

Package ‘plotGoogleMaps’

July 2, 2014

Type Package

Title Plot spatial or spatio-temporal data over Google Maps

Version 2.1

Date 2012-04-02

Author Milan Kilibarda

Maintainer Milan Kilibarda <kili@grf.bg.ac.rs>

Description This package provides a interactive plot device for handling the geographic data for web browsers. It is designed for the automatic creation of web maps as a combination of users' data and Google Maps layers.

Depends R (>= 2.15.0), sp, spacetime

Imports rgdal, maptools,lattice, raster

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2014-06-19 15:42:40

R topics documented:

bubbleGoogleMaps	2
bubbleSP	5
ellipseGoogleMaps	7
elSPDF	10
iconlabels	11
mcGoogleMaps	13
pieSP	16
plotGoogleMaps	18
PolyCol	21

segmentGoogleMaps	23
stfdfGoogleMaps	26
stplotGoogleMaps	28
vectorsSP	30

Index	32
--------------	-----------

bubbleGoogleMaps	<i>Create a bubble plot of spatial data on Google Maps</i>
------------------	--

Description

Plot htm output with Google Maps API in form of bubble plot of spatial data, with options for bicoural residual plots. Ready to use as local htm file or into your own website.

Usage

```

bubbleGoogleMaps(SP,
                  filename = '',
                  zcol = 1,
                  max.radius = 100,
                  key.entries = quantile(SP@data[, zcol], (1:5)/5),
                  do.sqrt = TRUE,
                  add = FALSE,
                  previousMap = NULL,
                  colPalette = NULL,
                  strokeColor = "",
                  strokeOpacity = 1,
                  fillOpacity=0.7,
                  strokeWeight = 1,
                  geodesic = TRUE,
                  clickable = TRUE,
                  zIndex = 'null',
                  shape = 'c',
                  map.width="100%",
                  map.height="100%",
                  layerName="",
                  control.width="100%",
                  control.height="100%",
                  zoom = 15,
                  fitBounds = TRUE,
                  mapTypeId = 'HYBRID',
                  disableDoubleClickZoom = FALSE,
                  draggable = TRUE,
                  keyboardShortcuts = TRUE,
                  mapTypeControlOptions = 'DEFAULT',
                  navigationControl = TRUE,
                  navigationControlOptions = 'DEFAULT',

```

```

scaleControlOptions = 'STANDARD',
noClear = FALSE,
scrollwheel = TRUE,
streetViewControl = FALSE,
legend=TRUE,
control=TRUE,
InfoWindowControl=list(map=map, event="click",
position="event.latLng",disableAutoPan=FALSE,
maxWidth=330,pixelOffset="null",
                                zIndex="null") ,
map="map",
mapCanvas="map_canvas",
css = "",
api="https://maps.google.com/maps/api/js?sensor=false",
openMap=TRUE)

```

Arguments

SP	object of SpatialPointsDataFrame-class with associated coordinate reference systems
filename	the name of the output htm or html file. If filename is not defined the function creates temporary file.
zcol	variable column name, or column number after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column
max.radius	value for largest circle (the plotting symbols) in metre, circumcircle of triangle or quadrangle (square)
key.entries	the values that will be plotted in the key; by default the five quantiles min, q.25, median q.75, max
do.sqrt	logical; if TRUE the plotting symbol area (sqrt(diameter)) is proportional to the value of the z-variable; if FALSE, the symbol size (diameter) is proportional to the z-variable
add	logical; if TRUE the result of the function will be a list stored as variable in the R. It is possible to combine more layers in the one plot, previously saved output from plotGoogleMaps should be given in the previousMap attribute.
previousMap	previously saved variable from plotGoogleMaps when attribute add is TRUE. It is solution for combining more than one layers in the one plot.
colPalette	colours to be used to fill plotting symbols; numeric vector of same size like key.entries
strokeColor	the color to draw the border of circle (the plotting symbols)
strokeOpacity	the stroke opacity between 0.0 and 1.0
fillOpacity	the fill opacity between 0.0 and 1.0
strokeWeight	the stroke width in pixels
geodesic	render each edge as a geodesic (a segment of a 'great circle')
clickable	indicates whether this plotting symbol handles click events

<code>zIndex</code>	the <code>zIndex</code> compared to other polygons
<code>shape</code>	the shape for plotting symbol; 'c' for circle; 't' for triangle; 'q' for square
<code>map.width</code>	the width of the map. Can be in percent or in pixels (px)
<code>map.height</code>	the height of the map. Can be in percent or in pixels (px)
<code>layerName</code>	the name of the layer. Layer name appears on the check box for switching layer of and on in output htm.
<code>control.width</code>	the width of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>control.height</code>	the height of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>zoom</code>	the initial Map zoom level. Zoom levels between 0 (the lowest zoom level, in which the entire world can be seen on one map) to 19 (the highest zoom level, down to individual buildings) are possible within the normal maps view. Argument <code>fitBounds</code> should be FALSE.
<code>fitBounds</code>	sets the maps to fit to the boundary box values of <code>sp</code> object
<code>mapTypeId</code>	the initial Map <code>mapTypeId</code> . 'HYBRID' - map type displays a transparent layer of major streets on satellite images. 'ROADMAP' - map type displays a normal street map. 'SATELLITE' - map type displays satellite images. 'TERRAIN' - map type displays maps with physical features such as terrain and vegetation.
<code>disableDoubleClickZoom</code>	enables or disables zoom and center on double click.
<code>draggable</code>	if FALSE, prevents the map from being dragged
<code>keyboardShortcuts</code>	if FALSE, prevents the map from being controlled by the keyboard.
<code>mapTypeControlOptions</code>	the initial display options for the Map type control. 'DEFAULT' - uses the default map type control. 'DROPDOWN_MENU' - a dropdown menu for the screen realestate conscious. 'HORIZONTAL_BAR' - the standard horizontal radio buttons bar.
<code>navigationControl</code>	enabled or disabled state of the navigation control
<code>navigationControlOptions</code>	the initial display options for the navigation control. 'ANDROID' - the small zoom control similar to the one used by the native Maps application on Android. 'DEFAULT' - the default navigation control. The control which DEFAULT maps to will vary according to map size and other factors. It may change in future versions of the API. 'SMALL' - the small, zoom only control. 'ZOOM_PAN' - the larger control, with the zoom slider and pan directional pad.
<code>scaleControlOptions</code>	the initial display options for the scale control.
<code>noClear</code>	if TRUE, do not clear the contents of the Map div.
<code>scrollwheel</code>	if FALSE, disables scrollwheel zooming on the map.
<code>streetViewControl</code>	the initial enabled or disabled state of the Street View pegman control

legend	if FALSE, disables legend.
control	if FALSE, disables control.
InfoWindowControl	see https://developers.google.com/maps/documentation/javascript/reference#InfoWindowOptions
map	name of map object in JavaScript
mapCanvas	name of map DIV element
css	CSS for mapCanvas and cBoxes
api	see https://developers.google.com/maps/documentation/javascript/tutorial
openMap	if TRUE map is browsed by default browser

Value

This function returns a list with HTML, JavaScript, Google Maps API key and else what is necessary for the final web map. The second output is htm file ready to use stored in working direcoty.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[plotGoogleMaps](#), [ellipseGoogleMaps](#), [segmentGoogleMaps](#)

Examples

```
# Data preparation
data(meuse)
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS('+init=epsg:28992')

m<-bubbleGoogleMaps(meuse,zcol='zinc')

m<-bubbleGoogleMaps(meuse,zcol='cadmium',layerName='Bubble plot - meuse',
                    colPalette=terrain.colors(5),strokeColor='')
```

bubbleSP

Create bubble data in form of SpatialPolygonsDataFrame.

Description

Create bubble data in form of SpatialPolygonsDataFrame based on one attribute.

Usage

```
bubbleSP(SPDF,
         zcol=1,
         scale_e=1,
         max.radius=100,
         key.entries = quantile(SPDF@data[,zcol],(1:5)/5, na.rm=TRUE),
         do.sqrt = TRUE,
         radius.vector=NULL)
```

Arguments

SPDF	object of SpatialPointsDataFrame-class with associated coordinate reference systems
zcol	variable column names, or column numbers after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column
scale_e	scale factor for radiuses
max.radius	value for maximum radius in meter
key.entries	the values that will be plotted in the key; by default the five quantiles min, q.25, median q.75, max
do.sqrt	logical; if TRUE the plotting symbol area (sqrt(diameter)) is proportional to the value of the z-variable; if FALSE, the symbol size (diameter) is proportional to the z-variable
radius.vector	vector of predefined radius for each point

Value

The function provide [SpatialPolygonsDataFrame](#) based on one attribute, ready to use for plot-GoogleMaps or spplot.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[pieSP](#), [elSPDF](#), [bubbleGoogleMaps](#)

Examples

```
# Data preparation
wind<-data.frame( Station=c('Zlatibor' , 'Sjenica' , 'Pozega' ,
                          'Kraljevo' , 'Kopaonik' , 'Kursumlija' , 'Krusevac'),
                 Lon=c(19.708418 , 19.999311 , 20.029322,
                      20.724012, 20.776111, 21.279513, 21.307998 ),
                 Lat=c(43.729755 , 43.267913 ,43.84859 ,
                      43.726812,43.2725,43.137053 ,43.574188),
                 Temp=c(7, 7,5,6,1,6,9),
                 Wind_spd=c(3 , 6,1 ,1,7 ,1 ,2),
```

```

Wind_dir=c(45, 162, 138 ,265,335 ,180 ,270) )

coordinates(wind)<- ~Lon+Lat
wind@proj4string=CRS('+proj=longlat +datum=WGS84')

b_temp=bubbleSP(wind,zcol=c('Temp'),max.radius=10000, do.sqrt=FALSE)

pal<-colorRampPalette(c( "#FFF7BC", "#FEC44F", "#D95F0E"), space = "Lab")
spplot(b_temp,'Temp', col.regions=pal(5) )
m=plotGoogleMaps(b_temp,zcol='Temp', colPalette=pal(5),strokeWeight=0.5,strokeColor='black')

# plotKML(b_temp, col='Temp')

```

ellipseGoogleMaps	<i>Create plot of ellipses from spatial points data frame on Google Maps, by using three attributes: semi-major axis, semi-minor axis, and orientation in degrees.</i>
-------------------	--

Description

Plot htm output with Google Maps API in form of ellipses plot of spatial data. Ready to use as local htm file or into your own website.

Usage

```

ellipseGoogleMaps(SP,
  filename = '',
  zcol = 1:3,
  add = F,
  previousMap = NULL,
  scale_e = 10,
  colPalette = NULL,
  strokeColor = '#FFAA00',
  strokeOpacity = 1,
  fillOpacity=0.7,
  strokeWeight = 1,
  geodesic = TRUE,
  clickable = TRUE,
  zIndex = 'null',
  map.width="100%",
  map.height="100%",
  layerName="",
  control.width="100%",
  control.height="100%",
  zoom = 15,
  fitBounds = TRUE,
  mapTypeId = 'HYBRID',
  disableDoubleClickZoom = FALSE,

```

```

draggable = TRUE,
keyboardShortcuts = TRUE,
mapTypeControlOptions = 'DEFAULT',
navigationControl = TRUE,
navigationControlOptions = 'DEFAULT',
scaleControlOptions = 'STANDARD',
noClear = FALSE,
scrollwheel = TRUE,
streetViewControl = FALSE,
legend=TRUE,
control=TRUE,
InfoWindowControl=list(map="map", event="click", position="event.latLng",
                        disableAutoPan=FALSE, maxWidth=330,
                        pixelOffset="null", zIndex="null") ,
api="https://maps.google.com/maps/api/js?sensor=false",
openMap=TRUE)

```

Arguments

SP	object of SpatialPointsDataFrame-class with associated coordinate reference systems and minimum three attributes: semi-major axis, semi-minor axis, and orientation in degrees.
filename	the name of the output htm or html file. If filename is not defined the function creates temporary file.
zcol	variable column names, or column numbers after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column. Attribute names or numbers of columns contains semi-major axis, semi-minor axis, and orientation in degrees respectively
add	logical; if TRUE the result of the function will be a list stored as variable in the R. It is possible to combine more layers in the one plot, previously saved output from plotGoogleMaps should be given in the previousMap attribute.
previousMap	previously saved variable from plotGoogleMaps when attribute add is TRUE. It is solution for combining more than one layers in the one plot.
scale_e	if is 10 then ellipses is drawn 10:1
colPalette	colours to be used to fill plotting symbols; numeric vector of same size like key.entries
strokeColor	the color to draw the border of circle (the plotting symbols), NULL for no border
strokeOpacity	the stroke opacity between 0.0 and 1.0
fillOpacity	the fill opacity between 0.0 and 1.0
strokeWeight	the stroke width in pixels
geodesic	render each edge as a geodesic (a segment of a 'great circle')
clickable	indicates whether this plotting symbol handles click events
zIndex	the zIndex compared to other polygons
map.width	the width of the map. Can be in percent or in pixels (px)

<code>map.height</code>	the height of the map. Can be in percent or in pixels (px)
<code>layerName</code>	the name of the layer. Layer name appears on the check box for switching layer of and on in output htm.
<code>control.width</code>	the width of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>control.height</code>	the height of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>zoom</code>	the initial Map zoom level. Zoom levels between 0 (the lowest zoom level, in which the entire world can be seen on one map) to 19 (the highest zoom level, down to individual buildings) are possible within the normal maps view. Argument <code>fitBounds</code> should be FALSE.
<code>fitBounds</code>	sets the maps to fit to the boundary box values of <code>sp</code> object
<code>mapTypeId</code>	the initial Map <code>mapTypeId</code> . 'HYBRID' - map type displays a transparent layer of major streets on satellite images. 'ROADMAP' - map type displays a normal street map. 'SATELLITE' - map type displays satellite images. 'TERRAIN' - map type displays maps with physical features such as terrain and vegetation.
<code>disableDoubleClickZoom</code>	enables or disables zoom and center on double click.
<code>draggable</code>	if FALSE, prevents the map from being dragged
<code>keyboardShortcuts</code>	if FALSE, prevents the map from being controlled by the keyboard.
<code>mapTypeControlOptions</code>	the initial display options for the Map type control. 'DEFAULT' - uses the default map type control. 'DROPDOWN_MENU' - a dropdown menu for the screen realestate conscious. 'HORIZONTAL_BAR' - the standard horizontal radio buttons bar.
<code>navigationControl</code>	enabled or disabled state of the navigation control
<code>navigationControlOptions</code>	the initial display options for the navigation control. 'ANDROID' - the small zoom control similar to the one used by the native Maps application on Android. 'DEFAULT' - the default navigation control. The control which DEFAULT maps to will vary according to map size and other factors. It may change in future versions of the API. 'SMALL' - the small, zoom only control. 'ZOOM_PAN' - the larger control, with the zoom slider and pan directional pad.
<code>scaleControlOptions</code>	the initial display options for the scale control.
<code>noClear</code>	if TRUE, do not clear the contents of the Map div.
<code>scrollwheel</code>	if FALSE, disables scrollwheel zooming on the map.
<code>streetViewControl</code>	the initial enabled or disabled state of the Street View pegman control
<code>legend</code>	if FALSE, disables legend.
<code>control</code>	if FALSE, disables control.

InfoWindowControl
 see <https://developers.google.com/maps/documentation/javascript/reference#InfoWindowOptions>

api
 see <https://developers.google.com/maps/documentation/javascript/tutorial>

openMap
 if TRUE map is browsed by default browser

Value

This function returns a list with HTML, JavaScript, Google Maps API key and else what is necessary for the final web map. The second output is htm file ready to use stored in working directory.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[plotGoogleMaps](#), [segmentGoogleMaps](#), [bubbleGoogleMaps](#)

Examples

```
# Data preparation
ell<- data.frame(E=c(7456263,7456489,7456305),
  N=c(4954146,4952978,4952695),A=c( 2.96,4.55,7.10),
  B=c(2.35,2.11,2.29),teta=c(28.35242,41.04491,38.47216))

coordinates(ell) <- ~E+N
proj4string(ell) <- CRS("+proj=tmerc +lat_0=0 +lon_0=21 +k=0.9999
  +x_0=7500000 +y_0=0 +ellps=bessel
  +towgs84=574.027,170.175,401.545,4.88786,-0.66524,-13.24673,0.99999311067
  +units=m")

m<-ellipseGoogleMaps(ell,mapTypeId='ROADMAP')
```

e1SPDF

Create ellipses in form of SpatialPolygonsDataFrame.

Description

Create ellipses in form of SpatialPolygonsDataFrame based on attributes: semi-major axis, semi-minor axis, and orientation in degrees.

Usage

```
e1SPDF(SPDF,
  zcol=1:3,
  scale_e=10)
```

Arguments

SPDF	object of SpatialPointsDataFrame-class with associated coordinate reference systems and minimum three attributes: semi-major axis, semi-minor axis, and orientation in degrees.
zcol	variable column names, or column numbers after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column
scale_e	scale factor

Value

The function provide SpatialPolygonsDataFrame of ellipses.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[pieSP](#), [elSPDF](#), [bubbleGoogleMaps](#)

Examples

```
# Data preparation
ell<- data.frame(E=c(7456263,7456489,7456305),N=c(4954146,4952978,4952695),
  A=c( 2.96,4.55,7.10),B=c(2.35,2.11,2.29),
  teta=c(28.35242,41.04491,38.47216))
coordinates(ell) <- ~E+N
proj4string(ell) <- CRS("+proj=tmerc +lat_0=0 +lon_0=21 +k=0.9999
+x_0=7500000 +y_0=0 +ellps=bessel
+towgs84=574.027,170.175,401.545,4.88786,-0.66524,-13.24673,0.99999311067
+units=m")

ellpses<-elSPDF(ell)
splot(ellpses,"A")
```

iconlabels

Create icon markers depending on attribute data.

Description

The function provide links to icon markers or icon labels ready to use for plotGoogleMaps.

Usage

```
iconlabels(attribute,
           colPalette=NULL,
           at=NULL,
           height=10,
           icon=FALSE,
           scale=0.6
          )
```

Arguments

attribute	vector of attribute data
colPalette	colours to be used to fill marker symbols or labels
at	values at which colours will change
height	text height in pixels
icon	if false create just labels, contrariwise create markers and labels in markers
scale	scale of marker icon

Value

The function provide links to icon markers or icon labels ready to use for plotGoogleMaps.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[plotGoogleMaps](#), [ellipseGoogleMaps](#), [bubbleGoogleMaps](#)

Examples

```
# Data preparation
data(meuse)
coordinates(meuse)<--x+y
proj4string(meuse) <- CRS('+init=epsg:28992')

m<-plotGoogleMaps(meuse,zcol='zinc')

# zinc labels
ic=iconlabels(meuse$zinc, height=12)
m<-plotGoogleMaps(meuse,zcol='zinc', iconMarker=ic)

# landuse labels and markers
ic=iconlabels(meuse$landuse, height=12, colPalette=rainbow(15) )
m<-plotGoogleMaps(meuse,zcol='landuse',colPalette=rainbow(15), iconMarker=ic)

ic=iconlabels(meuse$landuse, height=12, colPalette='#9ECAE1', icon=TRUE)
```

```
m<-plotGoogleMaps(meuse,zcol='landuse',colPalette='#9ECAE1', iconMarker=ic)
```

mcGoogleMaps	<i>Create a plot of spatial data on Google Maps, creates and manages per-zoom-level clusters for large amounts of markers</i>
--------------	---

Description

Plot htm output with Google Maps API and a plot of spatial points data as a combination of users' data and Google Maps layers. Ready to use as local htm file or into your own website. This function uses MarkerClusterer for Google Maps v3 - library creates and manages per-zoom-level clusters for large amounts of markers.

Usage

```
mcGoogleMaps(SP,
  filename = '',
  zcol = 1,
  at=NULL,
  add = FALSE,
  previousMap = NULL,
  colPalette = NULL,
  geodesic = TRUE,
  clickable = TRUE,
  draggableMarker = FALSE,
  iconMarker='',
  flat = TRUE,
  visible = TRUE,
  zIndex = 'null',
  map.width="100%",
  map.height="100%",
  layerName="",
  control.width="100%",
  control.height="100%",
  zoom = 15,
  fitBounds = TRUE,
  mapTypeId = 'HYBRID',
  disableDoubleClickZoom = FALSE,
  draggable = TRUE,
  keyboardShortcuts = TRUE,
  mapTypeControlOptions = 'DEFAULT',
  navigationControl = TRUE,
  navigationControlOptions = 'DEFAULT',
  scaleControlOptions = 'STANDARD',
  noClear = FALSE,
  scrollwheel = TRUE,
```

```

streetViewControl = FALSE,
legend=TRUE,
control=TRUE,
InfoWindowControl=list(map=map, event="click",position="event.latLng",
                        disableAutoPan=FALSE, maxWidth=330,
                        pixelOffset="null",
                        zIndex="null") ,

map="map",
mapCanvas="map_canvas",
css = "",
api="https://maps.google.com/maps/api/js?sensor=false",
apiMarkerClusterer=paste('http://google-maps-utility-library-v3.',
                          'googlecode.com/svn/trunk/',
                          'markerclusterer/src/markerclusterer.js',sep=""),
openMap=TRUE)

```

Arguments

SP	object of Spatial-class with associated coordinate reference systems
filename	the name of the output htm or html file. If filename is not defined the function creates temporary file.
zcol	variable column name, or column number after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column
at	values at which colours will change
add	logical; if TRUE the result of the function will be a list stored as variable in the R. It is possible to combine more layers in the one plot, previously saved output from plotGoogleMaps should be given in the previousMap attribute.
previousMap	previously saved variable from plotGoogleMaps or bubbleGoogleMaps when attribute add is TRUE. It is solution for combining more than one layers in the one plot.
colPalette	colours to be used to fill polygon or colour for the line
geodesic	render each edge as a geodesic (a segment of a 'great circle')
clickable	indicates whether this marker (point), polygon or line handles click events
draggableMarker	if TRUE, the marker can be dragged
iconMarker	if "" default marker icon, link to the image or local image to be used as marker image
flat	if TRUE, the marker shadow will not be displayed
visible	if TRUE, the marker is visible
zIndex	the zIndex compared to other polygons
map.width	the width of the map. Can be in percent or in pixels (px)
map.height	the height of the map. Can be in percent or in pixels (px)
layerName	the name of the layer. Layer name appears on the check box for switching layer of and on in output htm.

<code>control.width</code>	the width of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>control.height</code>	the height of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>zoom</code>	the initial Map zoom level. Zoom levels between 0 (the lowest zoom level, in which the entire world can be seen on one map) to 19 (the highest zoom level, down to individual buildings) are possible within the normal maps view. Argument <code>fitBounds</code> should be FALSE.
<code>fitBounds</code>	sets the maps to fit to the boundary box values of <code>sp</code> object
<code>mapTypeId</code>	the initial Map <code>mapTypeId</code> . 'HYBRID' - map type displays a transparent layer of major streets on satellite images. 'ROADMAP' - map type displays a normal street map. 'SATELLITE' - map type displays satellite images. 'TERRAIN' - map type displays maps with physical features such as terrain and vegetation.
<code>disableDoubleClickZoom</code>	enables or disables zoom and center on double click.
<code>draggable</code>	if FALSE, prevents the map from being dragged
<code>keyboardShortcuts</code>	if FALSE, prevents the map from being controlled by the keyboard.
<code>mapTypeControlOptions</code>	the initial display options for the Map type control. 'DEFAULT' - uses the default map type control. 'DROPDOWN_MENU' - a dropdown menu for the screen realestate conscious. 'HORIZONTAL_BAR' - the standard horizontal radio buttons bar.
<code>navigationControl</code>	enabled or disabled state of the navigation control
<code>navigationControlOptions</code>	the initial display options for the navigation control. 'ANDROID' - the small zoom control similar to the one used by the native Maps application on Android. 'DEFAULT' - the default navigation control. The control which DEFAULT maps to will vary according to map size and other factors. It may change in future versions of the API. 'SMALL' - the small, zoom only control. 'ZOOM_PAN' - the larger control, with the zoom slider and pan directional pad.
<code>scaleControlOptions</code>	the initial display options for the scale control.
<code>noClear</code>	if TRUE, do not clear the contents of the Map div.
<code>scrollwheel</code>	if FALSE, disables scrollwheel zooming on the map.
<code>streetViewControl</code>	The initial enabled or disabled state of the Street View pegman control
<code>legend</code>	if FALSE, disables legend.
<code>control</code>	if FALSE, disables control.
<code>InfoWindowControl</code>	see https://developers.google.com/maps/documentation/javascript/reference#InfoWindowOptions
<code>map</code>	name of map object in JavaScript

mapCanvas	name of map DIV element
css	CSS for mapCanvas and cBoxes
api	see https://developers.google.com/maps/documentation/javascript/tutorial
apiMarkerClusterer	see http://google-maps-utility-library-v3.googlecode.com/svn/trunk/markerclusterer/docs/reference.html
openMap	if TRUE map is browsed by default browser

Value

This function returns a list with HTML, JavaScript, Google Maps API key and else what is necessary for the final web map. The second output is html file ready to use.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[plotGoogleMaps](#), [bubbleGoogleMaps](#), [ellipseGoogleMaps](#), [segmentGoogleMaps](#)

Examples

```
# Data preparation
# Point data
data(meuse)
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS('+init=epsg:28992')

m <- mcGoogleMaps(meuse, zcol='zinc')
```

pieSP

Create pie charts data in form of SpatialPolygonsDataFrame.

Description

Create pie charts data in form of SpatialPolygonsDataFrame depending on few attributes.

Usage

```
pieSP(SPDF,
      zcol=1:length(SPDF@data),
      scalelist=TRUE,
      max.radius=100,
      do.sqrt = TRUE
)
```


Arguments

SPDF	object of SpatialPointsDataFrame-class with associated coordinate reference systems
zcol	variable column names, or column numbers after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column
scalelist	if TRUE circles are different in size, FALSE all circles have same radius
max.radius	value for maximum radius in meter
do.sqrt	logical; if TRUE the plotting symbol area (sqrt(diameter)) is proportional to the value of the z-variable; if FALSE, the symbol size (diameter) is proportional to the z-variable

Value

The function provide SpatialPolygonsDataFrame depending on few attributes, ready to use for plot-GoogleMaps or spplot.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[pieSP](#), [elSPDF](#), [bubbleSP](#)

Examples

```
# Data preparation
data(meuse)
coordinates(meuse)<--x+y
proj4string(meuse) <- CRS('+init=epsg:28992')

pies<-pieSP(meuse,zcol=c('zinc','lead','copper'), max.radius=120)
pies$pie=rep(c('zinc','lead','copper'),155)

m=plotGoogleMaps(pies, zcol='pie')

pies$pie2=rep(1:3,155)
spplot(pies, 'pie2')

pies2<-pieSP(meuse,zcol=c('zinc','dist.m'), max.radius=70,scalelist=FALSE)
pies2$pie=rep(c('zinc','dist.m'),155)

m=plotGoogleMaps(pies2, zcol='pie')

# plotKML(pies, col='pie')
```

plotGoogleMaps	<i>Create a plot of spatial data on Google Maps</i>
----------------	---

Description

Plot htm output with Google Maps API and a plot of spatial data as a combination of users' data and Google Maps layers. Ready to use as local htm file or into your own website.

Usage

```
plotGoogleMaps(SP,
  filename = '',
  zcol = 1,
  at=NULL,
  add = FALSE,
  previousMap = NULL,
  colPalette = NULL,
  strokeColor = "",
  strokeOpacity = 1,
  fillOpacity=0.7,
  strokeWeight = 1,
  geodesic = TRUE,
  clickable = TRUE,
  draggableMarker = FALSE,
  iconMarker='',
  flat = TRUE,
  visible = TRUE,
  zIndex = 'null',
  map.width="100%",
  map.height="100%",
  layerName="",
  control.width="100%",
  control.height="100%",
  zoom = 15,
  fitBounds = TRUE,
  mapTypeId = 'HYBRID',
  disableDoubleClickZoom = FALSE,
  draggable = TRUE,
  keyboardShortcuts = TRUE,
  mapTypeControlOptions = 'DEFAULT',
  navigationControl = TRUE,
  navigationControlOptions = 'DEFAULT',
  scaleControlOptions = 'STANDARD',
  noClear = FALSE,
  scrollwheel = TRUE,
  streetViewControl = FALSE,
  legend=TRUE,
```

```

control=TRUE,
InfoWindowControl=list(map=map, event="click",position="event.latLng",
                        disableAutoPan=FALSE, maxWidth=330,
                        pixelOffset="null", zIndex="null") ,
map="map",
mapCanvas="map_canvas",
css = "",
api="https://maps.google.com/maps/api/js?sensor=false",
openMap=TRUE)

```

Arguments

SP	object of Spatial-class with associated coordinate reference systems
filename	the name of the output htm or html file. If filename is not defined the function creates temporary file.
zcol	variable column name, or column number after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column
at	values at which colours will change
add	logical; if TRUE the result of the function will be a list stored as variable in the R. It is possible to combine more layers in the one plot, previously saved output from plotGoogleMaps should be given in the previousMap attribute.
previousMap	previously saved variable from plotGoogleMaps or bubbleGoogleMaps when attribute add is TRUE. It is solution for combining more than one layers in the one plot.
colPalette	colours to be used to fill polygon or colour for the line
strokeColor	the color to draw the borders of polygon, NULL for no border
strokeOpacity	the stroke opacity between 0.0 and 1.0 for the polygon or line
fillOpacity	the fill opacity (polygons) between 0.0 and 1.0
strokeWeight	the stroke width in pixels for polygon border or line
geodesic	render each edge as a geodesic (a segment of a 'great circle')
clickable	indicates whether this marker (point), polygon or line handles click events
draggableMarker	if TRUE, the marker can be dragged
iconMarker	if "" default marker icon, link to the image or local image to be used as marker image
flat	if TRUE, the marker shadow will not be displayed
visible	if TRUE, the marker is visible
zIndex	the zIndex compared to other polygons
map.width	the width of the map. Can be in percent or in pixels (px)
map.height	the height of the map. Can be in percent or in pixels (px)
layerName	the name of the layer. Layer name appears on the check box for switching layer of and on in output htm.

<code>control.width</code>	the width of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>control.height</code>	the height of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>zoom</code>	the initial Map zoom level. Zoom levels between 0 (the lowest zoom level, in which the entire world can be seen on one map) to 19 (the highest zoom level, down to individual buildings) are possible within the normal maps view. Argument <code>fitBounds</code> should be FALSE.
<code>fitBounds</code>	sets the maps to fit to the boundary box values of <code>sp</code> object
<code>mapTypeId</code>	the initial Map <code>mapTypeId</code> . 'HYBRID' - map type displays a transparent layer of major streets on satellite images. 'ROADMAP' - map type displays a normal street map. 'SATELLITE' - map type displays satellite images. 'TERRAIN' - map type displays maps with physical features such as terrain and vegetation.
<code>disableDoubleClickZoom</code>	enables or disables zoom and center on double click.
<code>draggable</code>	if FALSE, prevents the map from being dragged
<code>keyboardShortcuts</code>	if FALSE, prevents the map from being controlled by the keyboard.
<code>mapTypeControlOptions</code>	the initial display options for the Map type control. 'DEFAULT' - uses the default map type control. 'DROPDOWN_MENU' - a dropdown menu for the screen realestate conscious. 'HORIZONTAL_BAR' - the standard horizontal radio buttons bar.
<code>navigationControl</code>	enabled or disabled state of the navigation control
<code>navigationControlOptions</code>	the initial display options for the navigation control. 'ANDROID' - the small zoom control similar to the one used by the native Maps application on Android. 'DEFAULT' - the default navigation control. The control which DEFAULT maps to will vary according to map size and other factors. It may change in future versions of the API. 'SMALL' - the small, zoom only control. 'ZOOM_PAN' - the larger control, with the zoom slider and pan directional pad.
<code>scaleControlOptions</code>	the initial display options for the scale control.
<code>noClear</code>	if TRUE, do not clear the contents of the Map div.
<code>scrollwheel</code>	if FALSE, disables scrollwheel zooming on the map.
<code>streetViewControl</code>	The initial enabled or disabled state of the Street View pegman control
<code>legend</code>	if FALSE, disables legend.
<code>control</code>	if FALSE, disables control.
<code>InfoWindowControl</code>	see https://developers.google.com/maps/documentation/javascript/reference#InfoWindowOptions
<code>map</code>	name of map object in JavaScript

mapCanvas	name of map DIV element
css	CSS for mapCanvas and cBoxes
api	see https://developers.google.com/maps/documentation/javascript/tutorial
openMap	if TRUE map is browsed by default browser

Value

This function returns a list with HTML, JavaScript, Google Maps API key and else what is necessary for the final web map. The second output is html file ready to use.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[bubbleGoogleMaps](#), [ellipseGoogleMaps](#), [segmentGoogleMaps](#)

Examples

```
# Data preparation
# Point data
data(meuse)
require(maptools)
coordinates(meuse)<--~x+y
proj4string(meuse) <- CRS('+init=epsg:28992')
# Line data
data(meuse.grid)
coordinates(meuse.grid)<-c('x','y')
meuse.grid<-as(meuse.grid,'SpatialPixelsDataFrame')
im<-as.image.SpatialGridDataFrame(meuse.grid['dist'])
cl<-ContourLines2SLDF(contourLines(im))
proj4string(cl) <- CRS('+init=epsg:28992')

# Create web map of Point data
m<-plotGoogleMaps(meuse)

# Combine point and line data
mapMeusePoints<- plotGoogleMaps(meuse,add=TRUE,mapTypeId='TERRAIN')
mapMeuseCl<- plotGoogleMaps(cl,previousMap=mapMeusePoints,filename='myMap2.htm')
# see results in your working directory
```

PolyCol

Create list of colors depending on attribute data.

Description

The function provide list of colors, unique colors, levels of attribute,attribute breaks.

Usage

```
PolyCol(attribute,
         colPalette=NULL,
         at=NULL)
```

Arguments

attribute	vector of attribute data
colPalette	colours to be used to fill features depending on attribute
at	values at which colours will change

Value

The function provide list of colors (cols), unique colors (col.uniq), levels of attribute (att), attribute breaks (brks).

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[plotGoogleMaps](#), [ellipseGoogleMaps](#), [bubbleGoogleMaps](#)

Examples

```
# Data preparation
data(meuse)
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS('+init=epsg:28992')

m <- plotGoogleMaps(meuse, zcol='zinc')
# see results in your working directory

# similar classic plot
cols = PolyCol(meuse$zinc)$cols
plot(meuse, col=cols)

str(PolyCol(meuse$zinc))

m <- plotGoogleMaps(meuse, zcol='zinc', at=c(0,300,600, max(meuse$zinc)), colPalette=rainbow(3))
# similar classic plot
cols = PolyCol(meuse$zinc, colPalette=rainbow(3), at=c(0,300,600, max(meuse$zinc)) )$cols
plot(meuse, col=cols)

# plotKML(meuse, colour='zinc', colour_scale =cols)
```

segmentGoogleMaps *Create a segment circles - pie charts plot of spatial data on Google Maps, by using two or more related attributes.*

Description

Plot htm output with Google Maps API in form of pie charts plot form spatial data by using two or more related attributes. Ready to use as local htm file or into your own website.

Usage

```
segmentGoogleMaps(SP,
                  zcol = 1:length(SP@data),
                  filename = '',
                  max.radius = 100,
                  scalelist = TRUE,
                  do.sqrt = FALSE,
                  add = F,
                  previousMap = NULL,
                  colPalette = rainbow(ncol(SP@data[, zcol])),
                  strokeColor = '',
                  strokeOpacity = 1,
                  strokeWeight = 1,
                  fillOpacity = 0.7,
                  geodesic = TRUE,
                  clickable = TRUE,
                  zIndex = 'null',
                  map.width="100%",
                  map.height="100%",
                  layerName="",
                  control.width="100%",
                  control.height="100%",
                  zoom = 15,
                  fitBounds = TRUE,
                  mapTypeId = 'HYBRID',
                  disableDoubleClickZoom = FALSE,
                  draggable = TRUE,
                  keyboardShortcuts = TRUE,
                  mapTypeControlOptions = 'DEFAULT',
                  navigationControl = TRUE,
                  navigationControlOptions = 'DEFAULT',
                  scaleControlOptions = 'STANDARD',
                  noClear = FALSE,
                  scrollwheel = TRUE,
                  streetViewControl = FALSE,
                  legend=TRUE,
                  control=TRUE,
```

```

InfoWindowControl=list(map=map, event="click",position="event.latLng",
  disableAutoPan=FALSE, maxWidth=330,pixelOffset="null",
  zIndex="null") ,
  map="map",
  mapCanvas="map_canvas",
  css = "",
  api="https://maps.google.com/maps/api/js?sensor=false",
  openMap=TRUE)

```

Arguments

SP	object of SpatialPointsDataFrame-class with associated coordinate reference systems
filename	the name of the output htm or html file. If filename is not defined the function creates temporary file.
zcol	variable column names, or column numbers after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column
max.radius	value for largest circle (the plotting symbols) in metre
scalelist	if TRUE circles are different in size, FALSE all circles have same radius
do.sqrt	logical; if TRUE the plotting symbol area (sqrt(diameter)) is proportional to the value of the z-variable; if FALSE, the symbol size (diameter) is proportional to the z-variable
add	logical; if TRUE the result of the function will be a list stored as variable in the R. It is possible to combine more layers in the one plot, previously saved output from plotGoogleMaps should be given in the previousMap attribute.
previousMap	previously saved variable from plotGoogleMaps when attribute add is TRUE. It is solution for combining more than one layers in the one plot.
colPalette	colours to be used to fill plotting symbols; numeric vector of same size like key.entries
strokeColor	the color to draw the border of circle (the plotting symbols), NULL for no border
strokeOpacity	the stroke opacity between 0.0 and 1.0
strokeWeight	the stroke width in pixels
fillOpacity	the fill opacity between 0.0 and 1.0
geodesic	render each edge as a geodesic (a segment of a 'great circle')
clickable	indicates whether this plotting symbol handles click events
zIndex	the zIndex compared to other polygons
map.width	the width of the map. Can be in percent or in pixels (px)
map.height	the height of the map. Can be in percent or in pixels (px)
layerName	the name of the layer. Layer name appears on the check box for switching layer of and on in output htm.
control.width	the width of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).

<code>control.height</code>	the height of the part of htm for controlling layer (legend, opacity,...) Can be in percent or in pixels (px).
<code>zoom</code>	the initial Map zoom level. Zoom levels between 0 (the lowest zoom level, in which the entire world can be seen on one map) to 19 (the highest zoom level, down to individual buildings) are possible within the normal maps view. Argument <code>fitBounds</code> should be <code>FALSE</code> .
<code>fitBounds</code>	sets the maps to fit to the boundary box values of <code>sp</code> object
<code>mapTypeId</code>	the initial Map <code>mapTypeId</code> . <code>'HYBRID'</code> - map type displays a transparent layer of major streets on satellite images. <code>'ROADMAP'</code> - map type displays a normal street map. <code>'SATELLITE'</code> - map type displays satellite images. <code>'TERRAIN'</code> - map type displays maps with physical features such as terrain and vegetation.
<code>disableDoubleClickZoom</code>	enables or disables zoom and center on double click.
<code>draggable</code>	if <code>FALSE</code> , prevents the map from being dragged
<code>keyboardShortcuts</code>	if <code>FALSE</code> , prevents the map from being controlled by the keyboard.
<code>mapTypeControlOptions</code>	the initial display options for the Map type control. <code>'DEFAULT'</code> - uses the default map type control. <code>'DROPDOWN_MENU'</code> - a dropdown menu for the screen realestate conscious. <code>'HORIZONTAL_BAR'</code> - the standard horizontal radio buttons bar.
<code>navigationControl</code>	enabled or disabled state of the navigation control
<code>navigationControlOptions</code>	the initial display options for the navigation control. <code>'ANDROID'</code> - the small zoom control similar to the one used by the native Maps application on Android. <code>'DEFAULT'</code> - the default navigation control. The control which <code>DEFAULT</code> maps to will vary according to map size and other factors. It may change in future versions of the API. <code>'SMALL'</code> - the small, zoom only control. <code>'ZOOM_PAN'</code> - the larger control, with the zoom slider and pan directional pad.
<code>scaleControlOptions</code>	the initial display options for the scale control.
<code>noClear</code>	if <code>TRUE</code> , do not clear the contents of the Map div.
<code>scrollwheel</code>	if <code>FALSE</code> , disables scrollwheel zooming on the map.
<code>streetViewControl</code>	the initial enabled or disabled state of the Street View pegman control
<code>legend</code>	if <code>FALSE</code> , disables legend.
<code>control</code>	if <code>FALSE</code> , disables control.
<code>InfoWindowControl</code>	see https://developers.google.com/maps/documentation/javascript/reference#InfoWindowOptions
<code>map</code>	name of map object in JavaScript
<code>mapCanvas</code>	name of map DIV element
<code>css</code>	CSS for <code>mapCanvas</code> and <code>cBoxes</code>
<code>api</code>	see https://developers.google.com/maps/documentation/javascript/tutorial
<code>openMap</code>	if <code>TRUE</code> map is browsed by default browser

Arguments

stfdf	object of spacetime (STFDF-class) with associated coordinate reference systems
zcol	variable column name, or column number after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column, it has to be numeric
filename	the name of the output htm or html file. If filename is not defined the function creates temporary file.
layerName	the name of the layer. Layer name appears on the check box for switching layer of and on in output htm.
plotNames	vector of of plot titles in infoWindow plots
aggregateFUN	a function to compute the summary statistics of time series
round.att	integer indicating the number of decimal places for plotting attribute, if not a number no round
plot.height	the height of of plot in infoWindow plot in pixels (px)
plot.width	the width of plot in infoWindow plot in pixels (px)
...	arguments from plotGoogleMaps

Value

This function returns a html(htm) file ready to use.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[stplotGoogleMaps](#), [plotGoogleMaps](#)

Examples

```
## Data preparation
## STFDF data from spacetime vignette spacetime: Spatio-Temporal Data in R
#library("maps")
# states.m = map('state', plot=FALSE, fill=TRUE)
# IDs <- sapply(strsplit(states.m$names, ":"), function(x) x[1])
# library("maptools")
#states = map2SpatialPolygons(states.m, IDs=IDs)
#yrs = 1970:1986
#time = as.POSIXct(paste(yrs, "-01-01", sep=""), tz = "GMT")
# library("plm")
#data("Produc")
#Produc.st = STFDF(states[-8], time, Produc[order(Produc[2], Produc[1]),])
#Produc.st@sp@proj4string=CRS('+proj=longlat +datum=WGS84')
# m <- stfdfGoogleMaps(Produc.st, zcol= 'unemp')
```

stplotGoogleMaps *Create a plot of spacetime (STDIF,STFDF) data on Google Maps*

Description

Plot htm output with Google Maps API and a plot of spacetime (STDIF,STFDF) data as a combination of users' data and Google Maps layers. Ready to use as local htm file or into your own website.

Usage

```
stplotGoogleMaps(SPT,
                 zcol=1,
                 stfilename='spacetime.htm',
                 filename='file',
                 w="100%",
                 h="49.5%",
                 openMap=FALSE,
                 colPalette=NULL,
                 do.bubble=FALSE,
                 at=NULL,
                 bubble= list(max.radius=10000,
                              key.entries = if(do.bubble)
                              {quantile(SPT@data[,zcol],(1:5)/5,
                              na.rm=T)}else{0},
                              do.sqrt = TRUE),
                 iconMarker="",
                 ...)
```

Arguments

SPT	object of spacetime (STFDF-class, STIDF-class) with associated coordinate reference systems
zcol	variable column name, or column number after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column
filename	string use as prefix to partial files, each time instance
stfilename	the name of the output htm or html file
w	the width of the map for each time instance. Can be in percent or in pixels (px)
h	the height of the map for each time instance. Can be in percent or in pixels (px)
openMap	if TRUE map is browsed by default browser for each time instance
colPalette	colours to be used to fill polygon or colour for the line
do.bubble	if TRUE creates bubble plots
at	values at which colours will change

bubble	list of parameters to be passed to bubble plot. The max.radius value for maximum radius in meter. The key.entries the values that will be plotted in the key; by default the five quantiles min, q.25, median q.75, max. The do.sqrt logical; if TRUE the plotting symbol area (sqrt(diameter)) is proportional to the value of the z-variable; if FALSE, the symbol size (diameter) is proportional to the z-variable
iconMarker	if it is empty string creates default marker icon, link to the image or local image to be used as marker image
...	arguments from plotGoogleMaps

Value

This function returns a html(htm) file ready to use.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[plotGoogleMaps](#), [stfdfGoogleMaps](#)

Examples

```
## Data preparation
## Point data
## data from plotKML package and plotKML tutorial
#library(plotKML)
#data(HRtemp08)
# HRtemp08$time <- as.POSIXct(HRtemp08$DATE, format="%Y-%m-%dT%H:%M:%SZ")
# library(spacetime)
# sp <- SpatialPoints(HRtemp08[,c("Lon", "Lat")])
# proj4string(sp) <- CRS("+proj=longlat +datum=WGS84")
# HRtemp08.st <- STIDF(sp, time = HRtemp08$time, data = HRtemp08[,c("NAME", "TEMP")])
# HRtemp08_jan <- HRtemp08.st[1:500]
#str(HRtemp08_jan)
# plot STIDF
# stplotGoogleMaps(HRtemp08_jan, zcol='TEMP', mapTypeId='ROADMAP', w='49%', h='49%')
# plot STIDF bubble
# stplotGoogleMaps(HRtemp08_jan, zcol='TEMP', stfilename='HR_temp.html',
#                   mapTypeId='ROADMAP', w='49%', h='49%', strokeOpacity = 0,
#                   do.bubble=T, bubble= list(max.radius=15000,
#                   key.entries =quantile(HRtemp08_jan@data[, 'TEMP'], (1:5)/5, na.rm=T),
#                   do.sqrt = F) )
#
#####
## STDF data from spacetime vignette spacetime: Spatio-Temporal Data in R
#library("maps")
# states.m = map('state', plot=FALSE, fill=TRUE)
# IDs <- sapply(strsplit(states.m$names, ":"), function(x) x[1])
# library("maptools")
```

```

#states = map2SpatialPolygons(states.m, IDs=IDs)
#yrs = 1970:1986
#time = as.POSIXct(paste(yrs, "-01-01", sep=""), tz = "GMT")
# library("plm")
#data("Produc")
#Produc.st = STDF(states[-8], time, Produc[order(Produc[2], Produc[1]),])
#Produc.st@sp@proj4string=CRS('+proj=longlat +datum=WGS84')
#library(RColorBrewer)
#ee= stplotGoogleMaps(Produc.st,zcol='unemp',stfilename='USA.htm',
# colPalette=brewer.pal(9, "YlOrRd"), mapTypeId='ROADMAP',
#w='49%',h='49%', fillOpacity=0.85)
## without control
# ee= stplotGoogleMaps(Produc.st,zcol='unemp',
# stfilename='USA2.htm',colPalette=brewer.pal(9, "YlOrRd"),
# mapTypeId='ROADMAP',w='33%',h='25%', fillOpacity=0.85, control.width=0)

```

vectorsSP

Create radius vector data in form of SpatialLinesDataFrame.

Description

Create radius vector data in form of SpatialLinesDataFrame based on radius and azimuth.

Usage

```

vectorsSP(SPDF,
          zcol=1:2,
          maxLength=30000,
          arrowSize=0.15,
          arrAng=30
          )

```

Arguments

SPDF	object of SpatialPointsDataFrame-class with associated coordinate reference systems and minimum two attributes: radius in meter and azimuth in degrees.
zcol	variable column names, or column numbers after removing spatial coordinates from x@data: 1 refers to the first non-coordinate column. Attribute names or numbers of columns contains radius in meter and azimuth in degrees
maxLength	value for maximum vector in meter
arrowSize	size of arrow relative to vector size
arrAng	angle of arrow in degrees, 0 no arrow

Value

The function provide SpatialLinesDataFrame based on radius and azimuth, ready to use for plot-GoogleMaps or splot.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

[pieSP](#), [elSPDF](#), [bubbleSP](#)

Examples

```
# Data preparation
wind<-data.frame( Station=c('Zlatibor' , 'Sjenica' ,
  'Pozega' , 'Kraljevo', 'Kopaonik', 'Kursumlija', 'Krusevac'),
  Lon=c(19.708418 , 19.999311 , 20.029322,
  20.724012, 20.776111, 21.279513, 21.307998 ),
  Lat=c(43.729755 , 43.267913 ,43.84859 ,
  43.726812 ,43.2725,43.137053 ,43.574188),
  Temp=c(7, 7,5,6,1,6,9),
  Wind_spd=c(3 , 6,1 ,1,7 ,1 ,2),
  Wind_dir=c(45, 162, 138 ,265,335 ,180 ,270) )

coordinates(wind)<- ~Lon+Lat
wind@proj4string=CRS('+proj=longlat +datum=WGS84')

wind_vect=vectorsSP(wind,zcol=c('Wind_spd', 'Wind_dir'))

pal<-colorRampPalette(c( "#CB181D", "#A50F15", "#67000D"), space = "Lab")
spplot(wind_vect, 'Wind_spd', col.regions=pal(5) )
m=plotGoogleMaps(wind_vect,zcol='Wind_spd',
  colPalette=pal(5), mapTypeId='ROADMAP',strokeWeight=3)

# plotKML(wind_vect)
```

Index

`bubbleGoogleMaps`, [2](#), [6](#), [10–12](#), [16](#), [21](#), [22](#), [26](#)
`bubbleSP`, [5](#), [17](#), [31](#)

`ellipseGoogleMaps`, [5](#), [7](#), [12](#), [16](#), [21](#), [22](#), [26](#)
`elSPDF`, [6](#), [10](#), [11](#), [17](#), [31](#)

`iconlabels`, [11](#)

`mcGoogleMaps`, [13](#)

`pieSP`, [6](#), [11](#), [16](#), [17](#), [31](#)

`plotGoogleMaps`, [5](#), [10](#), [12](#), [16](#), [18](#), [22](#), [26](#), [27](#),
[29](#)

`PolyCol`, [21](#)

`segmentGoogleMaps`, [5](#), [10](#), [16](#), [21](#), [23](#)

`Spatial-class`, [14](#), [19](#)

`SpatialPointsDataFrame-class`, [3](#), [6](#), [8](#), [11](#),
[17](#), [24](#), [30](#)

`stfdfGoogleMaps`, [26](#), [29](#)

`stplotGoogleMaps`, [27](#), [28](#)

`vectorsSP`, [30](#)