<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>initializeViz</td>
</tr>
<tr>
<td>initializeViz-methods</td>
</tr>
<tr>
<td>linegraph</td>
</tr>
<tr>
<td>linegraph-methods</td>
</tr>
<tr>
<td>names-methods</td>
</tr>
<tr>
<td>names&lt;-methods</td>
</tr>
<tr>
<td>navGraph</td>
</tr>
<tr>
<td>newgraph</td>
</tr>
<tr>
<td>ng_2d</td>
</tr>
<tr>
<td>ng_2d_myplot</td>
</tr>
<tr>
<td>ng_data</td>
</tr>
<tr>
<td>ng_get</td>
</tr>
<tr>
<td>ng_get-methods</td>
</tr>
<tr>
<td>ng_get_color</td>
</tr>
<tr>
<td>ng_get_size</td>
</tr>
<tr>
<td>ng_graph</td>
</tr>
<tr>
<td>ng_image_array_gray</td>
</tr>
<tr>
<td>ng_image_files</td>
</tr>
<tr>
<td>ng_set</td>
</tr>
<tr>
<td>ng_set-methods</td>
</tr>
<tr>
<td>ng_set&lt;-</td>
</tr>
<tr>
<td>ng_set&lt;-methods</td>
</tr>
<tr>
<td>ng_set_color&lt;-</td>
</tr>
<tr>
<td>ng_set_color&lt;-methods</td>
</tr>
<tr>
<td>ng_set_size&lt;-</td>
</tr>
<tr>
<td>ng_set_size&lt;-methods</td>
</tr>
<tr>
<td>ng_update</td>
</tr>
<tr>
<td>NG_Visualization-class</td>
</tr>
<tr>
<td>ng_walk</td>
</tr>
<tr>
<td>olive</td>
</tr>
<tr>
<td>plot-methods</td>
</tr>
<tr>
<td>scagEdgeWeights</td>
</tr>
<tr>
<td>scagGraph</td>
</tr>
<tr>
<td>scagNav</td>
</tr>
<tr>
<td>shortnames</td>
</tr>
<tr>
<td>shortnames-methods</td>
</tr>
<tr>
<td>shortnames&lt;-</td>
</tr>
<tr>
<td>shortnames&lt;-methods</td>
</tr>
<tr>
<td>updateViz</td>
</tr>
<tr>
<td>updateViz-methods</td>
</tr>
</tbody>
</table>

**Index** 38
Description

GUI to explore high dimensional data using graphs as navigational infrastructure.
This package implements some of the methods described in the Hurley and Oldford paper.
Visualization instructions can be linked to a bullet on the graph. The bullet can be moved along the
graph.
With the tk2d graphics device, provided by the RnavGraph package, one can display images,
glyphs, text and glyphs and modify their location smoothly.

Details

The package vignette provides a detailed overview of the RnavGraph package. Use vignette("RnavGraph").

Author(s)

Adrian Waddell and R. Wayne Oldford

References

C. B. Hurley and R. W. Oldford. "Graphs as navigational infrastructure for high dimensional data

See Also

navGraph

Examples

ng.iris <- ng_data(name = "iris", data = iris[,1:4],
group = iris$Species,
labels = substr(iris$Species,1,2))

nav <- navGraph(ng.iris)
**closeViz-methods**

---

### closeViz

*Method that gets called when switching away from a graph that is connected with the visualization instructions*

---

#### Description

See the vignette for more details.

#### Usage

```r
closeViz(viz, ngEnv)
```

#### Arguments

- `viz` Object of class `NG_Visualization`
- `ngEnv` Environment of `navGraph` session.

#### Value

Object of `NG_Visualization` (viz from argument)

#### Author(s)

Adrian Waddell and Wayne Oldford

#### See Also

- `initializeViz`
- `updateViz`

#### Examples

```r
## See vignette
```

---

### closeViz-methods

*Initialization of a new Display*

---

#### Description

If custom visualization class (NG_Visualization) has been defined. This is the method that gets called if `navGraph` switches away from a graph that links to the visualization class.

See package vignette for more detail.

#### Methods

```r
signature(obj = "NG_Visualization")
```
**completegraph**

Create a complete graph of class graphAM

---

**Description**

(From Wikipedia) A complete graph is a simple graph in which every pair of distinct vertices is connected by a unique edge.

**Usage**

completegraph(nodeNames)

**Arguments**

- **nodeNames**  
  Numeric or character string vector.

**Value**

graphAM object.

**Author(s)**

Adrian Waddell and R. Wayne Oldford

**See Also**

navGraph, newgraph, linegraph

**Examples**

G1 <- completegraph(1:10)
G2 <- completegraph(LETTERS[1:7])

---

**initializeViz**

*Method that gets called for initializing a display for some visualization instructions*

**Description**

See the vignette for more details.

**Usage**

initializeViz(viz,ngEnv)
Arguments

viz Object of class NG_Visualization
ngEnv Environment of navGraph session.

Value

Object of NG_Visualization

Author(s)

Adrian Waddell and Wayne Oldford

See Also

updateViz, closeViz

Examples

## See vignette

---

initializeViz-methods  *Initialization of a new Display*

Description

If custom visualization class (NG_Visualization) has been defined. This is the method that gets called if navGraph switches to a graph that links to the visualization class.

See package vignette for more detail.

Methods

signature(obj = "NG_Visualization")

---

linegraph  *Linegraph of a graph*

Description

(From Wikipedia) The line graph L(G) of an undirected graph G is another graph L(G) that represents the adjacencies between edges of G. By definition, each vertex of L(G) represents an edge of G, and two vertices of L(G) are adjacent if and only if their corresponding edges share a common endpoint ("are adjacent") in G.

Usage

linegraph(graph, sep = "::")
Arguments

graph Undirected graph of class graph.
sep Separates the node names of G in the node names of the new graph L(G).

Value

graphNEL object.

Author(s)

Adrian Waddell and R. Wayne Oldford

See Also

navGraph, completegraph, newgraph

Examples

G <- completegraph(LETTERS[1:4])
LG <- linegraph(G, sep = "xx")
names<--methods  

Change the names of a NG_data object

Description

Change the names of a NG_data object. This end effectively changes the names of the data.frame wrapped in the NG_data object.

Methods

signature(x = "NG_data")

navGraph  

Start an navGraph session

Description

navGraph is an interactive data visualization program that lets the user explore high dimensional data by using graphs as a "road map". That is, navGraph connects a bullet on a graph to some user specified visualization of some data.

The package is discussed in detail in our vignette, see vignette("RnavGraph").

Usage

navGraph(data, graph = NULL, viz = NULL, settings = NULL)

Arguments

data  a single- or a list of objects generated by the ng_graph function. I.e. objects from the NG_data class.

two special cases are:
- either a navGrah handler to reinitialize a navGraph session.
- or the string tclreset to delete tk2d related settings and data.

graph  a single- or a list of objects generated by the ng_graph function. I.e. objects from the NG_data class.

viz  a single- or a list of visualization instruction objects generated for example by ng_2d or ng_2d_myplot function. I.e. objects from the NG_visualization class.

settings  a list of pailists. See details.
Details

The `RnavGraph` package comes with an extensive package vignette which we encourage to read. `navGraph` is capable to handle multiple graphs where each graph can be connected to multiple data displays.

`navGraph` needs to know the data to be explored, the graph whose node represent some low dimensional visualization of the data and some visualization instructions which connect the graph and data. Data, graph and visualization instruction have their own wrapper class provided by the `RnavGraph` package. See `ng_data`, `ng_graph` and for the visualization instructions `ng_Rd`. There are several shortcuts to start a `navgraph` session. All of them assume that the nodes of the graphs represent 2d scatterplots and the edges 3d rigid rotation or a 4d transition of one scatterplot into another. The easiest of them is to only pass by a data object created with the `ng_data` function. For the other ways consult the vignette.

The `settings` argument modifies the default appearance and interaction properties of the `navgraph` user interface. It must be a list of named (names are: color, interaction, display and `tk2d`) lists containing `tag=value` pairs. To get the possible `tag=value` options, study the class definitions of the `ColorSettings`, `InteractionSettings`, `DisplaySettings` and `Tk2dDisplay` classes with the `getClassDef` function. See the examples section below for an example.

Value

A `navGraph` handler which can be used to interface the `navGraph` session via the R prompt. Use `ng_get-methods` to get data from the function.

Use `ng_update` to update the `navGraph` handler with the current state of its associated `navGraph` session. If you have changed some attributes such as color or size in the `tk2d` display, you can get the new grouping with `ng_get(ng_get(nav,"data"),"group")` back, where `nav` is the `navGraph` handler.

Author(s)

Adrian Waddell and R. Wayne Oldford

See Also

`ng_data`, `ng_graph`, `ng_2d`, `ng_2d_myplot`, `scagNav`, `ng_get-methods`, `ng_set-methods`, `ng_update`, `ng_walk`, `ng_set_color<->`, `ng_get_color`, `ng_set_size<->`, `ng_get_size`

Examples

```r
## Define a NG_data object
ng.iris <- ng_data(name = "iris", data = iris[,1:4],
group = iris$Species,
labels = substr(iris$Species,1,2))

## Start navGraph
nav1 <- navGraph(ng.iris)

## navGraph session, data linked to previous session
```
newgraph <- navGraph(ng.iris, settings =
    list(color=list(background="steelblue", bullet="blue"),
         interaction=list(NSteps=1)))

## navGraph session, data is not linked
nav3 <- navGraph(ng.iris, settings=list(tk2d=list(linked=FALSE)))

---

newgraph  

Create a graph object of class graph

Description

Create a graph object of class graphNEL or graphAM. You might also use the graph creating facility provided by the graph package.

Usage

newgraph(nodeNames, mat, weights = NULL, directed = FALSE, isAdjacency = FALSE, ...)

Arguments

- **nodeName**
  Numeric or character string vector.

- **mat**
  Either an adjacency matrix or a from to matrix.

- **weights**
  Numeric weights for edges. Either in the same order as the from to matrix or as a square matrix, depending what one have chosen for the mat argument.

- **directed**
  Logical value for defining a directed or undirected graph.

- **isAdjacency**
  If argument mat is adjacency matrix.

- **...**
  Currently not needed.

Value

graphNEL or grapAM object.

Author(s)

Adrian Waddell and R. Wayne Oldford

See Also

- navGraph
- completegraph
- linegraph
### Examples

```r
## Using from to matrices
from <- c("A", "A", "C", "C")
to <- c("B", "C", "B", "D")
ftEmat <- cbind(from, to)

## note how the E node is added
G <- newgraph(nodeNames = LETTERS[1:5], mat = ftEmat)

## say you would like to add weights to the graph
weights <- c(2, 1, 3, 4)
G <- newgraph(nodeNames = LETTERS[1:5], mat = ftEmat, weights = weights)

## newgraph with adjacency matrix
adjM <- matrix(c(0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0), ncol = 4)
all(adjM == t(adjM)) ## is symmetric (undirected)
G <- newgraph(nodeNames = V, mat = adjM, isAdjacency = TRUE)

## if you use adjacency matrices, you can add a matrix with weights
adjM <- matrix(c(0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0), ncol = 4)
weightsM <- matrix(c(0, 0, 5, 0, 2, 0, 1, 3, 0, 0, 0, 0, 7, 0, 0), ncol = 4)
G <- newgraph(nodeNames = V, mat = adjM, weights = weightsM, directed = TRUE, isAdjacency = TRUE)
edgeData(G, attr = "weight")
```

---

### ng_2d

**Visualization instruction for 2d scatterplots on a tk2d display**

**Description**

Visualization instruction for a navGraph session that link the nodes of a navigation graph to 2d scatterplots and the edges to a 3d rigid rotation or a 4d transition.

The scatterplots will be displayed on a tk2d display, provided by the RnavGraph package. tk2d allows the user to display dots, glyphs, text and images at given x-y positions. Further interactivity such as selection, zooming, brushing, changing colors and changing sizes are provided by the tk2d display. tk2d displays can also link data between different navGraph sessions.

**Usage**

```r
ng_2d(data, graph, images = NULL, glyphs = NULL)
```

**Arguments**

- **data**
  - NG_data object.
- **graph**
  - NG_graph object.
- **images**
  - NG_image object. Order of the images must match the order of the data in the NG_data object.
- **glyphs**
  - Vector of character strings matching either the names or shortnames of the NG_data object.
Details

The text labels in the tk2d display are taken from the labels slot in the NG_data object. If the group slot of the NG_data object contains only colors from the colors function, objects get colored accordingly. Otherwise the group data gets mapped to some color key.

Value

NG_Viztk2d object that inherits from the NG_Visualization class.

Author(s)

Adrian Waddell and R. Wayne Oldford

See Also

navgraph, ng_data, ng_graph, ng_image_array_gray, ng_image_files, ng_get-methods, ng_set-methods, ng_2d_myplot

Examples

```r
## NG_data
ng.iris <- ng_data(name = "iris", data = iris[,1:4],
group = iris$Species,
labels = substr(iris$Species,1,2))

## NG_graph
G <- completegraph(shortnames(ng.iris))
LG <- linegraph(G, sep = "++")
ng.lg <- ng_graph("3d transition", LG, "++", "fruchtermanReingold")

## NG_image
## see the image demos: demo(package = "RnavGraph")

## Visualization instruction
viz1 <- ng_2d(ng.iris, ng.lg, glyphs = c("s.L","s.W","p.L","p.W"))
viz1
```

Description

Visualization instruction for a navGraph session that link the nodes of a navigation graph to 2d scatterplots and the edges to a 3d rigid rotation or a 4d transition. navGraph will call at any bullet state change a function the user defines in the global environment. The executed function should contain plot instruction and have any subset of these arguments
argument  description
x         x-coordinate
y         y-coordinate
group     group slot from NG_data object
labels    labels slot from NG_data object
order     order of points. In 3d rigid rotation, the order increases with the distance of a the point from the viewer.
from      node name the bullet moves from
to        node name the bullet moves to
percentage in between percentage of bullet
data       data name of NG_data object

Usage
ng_2d_myplot(data, graph, fnName, device = "base", scaled = TRUE)

Arguments
  data        NG_data object.
  graph       NG_graph object.
  fnName      Character sting of the function name defined in the global environment .GlobalEnv.
               If a function with the name fnName.init also exists, it gets called to initialize
               the plots.
  device      One of the following choices: "base", "grid", "ggplot2", "lattice" or "rgl".
  scaled      Logical. If TRUE, scaled x and y coordinates get passed to the function fnName
               which lie within the rectangle defined by (-1,-1) and (1,1).

Note
Note that the base graphic device (including grid, ggplot2 and the lattice device) don’t have double
buffering implemented in Linux and OSX but in Windows they have. Hence, in Windows you will
get smooth plots. But in OSX the plots only flush to the screen after a certain idle time from writing
data to the device. In Linux, the device gets updated after every plot command but the user will
experience a white flickering from redrawing the plot.

Author(s)
Adrian Waddell and R. Wayne Oldford

See Also
navGraph, ng_data, ng_graph, ng_get-methods, ng_set-methods, ng_2d

Examples
library(grid)

## NG_data
ng.iris <- ng_data(name = "iris", data = iris[,1:4],
ng_data

Create an NG_data object to be used by a navGraph session

Description

NG_data objects wrap the data to be explored and some meta data such as the data name, variable short names, the group identifier and text labels for each observation.

Usage

ng_data(name, data, shortnames = character(0), group = numeric(0), labels = character(0))
**Arguments**

- **name**: Character string containing no spaces. Note that for a single navGraph session, the names of all the NG_data objects passed by to navGraph must be unique.
- **data**: data.frame with only numeric variables.
- **shortnames**: If the variable names of the data argument are too long or contain spaces, shortnames can be specified.
- **group**: Vector of group identifiers for each observation.
  For the tk2d display, one can use color names (see `colors`) to map a certain color directly to the objects.
- **labels**: Factor- or character string vector with labels for each observation.

**Value**

NG_data object.

**Author(s)**

Adrian Waddell and R. Wayne Oldford

**See Also**

`navGraph`, `shortnames-methods`, `ng_get-methods`, `ng_set-methods`

**Examples**

```r
## minimal example
ng.iris <- ng_data(name = "iris", data = iris[,1:4])
gg.iris ## see output

## full specification
ng.iris <- ng_data(name = "iris", data = iris[,1:4],
  group = iris$Species,
  labels = substr(iris$Species,1,2))

## see shortnames
shortnames(ng.iris)
## change shortnames
shortnames(ng.iris) <- c("a","b","c","d")
## see variable names
names(ng.iris)
## change variable names
names(ng.iris) <- LETTERS[1:4]

## ng_get
ng_get(ng.iris)
gg_get(ng.iris,"group")

## ng_set
```
ng_set(ng.iris)
ng_set(ng.iris,"group") <- iris$Species

---

**ng_get**

Extract data from a NG_data, NG_graph, NG_path or navgraph handler object.

**Description**

Extract data from objects from some of the in RnavGraph specifically defined classes.

**Usage**

`ng_get(obj, what = NULL, ...)`

**Arguments**

- `obj`: Either a navgraph handler, NG_data, NG_path, NG_graph object.
- `what`: String of what should be extracted from the object. You can get a list of possible accessible objects by not specifying the what argument.
- `...`: Currently not used.

**Value**

Object that was requested with what.

**Author(s)**

Adrian Waddell and Wayne Oldford

**See Also**

`navGraph`, `ng_get-methods`, `ng_set`, `ng_set-methods`, `ng_set_color<->`, `ng_get_color`, `ng_set_size<->`, `ng_get_size`

**Examples**

```r
## Define a NG_data object
ng.iris <- ng_data(name = "iris", data = iris[,1:4])

## start a navGraph session
nav <- navGraph(ng.iris)

## See what you can extract from the navGraph handler
ng_get(nav)

## get group
ng_get(ng_get(nav,"data"),"group")
```
ng_get-methods

Access data from a NG_... object

Description
Access data in the wrapper classes in navGraph or the navGraph handler. The usage is either ng_get(object) or ng_get(object,"what"). See in the examples of ng_update.

Methods
signature(obj = "NavGraph_handler")
signature(obj = "NG_data")
signature(obj = "NG_graph")
signature(obj = "NG_path")

ng_get_color
Extract point coloring from an active navGraph session for a particular data set.

Description
If you use the brush and coloring tool in navGraph, you can retrieve your coloring information with this function as long the navGraph session is still active.

Usage
ng_get_color(obj, dataName)

Arguments
obj A navgraph handler of an active navGraph session.
dataName String of the data name. If not specified and only one data set is being used in the navGraph session, it will default to this data. Otherwise, if multiple data sets are being used in a navGraph session, the function will list the name of these data sets and ask you to specify one.

Value
Vector with color values for each point.

Author(s)
Adrian Waddell and Wayne Oldford
See Also

**ng_set_size**

Examples

```r
## Define a NG_data object
g_data <- ng_data(name = "IrisData", data = iris[,1:4])

## start a navGraph session
nav <- navGraph(g_data)

## get colors from active navGraph session
cols <- ng_get_color(nav, "IrisData")
```

**ng_get_size**

**Description**

If you use the brush and resizing tool in navGraph, you can retrieve your point sizes information with this function as long the navGraph session is still active.

**Usage**

```r
ng_get_size(obj, dataName)
```

**Arguments**

- **obj**: A navgraph handler of an active navGraph session.
- **dataName**: String of the data name. If not specified and only one data set is being used in the navGraph session, it will default to this data. Otherwise, if multiple data sets are being used in a navGraph session, the function will list the name of these data sets and ask you to specify one.

**Value**

Numerical vector with size values for each point.

**Author(s)**

Adrian Waddell and Wayne Oldford

**See Also**

**navGraph, ng_set_size**, **ng_get_color, ng_set_color**
Examples

```r
## Define a NG_data object
ng.iris <- ng_data(name = "IrisData", data = iris[,1:4])

## start a navGraph session
nav <- navGraph(ng.iris)

## get sizes from active navGraph session
sizes <- ng_get_size(nav, "IrisData")
```

---

### ng_graph

Create an NG_data object to be used by a navGraph session

---

### Description

NG_graph objects wrap the transition graphs and additional information such as graph name and graph layout.

The node names of the graph split by the chosen sep character string must match either the variable names- or the short names of the NG_data object.

### Usage

```r
ng_graph(name, graph, sep = ":", layout = "circle")
```

### Arguments

- `name`  
  Name of graph. This name will be displayed in the pull down menu within a navGraph session.

- `graph`  
  Undirected graph objects of class graph. See the package `graph`.

- `sep`  
  Node names represent a set of variables whose name are separated by the character string `sep` (containing no spaces).

- `layout`  
  One of the following strings: "circle", "kamadaKawaiSpring", "fruchterman-Reingold" or "random".

### Details

The number of sep occurrences in each node name of the graph must be the same.

### Value

NG_graph object.

### Author(s)

Adrian Waddell and R. Wayne Oldford
See Also

navGraph, newgraph, completegraph, linegraph, complement, ng_get-methods, ng_set-methods

Examples

```r
G <- completegraph(LETTERS[1:4])
LG <- linegraph(G, sep = "++")

ng.lg <- ng_graph("3d transition", LG, "++", "fruchtermanReingold")
plot(ng.lg)

## If you have the Rgraphviz package working, plot graph object of class graph
## Not run: library(Rgraphviz)
## Not run: plot(LG)
```

---

**ng_image_array_gray**  
*Convert a matrix or data.frame of image data into a NG_image object.*

**Description**

NG_image objects are needed to plot images in the tk2d device.

*ng_image_array_gray* imports gray scale images that are saved in a matrix structure, that is every row or column contains the pixel data of an image.

See the "alpha_letter", "digits", "faces", "frey" and "umist_faces" demos (demo(package="RnavGraph")) for examples.

**Usage**

```r
ng_image_array_gray(name, image_data, width, height, img_in_row = TRUE, invert = FALSE, rotate = 0)
```

**Arguments**

- `name`: Character string.
- `image_data`: data.frame of matrix with image data. Image information must be stored as a gray scale value (0-255) within a single row or column.
- `width`: Pixel width of image.
- `height`: Pixel height of image.
- `img_in_row`: Logical. Single image stored in a row or column in `image_data`.
- `invert`: For invert=FALSE 0 = wht, for invert=TRUE 0=black.
- `rotate`: Rotate the image: 0, 90, 180 or 270 degree.
**ng_image_files**

**Value**

NG_image object

**Author(s)**

Adrian Waddell and R. Wayne oldford

**See Also**

navGraph, ng_image_files, ng_2d

**Examples**

```r
Img <- matrix(c(0,0,0,255,255,255,
                0,0,255,255,255,
                0,0,255,255,255,
                255,255,255,0,0,0,
                255,255,255,0,0,0,
                255,255,255,0,0,0),
               byrow = TRUE, ncol=6)

ng.img <- ng_image_array_gray("Test",
                               cbind(as.vector(Img),as.vector(Img)),6,6,FALSE)

## See demos and vignette
```

**Description**

NG_image objects are needed to plot images in the tk2d device.

Supported file types are those supported by the tcl Img extension. If your R can not access the Img extension (currently under Windows) you can not import such files. However if you know tcl and R well enough, you might create your own NG_image object as shown in the images_iris demo.

See the "files_aloi" and "images_iris" demos (demo(package="RnavGraph")) for examples.

**Usage**

```r
ng_image_files(name, paths)
```

**Arguments**

- `name` Character string.
- `paths` A vector of image paths.
**Value**

NG_image object.

**Author(s)**

Adrian Waddell and R. Wayne Oldford.

**See Also**

`navGraph, ng_image_array_gray, ng_2d`

**Examples**

```r
## See demos and vignette
```

---

**Description**

Show what can be changed in objects from some of the in RnavGraph specifically defined classes.

**Usage**

`ng_set(object)`

**Arguments**

- `object` either a: navgraph handler, NG_data, NG_path, NG_graph object.

**Value**

No return value. Only a string gets printed onto the command prompt of all possible data that can be modified within the object.

**Note**

Using `ng_set` is risky because we don’t guarantee that the objects gets updated correctly. We recommend to just re-create the object.

**Author(s)**

Adrian Waddell and Wayne Oldford

**See Also**

`navGraph, ng_set-methods, ng_get, ng_get-methods, ng_set_color<-, ng_get_color, ng_set_size<-, ng_get_size`
**Description**

Care must be taken when using the `ng_set` method, as one might create inconsistency in some objects.

Its use is `ng_set(object)` to see what you can modify and `ng_set(object, "what") <- ....` See `ng_set<--methods`.

**Methods**

```r
signature(object = "NG_data")
signature(object = "NG_graph")
```

**Description**

Modifies data in some of the in RnavGraph specifically defined classes.

**Usage**

```r
ng_set(object, what) <- value
```

**Arguments**

- **object**: Either a: navgraph handler, NG_data, NG_path, NG_graph object.
- **what**: String of what should be modified within the object.
- **value**: Replacement value.

**Details**

Using `ng_set` is risky because we don’t guarantee that the objects gets updated correctly. We recommend to just re-create the object.

**Author(s)**

Adrian Waddell and Wayne Oldford

**See Also**

`navGraph`, `ng_set-methods`, `ng_get`, `ng_get-methods`, `ng_set_color<->`, `ng_get_color`, `ng_set_size<->`, `ng_get_size`
Description

~~ Methods for function ng_set<- ~~

Methods

signature(object = "NavGraph_handler")
signature(object = "NG_data")
signature(object = "NG_graph")

ng_set_color<-  Change colors of data points in an active navGraph session

Description

Specify new colors for each point for an active navGraph session.

Usage

ng_set_color(obj, dataName) <- value

Arguments

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>A navgraph handler of an active navGraph session.</td>
</tr>
<tr>
<td>dataName</td>
<td>String of the data name. If not specified and only one data set is being used in the navGraph session, it will default to this data. Otherwise, if multiple data sets are being used in a navGraph session, the function will list the name of these data sets and ask you to specify one.</td>
</tr>
<tr>
<td>value</td>
<td>Replacement vector or single value specifying valid colors.</td>
</tr>
</tbody>
</table>

Author(s)

Adrian Waddell and Wayne Oldford

See Also

navGraph, ng_get_color, ng_set_size<-, ng_get_size
Examples

```r
## Define a NG_data object
ng.iris <- ng_data(name = "IrisData", data = iris[,1:4])

## start a navGraph session
nav <- navGraph(ng.iris)

## set all point to red
ng_set_color(nav, "IrisData") <- 'red'
```

Description

Specify new colors for each point for an active navGraph session.

Methods

```r
signature(obj = "NavGraph_handler")
```

Description

Specify new sizes for each point for an active navGraph session.

Usage

```r
ng_set_size(obj, datumName) <- value
```

Arguments

- **obj** A navgraph handler of an active navGraph session.
- **dataName** String of the data name. If not specified and only one data set is being used in the navGraph session, it will default to this data. Otherwise, if multiple data sets are being used in a navGraph session, the function will list the name of these data sets and ask you to specify one.
- **value** Replacement vector or single value specifying the sizes (>0) of points.

Author(s)

Adrian Waddell and Wayne Oldford
See Also

`navGraph, ng_get_size, ng_set_color<-, ng_get_color`

Examples

```r
## Define a NG_data object
ng.iris <- ng_data(name = "IrisData", data = iris[,1:4])

## start a navGraph session
nav <- navGraph(ng.iris)

## set point sizes
ng_set_size(nav,'IrisData') <- sample(1:7, replace=TRUE, 150)
```

---

**ng_set_size<->methods**  
*Change sizes of data points in an active navGraph session*

**Description**

Specify new sizes for each point for an active navGraph session.

**Methods**

```r
signature(obj = "NavGraph_handler")
```

---

**ng_update**  
*Synchronize a navGraph handler with a running navGraph session*

**Description**

The navGraph handler can be used to interact with a running navGraph session via the R prompt. For the tk2d display one can change color and size of the data points. You can retrieve this information by updating the navGraph handler and reading the group attribute of the data object back. See the examples.

**Usage**

```r
ng_update(nghandler)
```

**Arguments**

```r
nghandler  navGraph handler of a running navGraph session.
```

**Value**

updated navGraph handler.
Author(s)
Adrian Waddell and R. Wayne Oldford

See Also

navgraph, ng_walk, ng_get-methods

Examples

```r
## Define a NG_data object
ng.iris <- ng_data(name = "iris", data = iris[,1:4])

## start a navGraph session
nav <- navGraph(ng.iris)

## modify colors of points

## update navGraph handler
nav <- ng_update(nav)
nav # show method

## get information form navGraph handler
ng_get(nav)

## get group attribute back
ng_get(ng_get(nav,"data"),"group")
```

---

NG_Visualization-class

Class "NG_Visualization"

Description

Used when custom visualization instructions are defined using initializeViz, updateViz and closeViz.

See package vignette for more details.

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

graph: Graph name in slot name of NG_graph object.
data: Data name in slot name of NG_data object.
from: Start from node. Not used anymore.
to: Start to node. Not used anymore.
varList: Vector with all variable names that are used in graph.
Methods

show signature(object = "NG_Visualization"): ...

Note
Read more about the use of this class in the package vignette.

Author(s)
Adrian Waddell and Wayne Oldford

See Also
initializeViz-methods

Examples

dshowClass("NG_Visualization")

ng_walk
Walk a path on the current graph in a navGraph session

Description
If you create a path outside the navGraph session, you can let navGraph walk you path.
A path has to be a sequence of node names of adjoining nodes in the current shown graph.
If a path has been walked through, it gets added to the activePath in the path tool of the running
navGraph session.

Usage

ng_walk(nghandler, path)

Arguments

nghandler navGraph handler of a running navGraph session.
path Vector of node names that are adjoining in the current graph. A single character
string with the node names separated by a space also works.

Author(s)
Adrian Waddell and R. Wayne Oldford

See Also

navGraph
Examples

```r
## Define a NG_data object
ng.iris <- ng_data(name = "iris", data = iris[,1:4],

## start a navGraph session
nav <- navGraph(ng.iris)

## Find linegraph
LG <- linegraph(completegraph(shortnames(ng.iris)))

## find a path
library(PairViz)
path = eulerian(LG)

## walk the path
ng_walk(nav, path)
```

olive

Fatty Acid Composition of Italian Olive Oils

Description

This data set records the percentage composition of 8 fatty acids (palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic, eicosenoic) found in the lipid fraction of 572 Italian olive oils. The oils are samples taken from three Italian regions varying number of areas within each region. The regions and their areas are recorded as shown in the following table:

<table>
<thead>
<tr>
<th>Region</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td>North-Apulia, South-Apulia, Calabria, Sicily</td>
</tr>
<tr>
<td>South</td>
<td>East-Liguria, West-Liguria, Umbria</td>
</tr>
<tr>
<td>Sardinia</td>
<td>Coastal-Sardinia, Inland-Sardinia</td>
</tr>
</tbody>
</table>

Usage

olive

Format

A data frame containing 572 cases and 10 variates.

References

plot-methods  

Plot the graph saved in a NG_data object.

Description

Once the graph is within a NG_graph object, it has a layout associated. Plot plots the graph with this layout.

Methods

signature(x = "NG_data")

scagEdgeWeights  

Create a from-to edge matrix with the scagnostic weights

Description

Create a from-to edge matrix with the scagnostic weights.

Usage

scagEdgeWeights(data, 
   scags = c("Clumpy", "NotClumpy", "Monotonic", "NotMonotonic",  
   "Convex", "NotConvex", "Stringy", "NotStringy",  
   "Skinny", "NotSkinny", "Outlying", "NotOutlying",  
   "Sparse", "NotSparse", "Striated", "NotStriated",  
   "Skewed", "NotSkewed"),  
   combineFn = NULL)

Arguments

data  
   object to calculate scagnostics on: NG_data, a vector, a matrix or a data.frame.

scags  
   Single element or a subset of (with possible a "Not" preceding):  
   "Outlying", "Skewed", "Clumpy", "Sparse", "Striated",  
   "Convex", "Skinny", "Stringy", "Monotonic"

combineFn  
   Must be a function that takes in a vector of length scags and returns a single value. This return value comprises the new weights of the nodes get selected from.

Value

a named list with fromToEdgeMatrix being a matrix and nodeNames being a vector.

Author(s)

Adrian Waddell and R. Wayne Oldford
scagGraph

See Also

`navGraph`, `scagNav`, `scagGraph`

Examples

data(olive)
ng.olive <- ng_data(name = "Olive",
data = olive[,,-c(1,2)],
shortnames = c("p1", "p2", "s", "ol", "l1", "l2", "a", "e"),
group = as.numeric(olive$Area)+1
)

edgewts <- scagEdgeWeights(data = ng.olive,
scags = c("Clumpy", "Skinny"),
combineFn = max)
edgewts$fromToEdgeMatrix[1:3,]

edgewts <- scagEdgeWeights(data = ng.olive,
scags = c("Clumpy", "Skinny"),
combineFn = function(x){
2*x[1] + 3*x[2]
})
edgewts$fromToEdgeMatrix[1:3,]

scagGraph

Create a list of graphs, given the scagnostics edge weights

Description

scagGraph is useful to turn the output of `scagEdgeWeights` into a list of graphs.

Usage

`scagGraph(edgeWeights, topFrac = 0.2)`

Arguments

- `edgeWeights`: List returned by the `scagEdgeWeights` function.
- `topFrac`: Keep the nodes with the `topFrac` fraction of the scagnostic weights.

Value

- graph object or a list of graph objects.

Author(s)

Adrian Waddell and R. Wayne Oldford
scagnav

Start a navGraph session and filter the navigation graph's node according to some scagnostic measures

Description


With scagnav one can start a navGraph session that constructs a navigation graph that shows certain properties most.

See the vignette for more examples.

Usage

scagnav(data,
        scags = c("Clumpy", "NotClumpy", "Monotonic", "NotMonotonic",
                   "Convex", "NotConvex", "Stringy", "NotStringy",
                   "Skinny", "NotSkinny", "Outlying", "NotOutlying",
                   "Sparse", "NotSparse", "Striated", "NotStriated",
                   "Skewed", "NotSkewed"),
        topFrac = 0.2, combineFn = NULL,
        settings = NULL, glyphs = NULL,
Arguments

data a single- or a list of objects generated by the \texttt{ng\_graph} function. I.e. objects from the \texttt{NG\_data} class.


topFrac Keep the nodes with the topFrac fraction of the scagnostic weights.

combineFn Must be a function that takes in a vector of length scags and returns a single value. This return value comprises the new weights of the nodes get selected from.

settings a list of pailists. See the \texttt{navGraph} documentation.

glyphs Vector of character strings matching either the \texttt{names} or \texttt{shortnames} of the \texttt{NG\_data} object.

images \texttt{NG\_image} object. Order of the images must match the order of the data in the \texttt{NG\_data} object.

sep Node names represent a set of variables whose name are separated by the character string \texttt{sep} (containing no spaces).

layout One of the following strings: "circle", "kamadaKawaiSpring", "fruchterman-Reingold" or "random".

Value

\texttt{navGraph} handler.

Author(s)

Adrian Waddell and R. Wayne Oldford

See Also

\texttt{navGraph}, \texttt{scagEdgeWeights}, \texttt{scagGraph}, \texttt{ng\_data}

Examples

```r
## Define a NG\_data object
data(olive)
ng.\texttt{olive} \leftarrow \texttt{ng\_data(name = "Olive"},
data = \texttt{olive[,,-c(1,2)]},
shortnames = \texttt{c("p1","p2","s","ol","l1","l2","a","e")},
\texttt{group} = \texttt{as.numeric(olive}\_\texttt{\$Area}+1

\texttt{)
nav \leftarrow \texttt{scagNav(data = ng.\texttt{olive},}
\texttt{scags = c("Skinny", "Sparse", "NotConvex")},
```
shortnames

Returns the shortnames of a NG_data object.

Description

Shortnames are useful to reduce the node names of navigation graphs. The node names of navigation graphs are either linked with the names or the shortnames of the NG_data object.

Usage

shortnames(x)

Arguments

x

NG_data object.

Value

Vector of strings.

Author(s)

Adrian Waddell and Wayne Oldford

See Also

navGraph, ng_data

Examples

## Define a NG_data object
ng.iris <- ng_data(name = "iris", data = iris[,1:4])

## Display shortnames
shortnames(ng.iris) ## no shortnames defined yet

## Modify the shortnames
shortnames(ng.iris) <- c("sL","sW","pL","pW")
## Display updated shortnames

```r
shortnames(ng.iris) # no shortnames defined yet
```

---

### shortnames-methods

#### Show the shortnames of a NG_data object

**Description**

The variable names coded in the node names of a graph in a `navGraph` session will be matched either with the names or shortnames of a NG_data object.

**Methods**

```r
signature(x = "NG_data")
```

---

#### shortnames<- Modify the shortnames of a NG_data object.

**Description**

Shortnames are useful to reduce the node names of navigation graphs. The node names of navigation graphs are either linked with the names or the shortnames of the NG_data object.

**Usage**

```r
shortnames(x) <- value
```

**Arguments**

- `x` NG_data object.
- `value` Vector of stings.

**Author(s)**

Adrian Waddell and Wayne Oldford

**See Also**

`navGraph`, `ng_data`
Examples

```r
## Define a NG_data object
g.iris <- ng_data(name = "iris", data = iris[,1:4])

## Display shortnames
shortnames(g.iris) ## no shortnames defined yet

## Modify the shortnames
shortnames(g.iris) <- c("sL","sW","pL","pW")

## Display updated shortnames
shortnames(g.iris) ## no shortnames defined yet
```

shortnames<-methods  
*Change the shortnames of a NG_data object*

Description

Change the shortnames of a NG_data object.

Note that the shortnames of a NG_data object can not contain any spaces.

Methods

```r
signature(x = "NG_data")
```

updateViz  
*Method that gets called when bullet state is changed on a graph that is connected with the particular visualization instructions*

Description

See the vignette for more details.

Usage

```r
updateViz(viz,ngEnv)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>viz</code></td>
<td>Object of class NG_Visualization</td>
</tr>
<tr>
<td><code>ngEnv</code></td>
<td>Environment of navGraph session.</td>
</tr>
</tbody>
</table>

Value

Object of NG_Visualization (viz from argument)
Author(s)
Adrian Waddell and Wayne Oldford

See Also
initializeViz, closeViz

Examples
## See vignette

<table>
<thead>
<tr>
<th>updateViz-methods</th>
<th>Initialization of a new Display</th>
</tr>
</thead>
</table>

Description
If custom visualization class (NG_Visualization) has been defined. This is the method that gets called if the bullet changes its state.
See package vignette for more detail.

Methods
signature(obj = "NG_Visualization")
Index

* Topic `textasciitilde` `textasciitilde`
other possible keyword(s)

  `ng_set<--methods`, 24

* Topic `classes`
  `NG_Visualization-class`, 27

* Topic `classif`
  `navGraph`, 8
  `RnavGraph-package`, 3

* Topic `datasets`
  `olive`, 29

* Topic `dynamic`
  `navGraph`, 8
  `ng_2d`, 11
  `RnavGraph-package`, 3

* Topic `graphs`
  `linegraph`, 6
  `navGraph`, 8
  `newgraph`, 10
  `ng_graph`, 19
  `RnavGraph-package`, 3

* Topic `graph`
  `completeGraph`, 5

* Topic `methods`
  `closeViz-methods`, 4
  `initializeViz-methods`, 6
  `linegraph-methods`, 7
  `names-methods`, 7
  `names<--methods`, 8
  `ng_get-methods`, 17
  `ng_set-methods`, 23
  `ng_set<--methods`, 24
  `ng_set_color<--methods`, 25
  `ng_set_size<--methods`, 26
  `plot-methods`, 30
  `shortnames-methods`, 35
  `shortnames<--methods`, 36
  `updateViz-methods`, 37

* Topic `multivariate`
  `navGraph`, 8

  `RnavGraph-package`, 3
  `closeViz`, 4, 6, 37
  `closeViz,NG_Visualization-method`
  `(closeViz-methods), 4
  `closeViz,NG_Viz2DAxis-method`
  `(closeViz-methods), 4
  `closeViz,NG_Viztk2d-method`
  `(closeViz-methods), 4
  `closeViz-methods`, 4
  `colors`, 15
  `complement`, 20
  `completeGraph`, 5, 7, 10, 20
  `linegraph`, 5, 6, 10, 20
  `linegraph,graph-method`
  `(linegraph-methods), 7`
  `linegraph-methods`, 7
  `names,NG_data-method (names-methods), 7`
  `names-methods`, 7
  `names<-,NG_data,ANY-method`
  `(names<--methods), 8`
  `names<--methods`, 8
  `newGraph`, 3, 5, 7, 8, 10, 12, 13, 15, 16, 18, 20–24, 26–28, 31–35
  `newgraph`, 5, 7, 10, 20
  `ng_2d`, 8, 9, 11, 13, 21, 22
  `ng_2d_myplot`, 8, 9, 12, 12
  `ng_data`, 9, 12, 13, 14, 33–35
  `ng_get`, 16, 22, 23
ng_getNavGraph_handler-method
  (ng_get-methods), 17
ng_get,NG_data-method (ng_get-methods), 17
ng_get,NG_graph-method
  (ng_get-methods), 17
ng_get,NG_path-method (ng_get-methods), 17
ng_get-methods, 17
ng_get_color, 9, 16, 17, 18, 22–24, 26
ng_graph, 8, 9, 12, 13, 19, 33
ng_image_array_gray, 12, 20, 22
ng_image_files, 12, 21, 21
ng_set, 16, 22
ng_set,NG_data-method (ng_set-methods), 23
ng_set,NG_graph-method
  (ng_set-methods), 23
ng_set-methods, 23
ng_set<-, 23
ng_set<-,NavGraph_handler-method
  (ng_set<--methods), 24
ng_set<-,NG_data-method
  (ng_set<--methods), 24
ng_set<-,NG_graph-method
  (ng_set<--methods), 24
ng_set<--methods, 24
ng_set_color<-, 24
ng_set_color<-,NavGraph_handler-method
  (ng_set_color<--methods), 25
ng_set_color<--methods, 25
ng_set_size<-, 25
ng_set_size<-,NavGraph_handler-method
  (ng_set_size<--methods), 26
ng_set_size<--methods, 26
ng_update, 9, 17, 26
NG_Visualization-class, 27
ng_walk, 9, 27, 28
olive, 29
plot,NG_graph,ANY-method
  (plot-methods), 30
plot-methods, 30
RnavGraph (RnavGraph-package), 3
RnavGraph-package, 3
scagEdgeWeights, 30, 31–33
scagGraph, 31, 31, 33
scagNav, 9, 31, 32, 32
shortnames, 34
shortnames,NG_data-method
  (shortnames-methods), 35
shortnames-methods, 35
shortnames<-, 35
shortnames<-,NG_data-method
  (shortnames<--methods), 36
shortnames<--methods, 36
updateViz, 4, 6, 36
updateViz,NG_Visualization-method
  (updateViz-methods), 37
updateViz,NG_Viz2DAxis-method
  (updateViz-methods), 37
updateViz,NG_Viztk2d-method
  (updateViz-methods), 37
updateViz-methods, 37

updateViz-methods, 37