Package ‘fisheyeR’

February 19, 2015

Type Package
Title Fisheye and Hyperbolic-space-alike Interactive Visualization Tools in R
Version 0.9
Date 2009-12-29
Author Eduardo San Miguel Martin
Maintainer Eduardo San Miguel Martin
<eduardo.san.miguel.martin@gmail.com>
Description fisheyeR provides tools for creating Interactive Data Visualizations by implementing ideas from Furnas, Munzner, Costa and Venturini.
License GPL-2
LazyLoad yes
Depends tkrplot, methods
Repository CRAN
Repository/R-Forge/Project fisheyer
Repository/R-Forge/Revision 10
Date/Publication 2010-01-28 07:49:41
NeedsCompilation no

R topics documented:

fisheyeR-package .................................................. 2
addNoise ............................................................. 4
centrarSalida ...................................................... 5
circulin .............................................................. 5
circulo .............................................................. 6
fishlin ............................................................... 7
HeavyWavoidCluttering .......................................... 8
IncVadjustMinus .................................................... 9
mPOIAnd-class .................................................. 10
Description

fisheyeR provides tools for creating Interactive Data Visualizations.

Details

Package: fisheyeR
Type: Package
Version: 0.9
Date: 2010-01-10
License: GPL-2
LazyLoad: yes
Depends: methods


A Fisheye effect allows you to selectively scale information such that readability is preserved for the part relevant to the user, while the rest remains available in a reduced form to serve as context.
Displaying information in a hyperbolic space commonly utilizes the Poincare disk model of hyperbolic geometry, though the Klein-Beltrami model can also be used. Both display the entire hyperbolic plane within a unit disk, making the entire set visible at once. The unit disk gives a fish-eye lens view of the plane, giving more emphasis to elements which are in focus and displaying elements further out of focus closer to the boundary of the disk.

Venturini and Costa Points Of Interest (POI) allows for the exploration of multidimensional data, by representing information according to its similarity with every POI defined for the set.

See references for details.

Author(s)

Eduardo San Miguel Martin

Maintainer: Eduardo San Miguel Martin <eduardo.san.miguel.martin@gmail.com>

References


See Also

plotPOI
addNoise (Add noise to a matrix)

Description
Add noise to duplicated rows in a matrix.

Usage
addNoise(m, tamanyo = 0.01)

Arguments
- m: Matrix to add noise to.
- tamanyo: Size of noise added.

Details
addNoise goal is to avoid any row duplicate in a matrix by adding a small (normal) noise to it.

Value
Original matrix with no duplicates.

Author(s)
Eduardo San Miguel Martin

See Also
fishIn, fishOut, PIIPlot-methods

Examples
addNoise(matrix(rep(1,100), ncol = 2))
**centrarSalida**  
*Force plot coordinates*

**Description**
For internal use of POIPlot function.

**Usage**
```
centrarSalida()
```

**Details**
For internal use of poiPLOT. Force plot coordinates to 0,0.

**Author(s)**
Eduardo San Miguel Martin

**See Also**
`fishIn, fishOut, POIPlot-methods`

---

**circulin**  
*Coordinates to plot a circle*

**Description**
For interactive use via middlebutton click.

**Usage**
```
circulin(cx, cy, r = 0.045, objeto, col = "blue", PLOT = TRUE, label = 0)
```

**Arguments**
- `cx`: x coordinate for circle center
- `cy`: y coordinate for circle center
- `r`: Radius
- `objeto`: Matrix with points plotted
- `col`: Circle color to be plotted
- `PLOT`: Should circle be plotted?
- `label`: Apply labels to plot?
Details
Not to be called directly by user.

Value
A matrix containing circle coordinates to be plotted. A vector (called insiders) containing elements selected is created in POI.env environment.

Author(s)
Eduardo San Miguel Martin

See Also
circulo, POIPlot-methods

circulo  

Function to plot a circle

Description
Internal function. Not intended to be called directly by user.

Usage
circulo(cx, cy, r, circleCol, PLOT = TRUE)

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cx</td>
<td>x coordinate for circle center</td>
</tr>
<tr>
<td>cy</td>
<td>y coordinate for circle center</td>
</tr>
<tr>
<td>r</td>
<td>Radius</td>
</tr>
<tr>
<td>circleCol</td>
<td>Circle color to be plotted</td>
</tr>
<tr>
<td>PLOT</td>
<td>Should circle be plotted?</td>
</tr>
</tbody>
</table>

Details
Not to be called directly by user.

Value
A matrix containing circle coordinates to be plotted.

Author(s)
Eduardo San Miguel Martin
**fishIn**

See Also

circulin, POIPlot-methods

---

**fishIn**

*Nonlinear Focus+Context Transformations*

**Description**

Functions to apply nonlinear Focus+Context transformations

**Usage**

```r
fishIn(x, value)
fishOut(x, value)
```

**Arguments**

- `x` A number
- `value` nonlinear factor to apply

**Details**

Geometric transformations are widely used in interface design, particularly in visualization systems where the amount of information to display exceeds available screen resolution, and in situations that require navigation through a two- or threedimensional scene.

Likewise, scaling is extremely popular; for example, thumbnails are widely used as icons. Unfortunately, scaling only works to a certain extent: When the size of an image is reduced too far, its details become indiscernible.

One possible remedy is to selectively scale it such that readability is preserved for the part of the image relevant to the user, while the rest remains available in a reduced form to serve as context.

The class of Focus+Context techniques does so by providing both an unscaled focus and a scaled-down context in a single integrated image. Focus+Context can be realized using a nonlinear transformation called a fisheye transformation, which has two main variants: rectangular and polar. See references.

**Value**

Number transformed by the factor applied.

**Author(s)**

Eduardo San Miguel Martin
References


See Also

POIPlot-methods, plotPOI

Examples

sapply(seq(0,1,.1), fishIn, 3)
sapply((sapply(seq(0,1,.1), fishIn, 3)), fishOut, 3)

HeavyWavoidCluttering  Find Similar Rows in a Matrix

Description

Function to retrieve UNIQUE rows in a matrix.

Usage

HeavyWavoidCluttering(object, value = 3)
HeavyWavoidCluttering(object, value = 3)

Arguments

object  Matrix object to be grouped.
value   Grouping factor.

Details

HeavyWavoidCluttering and H1WavoidCluttering are a naive way to eliminate similar rows in a matrix by calling duplicated on their rounded value.

Value

newobjeto  Matrix without duplicated rows.
uniques    Vector with Unique rows in original matrix
clusters   Vector where non-zero entries are duplicated elements in original matrix.

See Also

duplicated
IncVadjustMinus

Functions for GUI interaction

Description
Not intender to be called directly by user. See details.

Usage
IncVadjustMinus()
IncVadjustPlus()
MadjustMinus()
MadjustPlus()
MouseWheel(D)
OnClickMotion(x, y)
OnDobleClick(x, y)
OnMiddleClick(x, y)
resetear(x, y)

Arguments
x x coordinate value.
y y coordinate value.
D Mousewheel value.

Details
These functions are used to allow user interaction. tkrplot package is needed.

'+'
Increments fisheye distortion factor. Same functionality using mousewheel.
'-' Decrements fisheye distortion factor. Same functionality using mousewheel.
'0'
Increments animation smoothness.
'.'
Decrements animation smoothness.
Mouse Double Clicking
Retunrs closest point.
Mouse Middle Clicking
Draw a circle and returns and object of class vector called 'insiders' with every element inside. "insiders" object will be created in POI.env. (POI.env$insiders)
Mouse Click Motion
Drag points over. Basic interaction which allows you to move any point to the center of the disk, where detail is augmented.

Author(s)
Eduardo San Miguel Martin

See Also
POIPlot-methods,POI-class,plotPOI,POPlot

mPOIAnd-class
Class "mPOIAnd"

Description
Multi POI (logical AND) object

Objects from the Class
Objects can be created by calls of the form new("mPOIAnd", ...).

Slots
matrizSim: Object of class "matrix" with similarity matrix between set and POIs
cos.query.docs: Object of class "vector" with Similarity of each element of the set
wordsInQuery: Object of class "ANY" Items used as POIs
docs: Object of class "matrix" Items of the set
objeto: Object of class "matrix" 2D Coordinates of the set
objetoC: Object of class "matrix" 2D Coordinates of the set with fisheye effect applied
pcoords: Object of class "matrix" 2D coordinates of the POIs
pcoordsFI: Object of class "matrix" 2D coordinates of the POIs with fisheye effect applied
newpcoords: Object of class "matrix" Use by tkrplot to allow graphical user interaction
newcoords: Object of class "numeric" Use by tkrplot to allow graphical user interaction
newcoords_1: Object of class "numeric" Use by tkrplot to allow graphical user interaction
M: Object of class "numeric" Fisheye Factor to Apply
poisTextCol: Object of class "character" Color to be plotted
colores: Object of class "vector" Color to be plotted
poisCircleCol: Object of class "character" Color to be plotted
linesCol: Object of class "character" Color to be plotted
itemsCol: Object of class "character" Color to be plotted
mPOIAnd-class

LABELS: Object of class "logical" Should POIs be plotted?
vscale: Object of class "numeric" Vertical size of plot
hscale: Object of class "numeric" Horizontal size of plot
circleCol: Object of class "character" Color to be plotted
plotCol: Object of class "character" Color to be plotted
itemsFamily: Object of class "character" Font to use
pal: Object of class "character" Color to be plotted
selected: Object of class "numeric" Used by tkplot to allow graphical user interaction
circRadio: Object of class "numeric" Radio of circle use to select points.
incVsacle: Object of class "numeric" Animation smoth factor
cgnsphrFont: Object of class "numeric" Font to use for labels
xClick_old: Object of class "numeric" Used by tkplot to allow graphical user interaction
yClick_old: Object of class "numeric" Used by tkplot to allow graphical user interaction
wordsInQueryFull: Object of class "character"
clustered: Object of class "logical" should clustered plot be use?

Extends

Class "multiPOI", directly. Class "POI", by class "multiPOI", distance 2.

Methods

No methods defined with class "mPOIAnd" in the signature.

Author(s)

Eduardo San Miguel Martin

References


See Also

plotPOI, POIPlot P0I

### mPOIOr-class

#### Class "mPOIOr"

**Description**

Multi POI (logical OR) object

**Objects from the Class**

Objects can be created by calls of the form `new("mPOIOr", ...)`. 

**Slots**

- `matrizSim` Object of class "matrix" with similarity matrix between set and POIs 
- `cos.query.docs` Object of class "vector" with Similarity of each element of the set 
- `wordsInQuery` Object of class "ANY" Items used as POIs 
- `docs` Object of class "matrix" Items of the set 
- `objet0` Object of class "matrix" 2D Coordinates of the set 
- `objet0C` Object of class "matrix" 2D Coordinates of the set with fisheye effect applied 
- `pcoords` Object of class "matrix" 2D coordinates of the POIs 
- `pcoordsFI` Object of class "matrix" 2D coordinates of the POIs with fisheye effect applied 
- `newPcoords` Object of class "matrix" Use by tkrplot to allow graphical user interaction 
- `newcoords` Object of class "numeric" Use by tkrplot to allow graphical user interaction 
- `newcoords_1` Object of class "numeric" Use by tkrplot to allow graphical user interaction 
- `M` Object of class "numeric" Fisheye Factor to Apply 
- `poisTextCol` Object of class "character" Color to be plotted 
- `colores` Object of class "vector" Color to be plotted 
- `poisCircleCol` Object of class "character" Color to be plotted 
- `linesCol` Object of class "character" Color to be plotted 
- `itemsCol` Object of class "character" Color to be plotted 
- `LABELS` Object of class "logical" Should POIs be plotted? 
- `vscale` Object of class "numeric" Vertical size of plot 
- `hscale` Object of class "numeric" Horizontal size of plot
circleCol: Object of class "character" Color to be plotted
plotCol: Object of class "character" Color to be plotted
itemsFamily: Object of class "character" Font to use
pal: Object of class "character" Color to be plotted
selected: Object of class "numeric" Used by tkrplot to allow graphical user interaction
circRadio: Object of class "numeric" Radio of circle use to select points.
incVsclae: Object of class "numeric" Animation smooth factor
cgnsphrFont: Object of class "numeric" Font to use for labels
xClick_old: Object of class "numeric" Used by tkrplot to allow graphical user interaction
yClick_old: Object of class "numeric" Used by tkrplot to allow graphical user interaction
wordsInQueryFull: Object of class "character"
clustered: Object of class "logical" should clustered plot be use?

Extends

Class "multiPOI", directly. Class "POI", by class "multiPOI", distance 2.

Methods

No methods defined with class "mPOIOr" in the signature.

Author(s)

Eduardo San Miguel Martin

References


multiPOI-class

Class "multiPOI"

Description

multiPOI class extends POI class in order to represent POIs as set of POIs

Objects from the Class

Objects can be created by calls of the form new("multiPOI", ...).

Slots

matrizSim: Object of class "matrix" with similarity matrix between set and POIs
cos.query.docs: Object of class "vector" with Similarity of each element of the set
wordsInQuery: Object of class "ANY" Items used as POIs
docs: Object of class "matrix" Items of the set
objeto: Object of class "matrix" 2D Coordinates of the set
objetoC: Object of class "matrix" 2D Coordinates of the set with fisheye effect applied
pcoords: Object of class "matrix" 2D coordinates of the POIs
pcoordsFI: Object of class "matrix" 2D coordinates of the POIs with fisheye effect applied
newpcoords: Object of class "matrix" Use by tkrplot to allow graphical user interaction
newcoords: Object of class "numeric" Use by tkrplot to allow graphical user interaction
newcoords_1: Object of class "numeric" Use by tkrplot to allow graphical user interaction
M: Object of class "numeric" Fisheye Factor to Apply
poisTextCol: Object of class "character" Color to be plotted
colores: Object of class "vector" Color to be plotted
poisCircleCol: Object of class "character" Color to be plotted
linesCol: Object of class "character" Color to be plotted
itemsCol: Object of class "character" Color to be plotted
LABELS: Object of class "logical" Should POIs be plotted?
vscale: Object of class "numeric" Vertical size of plot
hscale: Object of class "numeric" Horizontal size of plot
circleCol: Object of class "character" Color to be plotted
plotCol: Object of class "character" Color to be plotted
itemsFamily: Object of class "character" Font to use
pal: Object of class "character" Color to be plotted

See Also

plotPOI, POIPlot POI
multiPOI-class

selected: Object of class "numeric" Used by tkrplot to allow graphical user interaction
circRadio: Object of class "numeric" Radio of circle use to select points.
IncVscale: Object of class "numeric" Animation smooth factor
cgnsphrFont: Object of class "numeric" Font to use for labels
xClick_old: Object of class "numeric" Used by tkrplot to allow graphical user interaction
yClick_old: Object of class "numeric" Used by tkrplot to allow graphical user interaction
wordsInQueryFull: Object of class "character"
clustered: Object of class "logical" should clustered plot be use?

Extends

Class "POI", directly.

Methods

POIcalculate signature(object = "multiPOI"):

POIPlot signature(POI = "multiPOI"):

Author(s)

Eduardo San Miguel Martin

References


See Also

plotPOI, POIPlot P0I
plotPOI

Functions for POI plotting

Description
Function for plotting objects of class POI.

Usage
plotPOI(POI)
plotPOIGraph(POI)

Arguments
POI Object of class POI

Details
plotPOI and plotPOIGraph will try to load tkrplot in order to allow graphical user interaction. Otherwise POIPlot will be used.

Graphical User Interaction defaults:
Mouse Click Motion
Drag points over. Basic interaction which allows you to move any point to the center of the disk, where detail is augmented.
'+'
Increments fisheye distortion factor. Same functionality using mousewheel.
'-'
Decrements fisheye distortion factor. Same functionality using mousewheel.
'0'
Increments animation smoothness.
'.'
Decrements animation smoothness.

Mouse Double Clicking
Returns closest point.

Mouse Middle Clicking
Draw a circle and returns and object of class vector called 'insiders' with every element inside. "insiders" object will be created in POI.env. (POI.env$insiders)

See Also
POI-class, POIPlot-methods, POIPlot
Examples

## Not run:
```r
## rgb colors
rgbPOI = POICreate(type = 'POI', wordsInQuery = c('red','green','blue'),
colores = colors(), itemsCol = colors(),
docs = cbind(colors(), 1:length(colors())),
cos.query.docs = rep(1,length(colors())),
matrixSim = t(t(col2rgb(colors()))) / max(t(t(col2rgb(colors()))))
)
P0Icoords(rgbPOI) <- POICalc(rgbPOI ,length(rgbPOI@wordsInQuery))
try(rm('POI.env'), silent = T)
plotPOI(rgbPOI)
```

## graph example
```
# igraph package -- graph.tree example looks great!
if (require(igraph)) {
  GRAPH <- graph.tree(500, children = 10, mode = 'in')
fCompress <- 350 # compress factor
graphPOI <- POICreate(type = 'POIGraph')
graphPOI@objetos <- layout.fruchterman.reingold(GRAPH,dim = 2) / fCompress
graphPOI@EDGES <- cbind(GRAPH[[3]],GRAPH[[4]]) + 1
graphPOI@docs <- matrix(c(seq(1:nrow(graphPOI@objetos)), seq(1:nrow(graphPOI@objetos))), ncol = 2)
try(rm('POI.env'), silent = T)
plotPOIGraph(graphPOI)
}
```

# manually made -- but igraph example looks great!!
```
graphPOI <- POICreate(type = 'POIGraph')
graphPOI@objetos <- graphPOI@objetos <- cbind(c(0,.05), c(.05,0), c(0,-.05), c(-.05,0) ,round(circulo(0,0,.3,PL)))
graphPOI@EDGES <- matrix(c(rep(1,25), rep(2,25), rep(3,25), rep(4,25), seq(1,100)), ncol = 2)
graphPOI@docs <- matrix(c(seq(1:nrow(graphPOI@objetos)), seq(1:nrow(graphPOI@objetos))), ncol = 2)
graphPOI@colores <- c(rep(2,25), rep(3,25), rep(4,25), rep(5,25))
try(rm('POI.env'), silent = T)
plotPOIGraph(graphPOI)
```

## IRIS Example
```
data(iris)
# distance of each element to each dimension max and min
matrixSim = cbind(
  1 - (max(iris[,1]) - iris[,1]) / (max(max(iris[,1]) - iris[,1])),
  1 - (max(iris[,2]) - iris[,2]) / (max(max(iris[,2]) - iris[,2])),
  1 - (max(iris[,3]) - iris[,3]) / (max(max(iris[,3]) - iris[,3])),
  1 - (max(iris[,4]) - iris[,4]) / (max(max(iris[,4]) - iris[,4])),
  1 - (min(iris[,1]) - iris[,1]) / (min(min(iris[,1]) - iris[,1])),
  1 - (min(iris[,2]) - iris[,2]) / (min(min(iris[,2]) - iris[,2])),
  1 - (min(iris[,3]) - iris[,3]) / (min(min(iris[,3]) - iris[,3])),
  1 - (min(iris[,4]) - iris[,4]) / (min(min(iris[,4]) - iris[,4]))
)
matrixSim = matrixSim^3
irisPOI = POICreate('POI')
irisPOI@matrixSim <- matrixSim
irisPOI@wordsInQuery <- c('High.Sepal.Length', 'High.Sepal.Width',
```

POIcoords(irisPOI) <- POICalc(irisPOI, length(irisPOI@wordsInQuery))
irisPOIdocs <- cbind(matrix(seq(1:nrow(irisPOI@objeto)), matrix(seq(1:nrow(irisPOI@objeto)))))
irisPOI@colores <- c(rep(2,50), rep(3,50), rep(4,50))
try(rm('POI.env'), silent = T)
plotPOI(irisPOI)

## USAReat Example
# POIS = (high - low) murder, assault and rape rates
# colors = Population
data(USAReat)
matrizSim = cbind(
  1 - (max(USArrests[,1] - USArrests[,1]) / (max(max(USArrests[,1]) - USArrests[,1]))),
  1 - (max(USArrests[,2] - USArrests[,2]) / (max(max(USArrests[,2]) - USArrests[,2]))),
  1 - (max(USArrests[,4] - USArrests[,4]) / (max(max(USArrests[,4]) - USArrests[,4]))),
  1 - (min(USArrests[,1] - USArrests[,1]) / (min(min(USArrests[,1]) - USArrests[,1]))),
  1 - (min(USArrests[,2] - USArrests[,2]) / (min(min(USArrests[,2]) - USArrests[,2]))),
  1 - (min(USArrests[,4] - USArrests[,4]) / (min(min(USArrests[,4]) - USArrests[,4])))
)

usaPOI = POICreate('POI')
usaPOI@matrizSim <- matrizSim
usaPOI@wordsInQuery <- c(paste('High', names(USArrests[,c(1,2,4)])), paste('Low', names(USArrests[,c(1,2,4)])),
POIcoords(usaPOI) <- POICalc(usaPOI, length(usaPOI@wordsInQuery))
usaPOI@docs <- cbind(matrix(rnames(USArrests)), matrix(seq(1:nrow(usaPOI@objeto))))
usaPOI@cos.query.docs <- USArrests[,3] / max(USArrests[,3])
POIcolors(usaPOI) <- query2Cols(usaPOI, 'terrain')
try(rm('POI.env'), silent = T)
plotPOI(usaPOI)

## clusters EXAMPLE
x <- matrix(rnorm(1500, mean = 0, sd = .5), ncol = 5)
atipV1 = sample(nrow(x), as.integer(nrow(x)/3)) # outliers in V1
atipV2 = sample(nrow(x), as.integer(nrow(x)/3)) # outliers in V2
x[atipV1, 1] <- rnorm(100, mean = 2, sd = .5)
x[atipV2, 2] <- rnorm(100, mean = 2, sd = .5)
c1 <- kmeans(x, 3, iter.max = 100, nstart = 25)
matrizSim = sqrt(round((x - colMeans(x))^2,1)/nrow(x)) # similarity within outliers
# OR (uncomment one)
# matrizSim = 1 - sqrt(round((x - colMeans(x))^2,1)/nrow(x)) # similarity within mean
varPOI = POICreate('POI')
varPOI@matrizSim <- matrizSim
varPOI@wordsInQuery <- 1:ncol(matrizSim)
POIcoords(varPOI) <- POICalc(varPOI, length(varPOI@wordsInQuery))
# if elements labels bother
varPOI@docs <- cbind(rep('', nrow(varPOI@objeto)), matrix(seq(1:nrow(varPOI@objeto))))
varPOI@cos.query.docs <- rep(1,nrow(matrizSim))
varPOI@colores <- c1$cluster + 1
try(rm('POI.env'), silent = T)
plotPOI(varPOI)

## End(Not run)
**Description**

A Class representing a set of elements and its relations. See references for details.

**Objects from the Class**

Objects can be created by calls of the form `new("POI", ...)`. A POI object contains elements necessary for representing relations.

**Slots**

- `matrizSim`: Object of class "matrix" with similarity matrix between set and POIs
- `cosNqueryNdocs`: Object of class "vector" with Similarity of each element of the set
- `wordsInQuery`: Object of class "ANY" Items used as POIs
- `docs`: Object of class "matrix" Items of the set
- `objeto`: Object of class "matrix" 2D Coordinates of the set
- `objetoc`: Object of class "matrix" 2D Coordinates of the set with fisheye effect applied
- `coords`: Object of class "matrix" 2D coordinates of the POIs
- `coordsFl`: Object of class "matrix" 2D coordinates of the POIs with fisheye effect applied
- `newcoords`: Object of class "numeric" Use by tkrplot to allow graphical user interaction
- `newcoords_1`: Object of class "numeric" Use by tkrplot to allow graphical user interaction
- `M`: Object of class "numeric" Fisheye Factor to Apply
- `poisTextCol`: Object of class "character" Color to be plotted
- `colores`: Object of class "vector" Color to be plotted
- `poisCircleCol`: Object of class "character" Color to be plotted
- `linesCol`: Object of class "character" Color to be plotted
- `itemsCol`: Object of class "character" Color to be plotted
- `LABELS`: Object of class "logical" Should POIs be plotted?
- `vscale`: Object of class "numeric" Vertical size of plot
- `hscale`: Object of class "numeric" Horizontal size of plot
- `circleCol`: Object of class "character" Color to be plotted
- `plotCol`: Object of class "character" Color to be plotted
- `itemsFamily`: Object of class "character" Font to use
- `pal`: Object of class "character" Font to use
- `selected`: Object of class "numeric" Used by tkrplot to allow graphical user interaction
circRadio: Object of class "numeric" Radio of circle use to select points.
IncVsScale: Object of class "numeric" Animation smooth factor
cgnsphrFont: Object of class "numeric" Font to use for labels
xClick_old: Object of class "numeric" Used by tkplot to allow graphical user interaction
yClick_old: Object of class "numeric" Used by tkplot to allow graphical user interaction
wordsInQueryFull: Object of class "character"
clustered: Object of class "logical" should clustered plot be use?

Methods

POICalc signature(objeto = "POI"):

POIcalculate<- signature(object = "POI"):

POIcolors<- signature(object = "POI"):

POIcoords<- signature(object = "POI"):

POIPlot signature(POI = "POI"):

query2Cols signature(object = "POI"):

Author(s)

Eduardo San Miguel Martin

References


See Also

plotPOI, POIPlot multiPOI
POICalc

Function to generate coordinates points to plot using POIs

Description

POICalc Points Of Interest (POI) allows for the exploration of multidimensional data, by representing information according to its similarity with every POI defined for the set.

Usage

POICalc(objeto, NC, cx = 0, cy = 0, r = 1, ...)

Arguments

- **objeto**: Object of class POI
- **NC**: Number of POI (points of interest as proposed by Costa and Venturini. See references.
- **cx**: x coordinates
- **cy**: x coordinates
- **r**: Plot Radius
- **...**: further arguments

Details

POIs are located on a circle, and data are displayed within this circle according to their similarities to these POI. Interactive actions are possible: selection, zoom, dynamical change of POI.

Value

- **pcoords**: Matrix with POIs coordinates
- **pcoordsFI**: Matrix with POIs coordinates with fisheye effect applied.
- **newpcoords**: Matrix with coordinates for the lines joining POIs
- **objeto**: Matrix with coordinates for elements in the main set.

Author(s)

Eduardo San Miguel Martin
References


See Also

POIPlot-methods,POI-class,plotPOI

Examples

## Not run:
## IRIS Example
data(iris)

# distance of each element to each dimension max and min
matrizSim = cbind(
  1 - (max(iris[,1]) - iris[,1]) / (max(max(iris[,1]) - iris[,1])),
  1 - (max(iris[,2]) - iris[,2]) / (max(max(iris[,2]) - iris[,2])),
  1 - (max(iris[,3]) - iris[,3]) / (max(max(iris[,3]) - iris[,3])),
  1 - (max(iris[,4]) - iris[,4]) / (max(max(iris[,4]) - iris[,4])),
  1 - (min(iris[,1]) - iris[,1]) / (min(min(iris[,1]) - iris[,1])),
  1 - (min(iris[,2]) - iris[,2]) / (min(min(iris[,2]) - iris[,2])),
  1 - (min(iris[,3]) - iris[,3]) / (min(min(iris[,3]) - iris[,3])),
  1 - (min(iris[,4]) - iris[,4]) / (min(min(iris[,4]) - iris[,4]))
)

# exaggerate diffs
matrizSim = matrizSim^3

# Create POI plot
irisPOI = POICreate('POI')
irisPOI@matrizSim <- matrizSim
POICalc-methods

Methods for Function POICalc

Description
Methods for function POICalc

Methods

objeto = "POI" Function to generate coordinates points to plot using POIs

POICalc-methods

POicalc<- Replacement function for POI(Pcoords,PcoordsFI,newPcoords,objeto) slots

Description
Replacement function for POI(Pcoords,PcoordsFI,newPcoords,objeto) slots

Usage
POicalc(object) <- value

Arguments
object Object of class POI
value Values to fill slots with

Details
To be used with POICalc

Value
POI object with updated slots

See Also
POIPlot-methods,POI-class,plotPOI
POIcalculate<-methods

Methods for Function POIcalculate<-

Description

Methods for function POIcalculate<-

Methods

object = "multiPOI"  Replacement method
object = "POI"      Replacement method

POIcolors<-

POI-class slot Colores Replacement function

Description

Fill colores slot with appropriate value.

Usage

POIcolors(object) <- value

Arguments

object  Object of class POI
value   Object of class Vector with colors to apply

Value

Replace slot colores in object

POIcolors<--methods

Methods for Function POIcolors<-

Description

Methods for function POIcolors<-

Methods

object = "POI"  Fill slot colores
POIcoords <- Update POI slots

Description
This function allows POI to update its Pcoords, PcoordsFL newPcoords and objeto Slots.

Usage
POIcoords(object) <- value

Arguments
- object: Object of class POI
- value: object of class list with values to use for updating

Details
Usually POICalc returned value will be used as ‘value’ argument for POIcoords.

Value
Original POI object with slots updated.

See Also
POIPlot-methods, POI-class, plotPOI

Examples

## IRIS Example
data(iris)

# distance of each element to each dimension max and min
matrizSim = cbind(
    1 - (max(iris[,1]) - iris[,1]) / (max(max(iris[,1])) - iris[,1]),
    1 - (max(iris[,2]) - iris[,2]) / (max(max(iris[,2])) - iris[,2]),
    1 - (max(iris[,3]) - iris[,3]) / (max(max(iris[,3])) - iris[,3]),
    1 - (max(iris[,4]) - iris[,4]) / (max(max(iris[,4])) - iris[,4]),
    1 - (min(iris[,1]) - iris[,1]) / (min(min(iris[,1])) - iris[,1]),
    1 - (min(iris[,2]) - iris[,2]) / (min(min(iris[,2])) - iris[,2]),
    1 - (min(iris[,3]) - iris[,3]) / (min(min(iris[,3])) - iris[,3]),
    1 - (min(iris[,4]) - iris[,4]) / (min(min(iris[,4])) - iris[,4]))

# exaggerate diffs
matrizSim = matrizSim^3

# Create POI plot
irisPOI = POICreate('POI')
irisPOI@matrizSim <- matrizSim
POIcoords(irisPOI) <- POICalc(irisPOI, length(irisPOI@wordsInQuery))
irisPOI@docs <- cbind(matrix(seq(1:nrow(irisPOI@objeto))), matrix(seq(1:nrow(irisPOI@objeto))))
irisPOI@colores <- c(rep(2,50), rep(3,50), rep(4,50))
try(rm('POI.env'), silent = T)
plotPOI(irisPOI)

POIcoords<-methods  

Methods for Function POIcoords<-

Description
Methods for function POIcoords<-

Methods

object = "POI"  Method that allow POI to update its Pcoords, PcoordsFI, newPcoords and objeto 

Slots

POICreate  

Create an POI

Description
Function to create an object of class POI, POIGraph, multiPOI, mPOIAnd or mPOIOr

Usage
POICreate(type = "POI", ...)

Arguments

type  

Type of object to create: POI, POIGraph, multiPOI, mPOIAnd or mPOIOr

...

Further arguments to create objeto. Slots values of each class.

Value
A new object of selected class.

See Also

POI-class
**POIGraph-class**

**Class “POIGraph”**

### Description

Class for representing graphs.

### Objects from the Class

Objects can be created by calls of the form `new("POIGraph", ...)`.

### Slots

- **EDGES**: Object of class "matrix" with edges of the graph
- **matrizSim**: Object of class "matrix" with similarity matrix between set and POIs
- **cos.query.docs**: Object of class "vector" with similarity of each element of the set
- **wordsInQuery**: Object of class "ANY" Items used as POIs
- **docs**: Object of class "matrix" Items of the set
- **objeto**: Object of class "matrix" 2D Coordinates of the set
- **objetoC**: Object of class "matrix" 2D Coordinates of the set with fisheye effect applied
- **pcoords**: Object of class "matrix" 2D coordinates of the POIs
- **pcoordsFI**: Object of class "matrix" 2D coordinates of the POIs with fisheye effect applied
- **newPcoords**: Object of class "matrix" Use by tkrplot to allow graphical user interaction
- **newcoords**: Object of class "numeric" Use by tkrplot to allow graphical user interaction
- **newcoords_1**: Object of class "numeric" Use by tkrplot to allow graphical user interaction
- **M**: Object of class "numeric" Fisheye Factor to Apply
- **poisTextCol**: Object of class "character" Color to be plotted
- **colores**: Object of class "vector" Color to be plotted
- **poisCircleCol**: Object of class "character" Color to be plotted
- **linesCol**: Object of class "character" Color to be plotted
- **itemsCol**: Object of class "character" Color to be plotted
- **LABELS**: Object of class "logical" Should POIs be plotted?
- **vscale**: Object of class "numeric" Vertical size of plot
- **hsize**: Object of class "numeric" Horizontal size of plot
- **circleCol**: Object of class "character" Color to be plotted
- **plotCol**: Object of class "character" Color to be plotted
- **itemsFamily**: Object of class "character" Font to use
- **pal**: Object of class "character" Color to be plotted
selected: Object of class "numeric" Used by tkrplot to allow graphical user interaction
circRadio: Object of class "numeric" Radio of circle use to select points.
IncVscale: Object of class "numeric" Animation smooth factor
cgnsphrFont: Object of class "numeric" Font to use for labels
xClick_old: Object of class "numeric" Used by tkrplot to allow graphical user interaction
yClick_old: Object of class "numeric" Used by tkrplot to allow graphical user interaction
wordsInQueryFull: Object of class "character"
clustered: Object of class "logical" should clustered plot be use?

Extends
Class "POI", directly.

Methods

POIPlot signature(POI = "POIGraph"): ...

Author(s)
Eduardo San Miguel Martin

References


See Also

plotPOI, POIPlot multiPOI
**POIPlot**

*POIPlot*  
*Plot Objects of Class POI*

---

**Description**

Function for plotting objects of class POI.

**Usage**

```r
POIPlot(POI)
```

**Arguments**

- **POI**
  
  An object of class POI

**Details**

See POI class reference for details on POIs

**Value**

A plot (non interactive) of the object.

**See Also**

- `POI-class`
- `plotPOI`

**Examples**

```r
## rgb colors
rgbPOI = POICreate(type = 'POI', wordsInQuery = c('red','green','blue'),
                    colores = colors(), itemsCol = colors(),
                    docs = cbind(colors(), 1:length(colors())),
                    cos.query.docs = rep(1, length(colors())),
                    matrizSim = t(col2rgb(colors())) / max(t(col2rgb(colors()))))

POIcoords(rgbPOI) <- POICalc(rgbPOI ,length(rgbPOI@wordsInQuery))
rgbPOI@itemsFamily <- '' # R cmd check passing in examples
try(rm('POI.env'), silent = T)
POIPlot(rgbPOI)
```
Description

Methods for function POIPlot

Methods

POI = "POI"  See function documentation for details.
POI = "multiPOI"  See function documentation for details.
POI = "POIGraph"  See function documentation for details.

Description

puntosMedios connects points in 2D coordinates. A function taking 2D coordinates given and joining the corresponding points.

Usage

puntosMedios(Pcoords, detalle = 5)

Arguments

Pcoords  matrix object with coordinates
detalle  Detail

Details

See example below

Value

An object of class matrix with coordinates of the connectors.

See Also

plotPOI, lines

Examples

obj <- matrix(c(.5,0,.5, -.5,0, 0,-.5), ncol = 2, byrow = TRUE)
plot(0, 0, col = 'white')
points(puntosMedios(obj), col = 'yellow', type = 'l')
points(obj, col = 'blue')
query2Cols

Create color vector for elements in an object of class POI

Description

query2Cols assigns a color to every element in the POI, accordingly with its cos.query.docs value. Valid palettes: 'heat', 'topo', 'cm' and 'terrain'.

Note that cos.query.docs value must be in the range [0-1].

Usage

query2Cols(object, value)

Arguments

object object of class POI
value Palette to apply. One in: 'heat','topo','cm','terrain'

Value

A vector with corresponding color to each element of POI.

Author(s)

Eduardo San Miguel Martin

See Also

POIPlot-methods.POI-class.plotPOI

Examples

data(USArrests)
usaPOI = POICreate('POI')
usaPOI@cos.query.docs <- USArrests[,3]/max(USArrests[,3]) # urban population (1 high - 0 low)
usaPOI@colores <- query2Cols(usaPOI, 'terrain')
plot(usaPOI@cos.query.docs, col = usaPOI@colores)

usaPOI@colores <- query2Cols(usaPOI, 'heat')
plot(usaPOI@cos.query.docs, col = usaPOI@colores)
Methods for Function `query2Cols`

**Methods**

- **object = "POI"** Create color vector for a POI

---

### toCartesian

**Converting between Polar and Cartesian Coordinates**

#### Description

The Cartesian system locates points on a plane by measuring the horizontal and vertical distances from an arbitrary origin to a point. These are usually denoted as a pair of values (X,Y).

The Polar system locates the point by measuring the straight line distance, usually denoted by R, from the origin to the point and the angle of an imaginary line from the origin to the point, q, (Greek letter Theta), measured counterclockwise from the positive X axis.

The conversion math is fairly straightforward:

- **Polar from Cartesian:**
  - \( R = \sqrt{x^2 + y^2} \)
  - \( \theta = \arctan\left(\frac{y}{x}\right) \)

- **Cartesian From Polar:**
  - \( x = R \cos(\theta) \)
  - \( y = R \sin(\theta) \)

#### Usage

- `toCartesian(t, r)`
- `toPolar(x, y)`

#### Arguments

- **t**  
  - Theta
- **r**  
  - Radius
- **x**  
  - x coordinate
- **y**  
  - y coordinate
toHiperbolico

See Also
atan2, cos, sin

Examples

toPolar(1, 1)
toCartesian(toPolar(1, 1)[1], toPolar(1, 1)[2])

toHiperbolico  Hyperbolic-alike space effect

Description
This function combined with fisheye effect is used to simulate the hyperbolic space effect.

Usage
toHiperbolico(objeto, M = 1, cx = 0, cy = 0, r = 1)

Arguments

<table>
<thead>
<tr>
<th>objeto</th>
<th>A matrix with object coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>fisheye effect factor</td>
</tr>
<tr>
<td>cx</td>
<td>X coordinate for circle center.</td>
</tr>
<tr>
<td>cy</td>
<td>Y coordinate for circle center.</td>
</tr>
<tr>
<td>r</td>
<td>Radius</td>
</tr>
</tbody>
</table>

Details
Displaying information in a hyperbolic space commonly utilizes the Poincare disk model of hyperbolic geometry, though the Klein-Beltrami model can also be used. Both display the entire hyperbolic plane within a unit disk, making the entire set visible at once. The unit disk gives a fisheye lens view of the plane, giving more emphasis to elements which are in focus and displaying elements further out of focus closer to the boundary of the disk. See references for details.

Value

<table>
<thead>
<tr>
<th>objetoC</th>
<th>Matrix with new object coordinates. Cartesian system</th>
</tr>
</thead>
<tbody>
<tr>
<td>objetoP</td>
<td>Matrix with new object coordinates. Polar system</td>
</tr>
</tbody>
</table>

Author(s)
Eduardo San Miguel Martin
References


See Also

plotPOI, POIPlot, fishIn, fishOut,

Examples

circle1 = circulo(0,0,.25, PLOT = FALSE)
circle2 = circulo(0,0,1.25, PLOT = FALSE)
plot(0,0, xlim = c(-1.25,1.25), ylim= c(-1.25,1.25), col = 'white')
points(circle1, col = 'blue', cex = 0.5)
points(circle2, col = 'blue', cex = 0.5)
points(toHiperbolico(circle2,3)$objetoC, col = 'green', cex = 0.5)
points(toHiperbolico(circle1,3)$objetoC, col = 'green', cex = 0.5)
Index

*Topic methods
  POIcolors<--methods, 24
  POIcoords<--methods, 26
  POIPlot-methods, 30
  query2Cols-methods, 32

  addNoise, 4
  atan2, 33

  centrarSalida, 5
  circulin, 5, 7
  circulo, 6, 6
  cos, 33

  duplicated, 8

  fisheyeR (fisheyeR-package), 2
  fisheyeR-package, 2
  fishIIn, 4, 5, 7, 34
  fishIOut, 4, 5, 34
  fishIOut (fishIIn), 7

  H1WavoidCluttering
    (HeavyWavoidCluttering), 8
  HeavyWavoidCluttering, 8

  IncVadjjustMinus, 9
  IncVadjjustPlus (IncVadjjustMinus), 9

  lines, 30

  MadjustMinus (IncVadjjustMinus), 9
  MadjustPlus (IncVadjjustMinus), 9
  MouseWheel (IncVadjjustMinus), 9
  mPOIAnd-class, 10
  mPOIOr-class, 12
  multiPOI, 11, 13, 20, 28
  multiPOI-class, 14

  OnClickMotion (IncVadjjustMinus), 9
  OnDoubleClick (IncVadjjustMinus), 9

  OnMiddleClick (IncVadjjustMinus), 9

  plotPOI, 3, 8, 10, 12, 14, 15, 16, 20, 22, 23, 25, 28–31, 34
  plotPOIGraph (plotPOI), 16
  POI, 11–15, 28
  POI-class, 19
  POICalc, 21
  POICalc, POI-method (POI-class), 19
  POICalc-methods, 23
  POICalculate<-, 23
  POICalculate<-, multiPOI-method
    (multiPOI-class), 14
  POICalculate<-, POI-method (POI-class), 19

  POICalculate<--methods, 24
  POIColors<-, 24
  POIColors<-, POI-method (POI-class), 19
  POIColors<--methods, 24
  POIcoords<-, 25
  POIcoords<-, POI-method (POI-class), 19
  POIcoords<--methods, 26
  POICreate, 26
  POIGraph-class, 27
  POIPlot, 10, 12, 14–16, 20, 28, 29, 34
  POIPlot, multiPOI-method
    (multiPOI-class), 14
  POIPlot, POI-method (POI-class), 19
  POIPlot, POIGraph-method
    (POIGraph-class), 27
  POIPlot-methods, 30
  puntosMedios, 30

  query2Cols, 31
  query2Cols, POI-method (POI-class), 19
  query2Cols-methods, 32

  resetear (IncVadjjustMinus), 9

  sin, 33
toCartesian, 32
toHiperbolico, 33
toPolar (toCartesian), 32