

dim.radviz *Dimensions of a Radviz Object*

Description

Retrieves the dimensions of a Radviz projection

Usage

```
## S3 method for class 'radviz'
dim(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')
S <- make.S(das)
rv <- do.radviz(iris,S)

dim(rv)
nrow(rv) # for free!
```

do.density *Computes 2D density for contour plots of Radviz objects*

Description

Computes 2D density estimate of projected data for a Radivz object, using the kde2d function from the **MASS** package

Usage

```
do.density(x, n = 50)
```

Arguments

x	a radviz object as produced by do.radviz
n	Number of grid points in each direction. Can be scalar or a length-2 integer vector (see kde2d for details)

Details

Computes a 2D density estimate of Radviz projected data and stores the results in a `density` slot of the Radviz object. Invalid points, if any, will be excluded.

Value

the Radviz object with and extra slot `density` containing the 2D density estimates for use with `contour.radviz`

Author(s)

Yann Abraham

See Also

contour.radviz for plotting

Examples

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
rv <- do.density(rv)
contour(rv,point.shape=1,point.color=c('red','green','blue')[as.integer(iris$Species)])
```

---

do.hex *Computes 2D hexagonal bins from a Radviz projection*

Description

Computes 2D density using hexagon binning of projected data for a Radivz object, using the `hexbin` function from the **hexbin** package

Usage

```
do.hex(x, n = 30, channels = NULL, colramp = function(n)
  colorRampPalette(c("yellow", "grey", "blue"))(n), ncols = 8,
  use.quantile = F, fixed = NULL)
```

Arguments

`x`	a radviz object as produced by do.radviz
`n`	Number of bins in the grid (see hexbin for details)
`channels`	[optional] if channels is not `NULL`, the function will compute color scales for the specified channels

colramp	[optional] function accepting an integer n as an argument and returning n colors
ncols	[optional] the number of levels used to cut each channels
use.quantile	[optional] if channels is not NULL and use.quantile is TRUE, channels will be cut into ncols+1 quantiles otherwise a fixed sequence from 0 to 1 in ncols+1 steps will be used
fixed	[optional] if channels is not NULL and fixed is TRUE, channels will be cut into length(fixed)+1 steps using the values in fixed as breaking points

Details

The projected points will be binned into an hexagonal grid of size n. if channels is not NULL, for every channels the median intensity will be estimated over the complete grid and a color will be assigned from colramp using ncols, fixed and use.quantile; by default, channels will be split using a fixed sequence from 0 to 1 over ncols+1 levels unless use.quantile or fixed are set. Invalid points (if any) will not be used in computing the bins or the colors.

Value

the Radviz object with an extra slot hex containing the hexbin object; if channels is not NULL an extra hexcols will be present containing the color information for every channel

Author(s)

Yann Abraham

Dan Carr

See Also

hexplot for plotting, hexbin for original implementation

Examples

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
rv <- do.hex(rv,channels='Sepal.Length',ncols=4,use.quantile=TRUE)
summary(rv$hex)
```

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.

Value

A vector of values of the same lenght as x, scaled to the unit vector.

Author(s)

Yann Abraham

Examples

```
data(iris)
mat <- iris[,c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')]
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

```
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 permuations 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 (ie 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 `iter`) is reached, or when a number of generations (defined by `nlast`) as not improved over the best solution by more than a given threshold (specified by `lambda`).

Value

a list containing 3 sets of values:

- `perfs` the list of the best performances by generation
- `best` the best performing arrangement by generation
- `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)
scaled <- apply(iris[,das],2,do.L)
rv <- do.radviz(scaled,S)
plot(rv,main='Iris Columns',
     point.shape=1,
     point.color=c('red','green','blue')[as.integer(iris$Species)])
sim.mat <- cosine(scaled)
in.da(S,sim.mat) # the starting value
new <- do.optim(S,sim.mat,iter=10,n=100)
new.S <- make.S(get.optim(new))
new.rv <- do.radviz(scaled,new.S)
plot(new.rv,main='Optimized columns',
     point.shape=1,
     point.color=c('red','green','blue')[as.integer(iris$Species)])
```

`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 `make.S`

Usage

```
do.radviz(x, springs)
```

Arguments

`x`	a data.frame or matrix to be projected, with column names matching row names in springs
`springs`	A matrix of 2D dimensional anchor coordinates, as returned by `make.S`

Details

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

Value

An object of class radviz with the following slots:

- `data` the original data (`x`)
- `springs` the original `springs`
- `projected` the projection of `x` on `springs`, a matrix of 2D coordinates for every line in df
- `valid` a logical vector

Author(s)

Yann Abraham

Examples

```
# the first example generates a simple Radviz object
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
summary(rv)

# in case a point cannot be projected, a warning will be raise
iris0 <- rbind(iris,c(rep(0,length(das)),NA))
rv0 <- do.radviz(iris0,S)

# to find out how many points could not be projected:
with(rv0,sum(!valid))

# to find which points where invalid in the data
with(rv0,which(!valid))

# to review the original data points
with(rv0,subset(data,!valid))
```

`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)
scaled <- apply(iris[,das],2,do.L)
rv <- do.radviz(scaled,S)
plot(rv,main='Iris Columns',
     point.shape=1,
     point.color=c('red','green','blue')[as.integer(iris$Species)])
sim.mat <- cosine(scaled)
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
```

---

head.radviz *Radviz Object Summary, head, Print, Subset Methods*

Description

Provides helper function to deal with Radviz objects

Usage

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

Arguments

x	an object of class Radviz, as returned by do.radviz
n	the number of lines from each slots in the Radviz object to display (defaults to 6)
...	further arguments to be passed to or from other methods

Details

`head.radviz` shows the first `n` lines of the radviz `data` object.

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)

head(rv)
```

`hexplot` *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

```
hexplot(x, main = NULL, label.color = "orangered4", label.size = 1,
  mincnt = 0, color = NULL, style = "constant.col")
```

Arguments

`x`	a radviz object as produced by do.radviz
`main`	[Optional] a title to the graph, displayed on top
`label.color`	The color of the Dimensional Anchors (defaults to orangered4)
`label.size`	numeric character expansion factor for Dimensional Anchor labels multiplied by `par("cex")` yields the final character size. NULL and NA are equivalent to 1.0
`mincnt`	numeric; cells with counts smaller than mincnt are not shown (see grid.hexagons 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
`style`	character string specifying the type of plotting (see grid.hexagons for details)

Value

Plots the result of a `do.hex` function

Author(s)

Yann Abraham

Dan Carr

Nicholas Lewin-Koh

See Also

grid.hexagons and hexbin for original implementation

Examples

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
rv <- do.hex(rv,channels='Sepal.Length',ncols=4,use.quantile=TRUE)
hexplot(rv,color='Sepal.Length')
```

---

identify.radviz *Identify a Point in a Radviz Projection*

Description

Use this function to get the index of a point in a Radviz projection

Usage

```
## S3 method for class 'radviz'
identify(x, ..., n = 1)
```

Arguments

x	a radviz object as produced by do.radviz
...	further arguments to be passed to or from other methods
n	the number of points to identify, defaults to 1

Details

The function will use the row names of the data variable in the rv object for labeling the plot

Value

an integer vector containing the indices of the identified points, in the order they were identified.

Author(s)

Yann Abraham

See Also

identify

Examples

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
plot(rv,point.shape=1,point.color=c('red','green','blue')[as.integer(iris$Species)])
identify(rv)
```

---

in.da *Optimization functions for Dimensional Anchors in Radviz*

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

```
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 de cario 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

Author(s)

Yann Abraham

Examples

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
scaled <- apply(iris[,das],2,do.L)
sim.mat <- cosine(scaled)
in.da(S,sim.mat) # increases with better projections
rv.da(S,sim.mat) # decreases with better projections
```

---

is.radviz *Test if the object is a Radviz object*

Description

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

Usage

```
is.radviz(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')
S <- make.S(das)
rv <- do.radviz(iris,S)

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

make.S *Define Dimensional Anchors around the Unit Circle*

Description

make.S will return xy coordinates for n dimensional anchors equally spaced around the unit circle

Usage

```
make.S(x)
```

Arguments

x	a vector of dimensional anchors or the number of anchors to put on the circle

Details

If x is a vector, 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

```
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
```

plot.radviz *A Plotting Function for the Radviz Object*

Description

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

Usage

```
## S3 method for class 'radviz'
plot(x, main = NULL, label.color = "orangered4",
  label.size = 1, point.color = "black", point.shape = ".",
  point.size = 1, add = FALSE, anchors.only = FALSE, ...)
```

Arguments

x	a radviz object as produced by do.radviz
main	[Optional] a title to the graph, displayed on top
label.color	The color of the Dimensional Anchors (defaults to orangered4)
label.size	numeric character expansion factor for Dimensional Anchor labels; multiplied by par("cex") yields the final character size. NULL and NA are equivalent to 1.0
point.color	The point color (defaults to black)
point.shape	The point shape (defaults to '.')
point.size	The point size (defaults to 1)
add	Logical: if add is TRUE then only the projected points are plotted
anchors.only	plot only the anchors so that other plots can be freely overlaid
...	further arguments to be passed to or from other methods

Details

The add option allows plotting of additional data such as cluster centers onto an existing plot.

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,point.shape=1,point.color=c('red','green','blue')[as.integer(iris$Species)])
```

print.radviz *Print Radviz Object*

Description

Prints the first few lines of the data used to generate a Radviz projection

Usage

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

Arguments

x	an object of class Radviz, as returned by do.radviz
n	the number of lines from each slots in the Radviz object to display (defaults to 6)
...	further arguments to be passed to or from other methods

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)

print(rv)
```

Radviz *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

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
S <- make.S(das)
rv <- do.radviz(iris,S)
plot(rv,point.shape=1,point.color=c('red','green','blue')[as.integer(iris$Species)])
```

---

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

```
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

```
data(iris)
das <- c('Sepal.Length','Sepal.Width','Petal.Length','Petal.Width')
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

```
smoothRadviz(x, main = NULL, label.color = "orangered4", label.size = 1,
  smooth.color = colorRampPalette(c("white", blues9)),
  transformation = function(x) x^0.25, nbin = 128, nrpoints = 100,
  bandwidth)
```

Arguments

`x`	a radviz object as produced by `do.radviz`
`main`	[Optional] a title to the graph, displayed on top
`label.color`	The color of the Dimensional Anchors (defaults to orangered4)
`label.size`	numeric character expansion factor for Dimensional Anchor labels; multiplied by `par("cex")` yields the final character size. NULL and NA are equivalent to 1.0
`smooth.color`	`function` accepting an integer n as an argument and returning n colors (see smoothScatter for details)
`transformation`	`function` mapping the density scale to the color scale
`nbin`	numeric vector of length one (for both directions) or two (for x and y separately) specifying the number of equally spaced grid points for the density estimation; directly used as gridsize in bkde2D (see smoothScatter for details)
`nrpoints`	number of points to be superimposed on the density image (see smoothScatter for details)
`bandwidth`	numeric vector (length 1 or 2) of smoothing bandwidth(s). If missing, a more or less useful default is used. bandwidth is subsequently passed to function bkde2D (see smoothScatter for details)

Details

The add allows plotting of additional data such as cluster centers onto an existing plot.

Author(s)

Yann Abraham

Florian Hahne

See Also

smoothScatter for original implementation

Examples

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

---

subset.radviz *Title*

Description

Title

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

Value

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

Author(s)

Yann Abraham

Examples

```
# create a radviz object
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 Summary*

Description

Provides a summary for Radviz objects

Usage

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

Arguments

object	an object of class Radviz, as returned by do.radviz
n	the number of lines from each slots in the Radviz object to display (defaults to 6)
...	further arguments to be passed to or from other methods

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)

summary(rv)
```

text.radviz *Text annotations for for the Radviz Plots*

Description

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

Usage

```
## S3 method for class 'radviz'
text(x, ..., main = NULL, label.color = "orangered4",
  label.size = 1, labels = NULL, adj = NULL, pos = NULL, offset = 0.5,
  vfont = NULL, cex = 1, col = NULL, font = NULL, add = TRUE)
```

Arguments

x	a radviz object as produced by do.radviz
...	further arguments to be passed to or from other methods
main	[Optional] a title to the graph, displayed on top if add is TRUE
label.color	[Optional] The color of the Dimensional Anchors if add is TRUE (defaults to orangered4)
label.size	[Optional] numeric character expansion factor for Dimensional Anchor labels if add is TRUE; multiplied by par("cex") yields the final character size. NULL and NA are equivalent to 1.0
labels	a character vector specifying the text to be written. An attempt is made to coerce other language objects (names and calls) to characters using as.graphicsAnnot. If labels is longer than x and y, the coordinates are recycled to the length of labels.
adj	one or two values in [0, 1] which specify the x (and optionally y) adjustment of the labels. On most devices values outside that interval will also work.
pos	a position specifier for the text. If specified this overrides any adj value given. Values of 1, 2, 3 and 4, respectively indicate positions below, to the left of, above and to the right of the specified coordinates.
offset	when pos is specified, this value gives the offset of the label from the specified coordinate in fractions of a character width.
vfont	NULL for the current font family, or a character vector of length 2 for Hershey vector fonts. The first element of the vector selects a typeface and the second element selects a style. Ignored if labels is an expression.
cex	numeric character expansion factor; multiplied by par("cex") yields the final character size. NULL and NA are equivalent to 1.0.
col, font	the color and (if vfont = NULL) font to be used, possibly vectors. These default to the values of the global graphical parameters in par().
add	Logical: if add is TRUE then only the projected points are plotted

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,point.shape=1,point.color='grey')
med.iris <- split(iris,iris$Species)
med.iris <- lapply(med.iris,function(df) {
   spc <- unique(df$Species)
   df <-df[,names(df)!='Species']
   df <- apply(df,2,median)
   df <- data.frame(t(df))
   df$Species <- spc
   return(df)
 }
)
med.iris <- do.call('rbind',med.iris)
med.rv <- do.radviz(med.iris,S)
text(med.rv,labels=med.iris$Species,col=c('red','green','blue')[as.integer(med.iris$Species)])
```

Index