

# Package ‘GeoXp’

January 18, 2012

**Type** Package

**Title** Interactive exploratory spatial data analysis

**Encoding** latin1

**Version** 1.5.4

**Date** 2012-01-18

**Author** Yves Aragon, Thibault Laurent, Lauriane Robidou, Anne Ruiz-Gazen, Christine Thomas-Agnan

**Maintainer** Thibault Laurent <thibault.laurent@univ-tlse1.fr>

**Depends** R (>= 2.7.0), fields, KernSmooth, mvoutlier, rgl, robustbase,spdep, splancs, stats, tcltk

**Description** GeoXp is a tool for researchers in spatial statistics,spatial econometrics, geography, ecology etc allowing to link dynamically statistical plots with elementary maps. This coupling consists in the fact that the selection of a zone on the map results in the automatic highlighting of the corresponding points on the statistical graph or reversely the selection of a portion of the graph results in the automatic highlighting of the corresponding points on the map. GeoXp includes tools from different areas of spatial statistics including geostatistics as well as spatial econometrics and point processes. Besides elementary plots like boxplots,histograms or simple scatterplos, GeoXp also couples with maps Moran scatterplots, variogram cloud, Lorentz Curves,...In order to make the most of the multidimensionality of the data, GeoXp includes some dimension reduction techniques such as PCA.

**License** GPL (>= 2)

**URL** [http://www.tse-fr.eu/images/doc/wp/etrie/wp\\_etrie\\_99\\_2009.pdf](http://www.tse-fr.eu/images/doc/wp/etrie/wp_etrie_99_2009.pdf)

**Repository** CRAN

**Date/Publication** 2012-01-18 20:24:04

**R topics documented:**

GeoXp-package . . . . .	3
angleplotmap . . . . .	4
barmap . . . . .	7
barnbmap . . . . .	9
boxplotmap . . . . .	11
carte . . . . .	12
choix.bubble . . . . .	14
choix.couleur . . . . .	15
choixvarfunc . . . . .	16
clustermap . . . . .	16
dbledensitymap . . . . .	19
dblehistsmap . . . . .	21
densitymap . . . . .	23
driftmap . . . . .	25
genpca . . . . .	27
gini . . . . .	30
ginimap . . . . .	31
graphique . . . . .	32
histnbmap . . . . .	34
histobarmap . . . . .	37
histomap . . . . .	39
immob . . . . .	41
makeneighborsw . . . . .	42
misolationmap . . . . .	44
moranplotmap . . . . .	46
mp.school . . . . .	48
mvariocloudmap . . . . .	49
neighbourmap . . . . .	51
nonormmoran . . . . .	54
normw . . . . .	55
pcamap . . . . .	57
plot3dmap . . . . .	59
polyboxplotmap . . . . .	61
polylist2list . . . . .	63
rotation . . . . .	64
scattermap . . . . .	65
selectgraph . . . . .	67
selectmap . . . . .	68
selectstat . . . . .	69
slider1 . . . . .	70
spdf2list . . . . .	71
variocloudmap . . . . .	72

## Description

GeoXp is a tool for researchers in spatial statistics, spatial econometrics, geography, ecology etc allowing to link dynamically statistical plots with elementary maps. This coupling consists in the fact that the selection of a zone on the map results in the automatic highlighting of the corresponding points on the statistical graph or reversely, the selection of a portion of the graph results in the automatic highlighting of the corresponding points on the map. GeoXp includes tools from different areas of spatial statistics including geostatistics as well as spatial econometrics and point processes. Besides elementary plots like boxplots, histograms or simple scatterplots, GeoXp also couples with maps Moran scatterplots, variogram cloud, Lorentz Curves,... In order to make the most of the multidimensionality of the data, GeoXp includes some dimension reduction techniques such as PCA.

## Details

Package: GeoXp  
 Type: Package  
 Version: 1.5.4  
 Date: 2012-01-18  
 License: GPL Vesion 2 or later

In the version 1.5.0, GeoXp has adopted the `SpatialClass` object proposed by Roger Bivand in `sp` package. The main advantage of using this structure object is on one hand, a `SpatialClass` object can contain both spatial coordinates and a `data.frame` of observed variable and on an other hand, it offers the possibility to make spatial analysis using both packages derived from `sp` as `spdep`, `gstat` and `GeoXp`.

On the map, the coordinates of sites are represented by using the function `coordinates` included in `sp` package, which calculates longitude (for x-axis) and latitude (for y-axis), applied on a `Spatial Class Object`.

In `GeoXp`, we can find three main groups of functions:

- functions using only one variable: the interest variable is designed by argument `name.var`, a character corresponding to a column of the `data.frame` included in `sp.obj`, i.e. the `Spatial Class` object. It can be a numeric variable (`histomap()`, `densitymap()`, `angleplotmap()`) or a factor variable (or character) (`barmap()`,...).
- functions using both several variables: the variables of interest are designed by argument `names.var`, a vector of character corresponding to columns of the `data.frame` included in `sp.obj`. It can be two numeric variables (`dbلهistomap`, `dbلهdensitymap`), one numeric variable and one factor (`histobarmap()`, `polyboxplotmap()`), several numeric variables (`plot3dmap`, `pcamap()` and `clustermap()`).
- functions using both a variable and a spatial weight matrix created as a `nb` or `listw` object (see package `spdep`).

In the case where `sp.obj` is a `SpatialPolygonDataFrame`, user will have the opportunity to draw

the polygons of Spatial unit by using the Draw Spatial contours button in the Tk window. User can also give a spatial polygonal contour as background map with option `carte`: in this case, a spatial polygonal contour is a matrix of numeric values with 2 columns (x and y coordinates of the vertices of the polygons) where polygons are separated from each other by 3 rows of NaN. The functions (`polylist2list()` and `spdf2list()`) convert some spatial objects (`Polylist` and `SpatialPolygonDataFrame`) into matrix as described above to draw a background map.

Among options which are common to each function, users have the possibility to give a `criteria`, vector of boolean of size the number of Spatial units, with TRUE on specific sites. These sites are then represented by a green croice on the map by clicking on `preselected sites` button on the Tk window.

Moreover, users have the possibility to make bubbles and add some graphs (histogram, barplot or scattermap). The potential variables are included in the `data.frame` of the `SpatialObject`. Users can choose a proportional symbol mapping: in function `plot`, we give value  $var^{0.5}$ . User can choose if a legend has to appear on the map. He could choose then three values represented by bubbles of corresponding sizes.

Finally, users can choose to represent the graphical with different colors using argument `col`. In the case of factors (as function `barmap`), users could choose if a legend with corresponding colors will appear on the map. Users can also modify the representation of selected sites on map with argument `pch`.

Recent functions `barnbmap` and `histnbmap` give the opportunity to analyse spatial weight matrix build using functions included in `spdep` package.

### Author(s)

Christine Thomas-Agnan, Yves Aragon, Anne Ruiz-Gazen, Thibault Laurent, Laurianne Robidou  
Maintainer: Thibault Laurent <thibault.laurent@univ-tlse1.fr>

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

---

angleplotmap

*Detection of an eventual directional trend*

---

### Description

The function `angleplotmap()` is used to detect an eventual directional trend associated to variable `name.var`. It represents the absolute difference between the value of `name.var` at two sites as a function of the angle between vector  $\overrightarrow{s_i s_j}$  and the x-axis.

**Usage**

```
angleplotmap(sp.obj, name.var, quantiles=NULL,
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightblue",
xlab="angle", ylab="absolute magnitude", axes=FALSE, lablong="", lablat="")
```

**Arguments**

sp.obj	object of class extending Spatial-class
name.var	a character; attribute name or column number in attribute table
quantiles	a vector of numeric corresponding to quantile orders (the regression quantile is obtained by spline smoothing)
names.attr	a vector of character: names to use in panel (if different from the names of variable used in sp.obj)
criteria	a vector of boolean which permit to represent preselected sites with a cross, using the tcltk window (must be equal to the number of spatial units)
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	16 by default, symbol for selected points
col	"lightblue3" by default, color of bars on the cloud map
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

**Details**

For each couple of sites  $(s_i, s_j)$ , the graphic represents on the y-axis the absolute difference between  $var_i$  and  $var_j$  :

$$D_{ij} = |var_i - var_j|$$

and on the x-axis the angle  $\theta_{ij}$  between  $\overline{s_i s_j}$  and the x-axis. Possibility to represent a smoothing spline regression quantile  $g_\alpha$ . For  $0 < \alpha < 1$ ,

$$Pr[D_{ij} < g_\alpha(\theta_{ij})] = \alpha$$

If that case, only the pair of sites  $(s_i, s_j)$  verifying :

$$D_{ij} > g_{max(\alpha)}(\theta_{ij})$$

are represented.

**Value**

In the case where user click on save results button, a matrix of integer is created as a global variable in last.select object. It corresponds to the numbers of spatial unit corresponding to couple of sites selected just before leaving the Tk window.

**Author(s)**

Thomas-Agnan Christine, Aragon Yves, Ruiz-Gazen Anne, Laurent Thibault, Robidou Lauriane

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

**See Also**

[variocloudmap](#), [driftmap](#)

**Examples**

```
#####
# data Conflicts Africa
data(afcon)

# afcon is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
afcon.sp = SpatialPoints(cbind(afcon$x,afcon$y))
# ... and then by integrating other variables to create SpatialPointsDataFrame
afcon.spdf = SpatialPointsDataFrame(afcon.sp, afcon)
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
africa <- readShapePoly(system.file("shapes/Africa.shp", package = "GeoXp")[1])
africa.contour<-spdf2list(africa)$poly

# A basic call of histomap function
angleplotmap(afcon.spdf,"totcon", carte= africa.contour,
identify=TRUE, cex.lab=0.6)

#####
# Data Meuse
data(meuse)

# meuse is a data.frame object. We have to create
```

```

# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
meuse.sp = SpatialPoints(cbind(meuse$x,meuse$y))
# ... and then by integrating other variables to create SpatialPointsDataFrame
meuse.spdf = SpatialPointsDataFrame(meuse.sp, meuse)

# meuse.riv is used for contour plot
data(meuse.riv)

angleplotmap(meuse.spdf,"copper",
col="green",quantiles=0.9, cex.lab=0.7,
xlab="Concentration in plomb (in ppm)",pch=7,carthe=meuse.riv[c(21:65,110:153),])

```

barmap

*Interactive Bar plot and map***Description**

The function `barmap()` draws a bar plot (vertical bar) of the given factor variable `name.var` and a map with sites of coordinates `coordinates(sp.obj)`.

**Usage**

```

barmap(sp.obj, name.var, type = c("count", "percent"),
names.arg="", names.attr=names(sp.obj), criteria=NULL, carthe=NULL, identify=FALSE, cex.lab=0.8, pch=1
col="lightblue3", xlab="", ylab="", axes=FALSE, lablong="", lablat="")

```

**Arguments**

<code>sp.obj</code>	object of class extending Spatial-class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>type</code>	Character string indicating type of histogram to be drawn. "percent" and "count" give relative frequency and frequency histograms.
<code>names.arg</code>	a vector of level names of <code>name.var</code>
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the tcltk window
<code>carthe</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	a vector of symbol which must be equal to the number of level else all sites are printed with <code>pch[1]</code>

col	a vector of colors which must be equal to the number of level else all sites and all bars are printed with col[1]
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

### Details

The selection of a bar on the bar plot results in the corresponding sites coloured on the map with the corresponding colour observed on the bar. Reversely, the selection of sites on the map by "points" or "polygon" results in the drawing of the sub-barplot in red.

### Value

In the case where user click on save results button, a vector of integer is created as a global variable in last.select object. It corresponds to the number of spatial units selected just before leaving the Tk window.

### Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

### See Also

[histomap](#), [histobarmap](#), [scattermap](#), [densitymap](#)

### Examples

```
#####
# data eire
eire <- readShapePoly(system.file("etc/shapes/eire.shp", package="spdep")[1],
ID="names", proj4string=CRS("+proj=utm +zone=30 +units=km"))

# a basic usage ...
barmap(eire,"pale")

# ... with all options
barmap(eire,3, type = "percent",col=c("pink","orange"),
```

```
names.arg=c("not pale","pale"), names.attr=names(eire),
criteria=NULL, identify=TRUE, cex.lab=0.8, pch=c(10,11),
xlab="Are majority people are pale ?", ylab="Percent",
axes=TRUE, lablong="longitude", lablat="latitude")
```

barnbmap

*Bar plot of the number of neighbour and map***Description**

The function `barnbmap` draws the bar plot of the link number distribution for a neighbourhood structure given by a `nb` object (`spdep` package) and links this bar plot with a map.

**Usage**

```
barnbmap(sp.obj, nb.obj,
criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightblue3",
xlab="", ylab="", axes=FALSE, lablong="", lablat="")
```

**Arguments**

<code>sp.obj</code>	object of class extending <code>Spatial</code> -class
<code>nb.obj</code>	object of class <code>nb</code>
<code>criteria</code>	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the <code>tcltk</code> window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not <code>FALSE</code> , identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	16 by default, symbol for selected points
<code>col</code>	"lightblue3" by default, color of bars on the barplot
<code>xlab</code>	a title for the graphic x-axis
<code>ylab</code>	a title for the graphic y-axis
<code>axes</code>	a boolean with <code>TRUE</code> for drawing axes on the map
<code>lablong</code>	name of the x-axis that will be printed on the map
<code>lablat</code>	name of the y-axis that will be printed on the map

**Details**

For a selected site  $j$  on the map, are represented on the map its neighbours. For a selected bar on the graph, the corresponding sites are represented on the map with a link which means that two sites are neighbours.

**Value**

No value returned.

**Note**

When user select sites on the graph or on the map, he cannot add a selection by using the other graphic.

**Author(s)**

Thomas-Agnan C., Ruiz-Gazen A., Laurent T.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

**See Also**

[moranplotmap](#), [makeneighborsw](#), [normw](#), [nonormmoran](#)

**Examples**

```
#####
# data on price indices of real estate in France
data(immob)
row.names(immob)<-immob$Nom

# immob is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
immob.sp = SpatialPoints(cbind(immob$longitude,immob$latitude))

# Spatial weight matrix based on nearest neighbours
immob.nb <- dnearneigh(coordinates(immob.sp), 0,175000)

# a simple use of barnbmap
barnbmap(immob.sp,immob.nb)

#####
# Data columbus
example(columbus)

# col.gal.nb is a spatial weight matrix included in spdep package...
barnbmap(columbus,col.gal.nb,criteria=(columbus$EW==1),
col=colors()[98:106], identify=TRUE, cex.lab=0.7, pch=1:9)
```

---

boxplotmap                      *Interactive boxplot and map*

---

### Description

The function `boxplotmap()` draws a boxplot of the given variable `name.var` and a map with site of coordinates `coordinates(sp.obj)`.

### Usage

```
boxplotmap(sp.obj, name.var,
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightblue3",
xlab="", ylab="", axes=FALSE, lablong="", lablat="")
```

### Arguments

<code>sp.obj</code>	object of class extending Spatial-class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of spatial sites, which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	16 by default, symbol for selected points
<code>col</code>	"lightblue3" by default, color of bars on the boxplot
<code>xlab</code>	a title for the graphic x-axis
<code>ylab</code>	a title for the graphic y-axis
<code>axes</code>	a boolean with TRUE for drawing axes on the map
<code>lablong</code>	name of the x-axis that will be printed on the map
<code>lablat</code>	name of the y-axis that will be printed on the map

### Details

There is an interactivity only in one direction : the sites selected by interquartile on the boxplot are represented on the map in red.

### Value

In the case where user click on `save results` button, a vector of integer is created as a global variable in `last.select` object. It corresponds to the number of spatial units selected just before leaving the Tk window.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

**See Also**

[histomap](#), [histobarmap](#), [scattermap](#), [densitymap](#)

**Examples**

```
#####  
# data columbus  
example(columbus)  
  
# a basic use of boxplotmap  
boxplotmap(columbus,"CRIME", criteria=(columbus@data$CP==1),  
xlab="Crime",ylab="Percent",identify=TRUE, cex.lab=0.7)  
  
#####  
# data boston  
data(boston)  
  
# creation of a Spatial object  
boston.sp = SpatialPoints(cbind(boston.c$LON,boston.c$LAT))  
# ... and then by integrating other variables to create SpatialPointsDataFrame  
boston.spdf = SpatialPointsDataFrame(boston.sp, boston.c)  
  
# a simple use of boxplotmap  
boxplotmap(boston.spdf,"MEDV",criteria=(boston.c$CHAS==1))
```

---

carte

*Drawing a map*

---

**Description**

The function `carte()` draws a map with sites of coordinates (long,lat) and represents sites which have been selected in obs. This function is called by most of the functions of GeoXp (this is not an interactive function).

**Usage**

```
carte(long, lat, obs, criteria=NULL, bubble=FALSE, cbubble=NULL, nointer=FALSE,
carte=NULL, nocart=FALSE, label="", cex.lab=NULL, symbol=16, lablong="",
lablat="", method="", W=NULL, couleurs="blue", classe=NULL, legmap=NULL,
legends=list(FALSE, FALSE), labmod="", axis=FALSE)
```

**Arguments**

long	a vector x of size n
lat	a vector y of size n
obs	a boolean vector of size n with TRUE on selected sites
criteria	a vector of size n of boolean with TRUE on specific sites (these for non interactive selection)
bubble	a boolean with TRUE for drawing bubbles, FALSE otherwise
cbubble	vector of size n with size of each site depending on variable with which bubbles are constructed
nointer	a boolean with TRUE for drawing sites selected by criteria, FALSE otherwise
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
nocart	a boolean with TRUE for drawing spatial contours, FALSE otherwise
label	vector of character of size n with name of each site
cex.lab	character size of label
symbol	a vector of symbol which must be equal to the number of group else all sites are printed in pch[1]
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map
method	Specification for some graphics such as 'Neighbourplot1', 'Neighbourplot2', 'Cluster', 'Angleplot', 'Variocloud'
W	neighbourhood matrix of size $n \times n$ , necessary when method='Neighbourplot'
couleurs	Colors of the sites
classe	vector of n with class of sites
legmap	a list of (numeric,numeric,numeric,numeric,numeric,numeric,character) with the sizes and the corresponding values of the 3 bubbles represented in the legend and the variable name of variable choosed
legends	a list of c(boolean,boolean,c(numeric,numeric),c(numeric,numeric)) with TRUE for drawing legends of bubbles (1st argument) or factors (2nd argument) and the coordinates of the location of the upper left corner of the legend box
labmod	Name of factors
axis	a boolean with True for drawing axes on the map

**Details**

Options permit to represent sites with different colors or symbols, with different sizes of circle ('bubble'), using a spatial contour ('carte'), printing names of sites ('label'),etc. Possibility to draw a legend giving the sizes of bubbles and colors of levels.

**Value**

No values, only drawing of a map

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

---

choix.bubble

*Selection of a variable before plotting bubbles*

---

**Description**

The function `choix.bubble()` is used in most of the GeoXp functions to initialize some parameters before plotting bubbles.

**Usage**

```
choix.bubble(buble,listvar,listnomvar,legends)
```

**Arguments**

buble	boolean
listvar	list of variables
listnomvar	names of variables
legends	parameters of plot

**Details**

This function is not an interactive function.

**Value**

A list of parameters which permit to use the function `carte`.

**Author(s)**

Laurent T.

## References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

---

choix.couleur	<i>Selection of colors before plotting map</i>
---------------	--

---

## Description

The function `choix.couleur()` is used in most of the GeoXp functions to initialize some parameters before plotting map in the case of the use of a factor.

## Usage

```
choix.couleur(graphChoice,listvar=NULL,listnomvar=NULL,  
varChoice1=NULL,legends,col,pch)
```

## Arguments

<code>graphChoice</code>	kind of graphic chosen
<code>listvar</code>	list of variables
<code>listnomvar</code>	names of variables
<code>varChoice1</code>	the name of the chosen variable
<code>legends</code>	parameters of plot
<code>col</code>	color of plot
<code>pch</code>	symbols

## Details

This function is not an interactive function.

## Value

A list of parameters which permit to use the function `carte`.

## Author(s)

Laurent T.

## References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

---

choixvarfunc	<i>Selection of a variable before plotting bubbles</i>
--------------	--

---

**Description**

The function `choixvarfunc()` is used in the function `choix.bubble()`.

**Usage**

```
choixvarfunc(title, question, liste)
```

**Arguments**

title	a character as "Choice of variables"
question	a character as "Choose a variable"
liste	a list of character with names of variables

**Details**

This function is not an interactive function.

**Value**

Name of variable chosen.

**Author(s)**

Laurent T.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

---

clustermap	<i>Classification of dataset using kmeans or hclust algorithm and representation of clusters on a map.</i>
------------	--

---

**Description**

The function `clustermap()` performs a classification of the sites from the variables called in `names.var` and computes a bar plot of the clusters calculated. Classification methods come from `hclust()` (hierarchical cluster analysis) and `kmeans()` (k-means clustering) and number of class is chosen with `clustnum`.

**Usage**

```
clustermap(sp.obj, names.var, clustnum, method=c("kmeans", "hclust"), type=NULL, centers=NULL, scale=FA
names.arg="", names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=1
col="lightblue3", xlab="Cluster", ylab="Number", axes=FALSE, lablong="", lablat="")
```

**Arguments**

sp.obj	object of class extending Spatial-class
names.var	a vector of character; attribute names or column numbers in attribute table
clustnum	integer, number of clusters
method	two methods : 'kmeans' by default or 'hclust'
type	If method='hclust', type='complete' by default (the possibilities are given in help(hclust) as 'ward', 'single', etc). If method='kmeans', type="Hartigan-Wong" by default (the possibilities are given in help(kmeans) as 'Forgy', etc)
centers	If method='kmeans', user can give a matrix with initial cluster centers.
scale	If scale=TRUE, the dataset is reduced.
names.arg	a vector of character, names of cluster
names.attr	names to use in panel (if different from the names of variable used in sp.obj)
criteria	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the tcltk window
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	a vector of symbol which must be equal to the number of group else all sites are printed in pch[1]
col	a vector of colors which must be equal to the number of group else all sites and all bars are printed in col[1]
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

**Details**

The two windows are interactive : the sites selected by a bar chosen on the bar plot are represented on the map in red and the values of sites selected on the map by 'points' or 'polygon' are represented in red on the bar plot. The dendrogram is also drawn for 'hclust' method. In option, possibility to choose the classification method.

**Value**

In the case where user click on save results button, a list is created as a global variable in `last.select` object. `obs`, a vector of integer, corresponds to the number of spatial units selected just before leaving the Tk window, `vectclass`, vector of integer, corresponds to the number of cluster attributed to each spatial unit.

**Note**

To use the functions `hclust` and `kmeans`, we take many arguments by default. If the user would like to modify these arguments, he should call these functions first and then use the function `barmap` to visualize the calculated clusters.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Murtagh, F (1985). *Multidimensional Clustering Algorithms*.

Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics* 28, 100-108

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

**See Also**

[barmap](#), [pcamap](#)

**Examples**

```
#####
# data columbus
example(columbus)

# a basic example using the kmeans method
clustermap(columbus, c("HOVAL", "INC", "CRIME", "OPEN", "PLUMB", "DISCBD"), 3,
criteria=(columbus@data$CP==1), identify=TRUE, cex.lab=0.7)

# example using the hclust method
clustermap(columbus, c(7:12), 3, method="hclust",
criteria=(columbus@data$CP==1), col=colors()[20:22], identify=TRUE,
```

```
cex.lab=0.7, names.arg=c("Group 1","Group 2","Group 3"), xlab="Cluster")
```

---

 dbledensitymap

*Double Kernel density estimates and map*


---

## Description

The function `dbledensitymap` plots two kernel density estimates from 2 variables included in `names.var` computed with `bkde`, and a map with sites of coordinates `coordinates(sp.obj)`. Each site is associated to a value of `names.var[1]` and `names.var[2]` and the two windows are interactive.

## Usage

```
dbledensitymap(sp.obj, names.var, kernel='triweight',
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16,
col=c("grey","lightblue3"), xlab=c("", ""), ylab="", axes=FALSE, lablong="", lablat="")
```

## Arguments

<code>sp.obj</code>	object of class extending <code>Spatial</code> -class
<code>names.var</code>	a vector of character of size 2; attribute names or column numbers in attribute table
<code>kernel</code>	Smoothing kernel (see <code>help(bkde)</code> for list of options)
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the <code>tcltk</code> window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not <code>FALSE</code> , identify plotted objects (currently only working for points plots). Labels for identification are the <code>row.names</code> of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	16 by default, symbol for selected points
<code>col</code>	<code>c("grey","lightblue3")</code> by default, color of the two density curves
<code>xlab</code>	a list of title for the two x-axis graphics
<code>ylab</code>	a list of title for the two y-axis graphics
<code>axes</code>	a boolean with <code>TRUE</code> for drawing axes on the map
<code>lablong</code>	name of the x-axis that will be printed on the map
<code>lablat</code>	name of the y-axis that will be printed on the map

## Details

The user can choose an interval on the density curve by mouse clicking on the lower and upper boundaries of the interval or by giving directly these values. The selected sites are then represented on the map in red. A selection by 'points' or 'polygon' on the map results in the drawing of the density of the corresponding sub-distribution on the density plot. Finally, the user can modify the bandwidth parameter with a cursor in the tk window (parameter  $\alpha$ ).  $\alpha$  is the smoothing parameter for the kernel smooth : it represents the mean percentage of sample points involved in the local averaging (example :  $\alpha = 20$  means that on average,  $n \times 0.2$  points are in any interval of length  $2h$  where  $h$  is the usual bandwidth).

## Value

In the case where user click on save results button, a vector of integer is created as a global variable in last.select object. It corresponds to the number of spatial units selected just before leaving the Tk window.

## Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

## References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*. New York: Springer.

Wand M.P. et Jones M.C. (1995), *Kernel Smoothing*, Chapman & Hall.

## See Also

[histomap](#), [histobarmap](#), [scattermap](#), [densitymap](#)

## Examples

```
#####
# data auckland
data(auckland)

# creation of a Spatial object
auckland.sp = SpatialPoints(cbind(auckland$Easting,auckland$Northing))
# ... and then by integrating other variables to create SpatialPointsDataFrame
```

```

auckland.spdf = SpatialPointsDataFrame(auckland.sp, auckland)
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
contours.auckland<-polylist2list(auckpolys)

dbledensitymap(auckland.spdf, c("Deaths.1977.85", "Under.5.1981"), carte=contours.auckland,
xlab=c("Deaths.1977.85", "Under.5.1981"),
criteria=(auckland$Deaths.1977.85>mean(auckland$Deaths.1977.85)))

#####
# data eire
eire <- readShapePoly(system.file("etc/shapes/eire.shp", package="spdep")[1],
ID="names", proj4string=CRS("+proj=utm +zone=30 +units=km"))

dbledensitymap(eire,c("A", "towns"),kernel="normal",
xlab=c("Individuals rate of blood type A",
"Surface urbaine"),identify=TRUE)

```

---

dblehistomap

*Double Interactive Histogram and map*


---

## Description

The function `dblehistomap` draws two histograms of the given variables `names.var[1]` and `names.var[2]` and a map with sites of coordinates `coordinates(sp.obj)`. Each site is associated to a value of `names.var[1]` and `names.var[2]` and there is interactivity between the two windows created.

## Usage

```

dblehistomap(sp.obj, names.var, nbcol=c(10,10), type = c("count", "percent", "density"),
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16,
col=c("grey", "lightblue3"), xlab=c("", ""), ylab=c("count", "count"), axes=FALSE, lablong="", lablat=""

```

## Arguments

<code>sp.obj</code>	object of class extending Spatial-class
<code>names.var</code>	a vector of 2 characters; attribute name or column number in attribute table
<code>nbcol</code>	a vector of integer indicating number of cells for histogram 1 and histogram 2 (10 for each by default)
<code>type</code>	Character string indicating type of histogram to be drawn. "percent" and "count" give relative frequency and frequency histograms, "density" produces a density scale histogram.
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )

criteria	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the tcltk window
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	16 by default, symbol for selected points
col	a vector of character indicating colors for the bars on the histogram 1 and histogram 2
xlab	a vector of character containing titles for the two graphics x-axis
ylab	a vector of character containing titles for the two graphics y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

### Details

The selection of sites by ‘points’ or ‘polygons’ on the map results in the drawing of the red histograms of the subdistributions corresponding to this subset of sites.

### Value

In the case where user click on save results button, a vector of integer is created as a global variable in last.select object. It corresponds to the number of spatial units selected just before leaving the Tk window.

### Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Économétrie pour données géoréférencées : modèles et études de cas*

### See Also

[dblehistomap](#), [histobarmap](#), [scattermap](#), [densitymap](#)

### Examples

```
#####
# data on price indices of real estate in France
data(immob)
row.names(immob)<-immob$Nom
```

```

# immob is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
immob.sp = SpatialPoints(cbind(immob$longitude,immob$latitude))
# ... and then by integrating other variables to create SpatialPointsDataFrame
immob.spdf = SpatialPointsDataFrame(immob.sp, immob)
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
midiP <- readShapePoly(system.file("shapes/region.shp", package="GeoXp")[1])
cont_midiP<-spdf2list(midiP[-c(22,23),])$poly

# A basic call of dblehistomap function
dblehistomap(immob.spdf,c("prix.vente","prix.location"),
carte= cont_midiP, identify=TRUE, cex.lab=0.6)

###
# data colombus
x <- readShapePoly(system.file("shapes/columbus.shp", package="maptools")[1])

# example of use with many options
dblehistomap(x,c("HOVAL","CRIME"), nbcol=c(5,10),type="percent",
xlab=c("hoval","crime"),identify=TRUE, cex.lab=0.7, ylab=c("percent","percent"),
col=c("pink","orange"),pch=14,axes=TRUE)

```

---

densitymap

*Kernel density estimates and map*


---

## Description

The function `densitymap` draws kernel density estimates of the variable `name.var` with `bkde` and a map with sites of coordinate coordinates (`sp.obj`). Each site is associated to a value of `name.var` and there is interactivity between the two windows.

## Usage

```

densitymap(sp.obj, name.var, kernel='triweight',
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16,
col="lightblue3", xlab="", ylab="", axes=FALSE, lablong="", lablat="")

```

## Arguments

<code>sp.obj</code>	object of class extending <code>Spatial</code> -class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>kernel</code>	Smoothing kernel (see <code>help(bkde)</code> for list of options)
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )

criteria	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the tcltk window
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	16 by default, symbol for selected points
col	"lightblue3" by default, color of bars on the histogram
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

### Details

The user can choose an interval on the density curve by mouse clicking on the graph on the extremities of interval or by specifying directly values. The sites selected by an interval are then represented on the map in red. The selection of sites on the map by 'points' or 'polygon' results in the drawing of the kernel densities of the subdistributions corresponding to this subset of sites. Finally, the user can modify the bandwidth parameter with a cursor in the Tk window (parameter  $\alpha$ ).  $\alpha$  is the smoothing parameter for the kernel smooth : it represents the mean percentage of sample points involved in the local averaging (example :  $\alpha = 20$  means that on average,  $n \times 0.2$  points are in any interval of length  $2h$  where  $h$  is the usual bandwidth).

### Value

In the case where user click on save results button, a vector of integer is created as a global variable in last.select object. It corresponds to the number of spatial units selected just before leaving the Tk window.

### Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*. New York: Springer.

Wand M.P. et Jones M.C. (1995), *Kernel Smoothing*, Chapman & Hall.

**See Also**

[histomap](#), [histobarmap](#), [scattermap](#), [densitymap](#)

**Examples**

```
#####
# data oldcol
example(columbus)

# columbus is included in the Spatial-Class object
# a very simple use of histomap :
densitymap(columbus, "CRIME")

#####
# data on price indices of real estate in France
data(immob)
row.names(immob)<-immob$Nom

# immob is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
immob.sp = SpatialPoints(cbind(immob$longitude,immob$latitude))
# ... and then by integrating other variables to create SpatialPointsDataFrame
immob.spdf = SpatialPointsDataFrame(immob.sp, immob)
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
midiP <- readShapePoly(system.file("shapes/region.shp", package="GeoXp")[1])
cont_midiP<-spdf2list(midiP[-c(22,23),])$poly

# A basic call of densitymap function
densitymap(immob.spdf,"prix.vente", carte= cont_midiP, identify=TRUE,
xlab="housing price by square meter", cex.lab=0.6)
```

---

driftmap

*Interactive driftplot and map*


---

**Description**

The function `driftmap()` creates two devices : a device with the map of coord coordinates (`sp.obj`) which permits to make selection of sites and a device divided into 2 rows and 2 columns which contains : (cell 1) the selected sites divided into `m` rows and `q` columns (`m` and `q` are selected with the tk window), (cell 2) a scatter plot with coordinates(`sp.obj`)[,2] in x-axis and the mean and median of `name.var` calculated for the `m` rows in y-axis, (cell 3) a scatter plot with the mean and median of `name.var` calculated for the `q` columns in x-axis and coordinates(`sp.obj`)[,1] in y-axis and (cell 4) a legend indicating the direction of the North, the variable used, etc.

**Usage**

```
driftmap(sp.obj, name.var, interpol=TRUE, nuage=TRUE, lty=1:2, cex=0.7,
names.attr=names(sp.obj), carte=NULL, identify=FALSE, cex.lab=0.8, pch=rep(16,3),
col=c("lightblue3", "black", "red"), xlab="", axes=FALSE)
```

**Arguments**

<code>sp.obj</code>	object of class extending Spatial-class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>interpol</code>	if TRUE, the mean and median calculated are linearly interpolated
<code>nuage</code>	if TRUE, the values taken by var are also represented on right plot and on the left plot
<code>lty</code>	the line type for mean and median in the case of <code>interpol=TRUE</code>
<code>cex</code>	the amount by which plotting symbols on the cell 1 and in the case of <code>nuage=TRUE</code> cell 2 and cell 3
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	<code>c(16,16,16)</code> by default, list of symbols for (1) the values of var on the left and on the right plot (2) the mean points, (3) the median points
<code>col</code>	<code>c("lightblue3", "black", "red")</code> by default, list of colors for (1) the values of var on the left and on the right plot (2) the mean points, (3) the median points
<code>xlab</code>	name of var printed with the legend plot
<code>axes</code>	a boolean with TRUE for drawing axes on the map

**Details**

Possibility to change the number of cells in the grid with the tk window, to interpolate the means and medians calculated (by default), to work on a rotated map. At the beginning, all sites have been selected to the map : users have to deselect sites that they have choosen.

**Value**

In the case where user click on `save results` button, a vector of integer is created as a global variable in `last.select` object. It corresponds to the number of spatial units selected just before leaving the Tk window.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

## References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

## See Also

[angleplotmap](#), [variocloudmap](#), [rotation](#)

## Examples

```
#####
# data on price indices of real estate in France
data(immob)
row.names(immob)<-immob$Nom

# immob is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
immob.sp = SpatialPoints(cbind(immob$longitude,immob$latitude))
# ... and then by integrating other variables to create SpatialPointsDataFrame
immob.spdf = SpatialPointsDataFrame(immob.sp, immob)
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
midiP <- readShapePoly(system.file("shapes/region.shp", package="GeoXp")[1])
cont_midiP<-spdf2list(midiP[-c(22,23),])$poly

# a simple use of driftmap
driftmap(immob.spdf,"prix.vente", carte= cont_midiP)

# ... with options
driftmap(immob.spdf,"prix.vente", carte= cont_midiP,
interpol=FALSE, nuage=TRUE, lty=3:4,
identify=TRUE, cex.lab=0.3, xlab="Price of sell", axes=TRUE)

#####
# data eire
eire <- readShapePoly(system.file("etc/shapes/eire.shp", package="spdep")[1],
ID="names", proj4string=CRS("+proj=utm +zone=30 +units=km"))

# a basic usage ...
driftmap(eire,"pale",identify=TRUE,nuage=TRUE)
```

**Description**

The function `genpca` computes a generalized Principal Component Analysis (PCA). It calculates the principal components, the coordinates of the variables and in these principals components axes and the inertia of these principal components.

**Usage**

```
genpca(data, w=rep(1/nrow(data),length=nrow(data)),
m=diag(ncol(data)), center=NULL, reduc=TRUE)
```

**Arguments**

<code>data</code>	matrix $n \times p$
<code>w</code>	vector of size n of weight (by default : $weight = t(1/n, \dots, 1/n)$ )
<code>m</code>	matrix $p \times p$ (by default : metric=Identity matrix)
<code>center</code>	boolean. if TRUE, centered PCA (by default : center=TRUE)
<code>reduc</code>	boolean. if TRUE, reduced PCA (by default : reduce=TRUE)

**Details**

Let

$$W = \text{diag}(w)$$

$$x = \text{data} = (x'_1, \dots, x'_n)'$$

with

$$x_i = (x_i^1, \dots, x_i^p)$$

Let

$$1_n = (1, \dots, 1)'$$

with n rows and :

$$1_p = (1, \dots, 1)'$$

with p rows. Normalization of weight :

$$w_i = \frac{w_i}{\sum_i w_i}$$

Vector of means :

$$\bar{x} = (\bar{x}^1, \dots, \bar{x}^p)'$$

with:

$$\bar{x}^j = \sum_i w_i x_i^j$$

If center=True,

$$x_c = x - 1_n \bar{x}'$$

Standart deviation :

$$(\sigma^j)^2 = \sum_i w_i (x_i^j)^2 - (\bar{x}^j)^2$$

$$\Sigma = \text{diag}((\sigma^1)^2, \dots, (\sigma^p)^2)'$$

If `reduc=True` :

$$x_{cr} = x_c \times \Sigma^{-1/2}$$

Variance-Covariance matrix:

$$C = x_{cr}' W x_{cr}$$

Cholesky decomposition :  $M = LL'$  where  $M=m$

Let

$$C_1 = LCL'$$

Let U and D as :

$$C_1 U = U D$$

with  $D = \text{diag}(\lambda_1, \dots, \lambda_p)$

Let

$$V = L'U$$

Then :

Coordinates of individuals in the principals components basis :

$$CC = x_{cr} V$$

Coordinates of variables in principals components :

$$VC = CVD^{-1/2}$$

Inertia :

$$I = D1_p$$

### Value

Returns 'inertia' vector of size p with percent of inertia of each component (corresponding to I), 'casecoord' matrix  $n \times p$  (corresponding to matrix CC), 'varcoord' matrix  $p \times n$  (corresponding to matrix VC0).

### Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Caussinus H., Fekri M., Hakam S., Ruiz-Gazen A. (2003) , *A monitoring display of Multivariate Outliers*, Computational Statistics and Data Analysis, vol. 44, 1-2, 237-252.

### See Also

[clustermap](#), [pcamap](#)

---

gini	Calculates a Gini Index
------	-------------------------

---

**Description**

The function `gini()` calculates the Gini Index associated to the variable `var`.

**Usage**

```
gini(var)
```

**Arguments**

`var` a vector of numerical values of size `n`

**Details**

Let  $x_k$ ,  $k=1,\dots,K$  be the distinct values taken by `var`. For each site, `Gini.r` returns two pairs of frequencies. The pair  $(f,g)$  where  $f$  represents

$$f_k = \frac{1}{n} \sum_i 1(\text{Var}_i = x_k)$$

( $n$  is the length of `var`) and  $g$  represents

$$g_k = \frac{x_k}{\bar{x}} f_k$$

The pair  $(F,G)$  represents the corresponding cumulative frequencies.

The Gini Index is calculated as :

$$I_G = \frac{1}{2\bar{x}} \sum_{i=1}^K \sum_{j=1}^K |x_i - x_j|$$

**Value**

$(f, F, g, G, \text{gini})$  where  $f, F, g$  and  $G$  are vectors of size  $K$  and `gini` a numeric value.

**Note**

This function is used in `ginimap.R` but it is not an interactive function.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

**See Also**[ginimap](#)


---

ginimap	<i>Lorentz curve and map</i>
---------	------------------------------

---

**Description**

The function `ginimap()` computes a Lorentz curve from `name.var` and calculates the Gini Index associated to `name.var`.

**Usage**

```
ginimap(sp.obj, name.var, names.attr=names(sp.obj), criteria=NULL,
carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightblue3",
xlab="", ylab="", axes=FALSE, lablong="", lablat="")
```

**Arguments**

<code>sp.obj</code>	object of class extending Spatial-class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of Spatial units, which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	16 by default, symbol for selected points
<code>col</code>	"lightblue3" by default, color of bars on the histogram
<code>xlab</code>	a title for the graphic x-axis
<code>ylab</code>	a title for the graphic y-axis
<code>axes</code>	a boolean with TRUE for drawing axes on the map
<code>lablong</code>	name of the x-axis that will be printed on the map
<code>lablat</code>	name of the y-axis that will be printed on the map

**Details**

Users have the possibility to choose a threshold by mouse clicking on the Lorentz curve or by specifying it in the menu. The corresponding pair (F,G) and the value of `name.var` are then printed on the graph and the sites with a value of `name.var` lower or equal to the threshold are then selected on the map.

**Value**

In the case where user click on save results button, a vector of integer is created as a global variable in last.select object. It corresponds to the number of spatial units selected just before leaving the Tk window.

**Note**

The Gini Index is given in the tcltk window (see function gini for the formula used to calculate it).

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

**See Also**

[gini](#)

**Examples**

```
#####
# data eire
eire <- readShapePoly(system.file("etc/shapes/eire.shp", package="spdep")[1],
ID="names", proj4string=CRS("+proj=utm +zone=30 +units=km"))

# a basic usage ...
ginimap(eire,"INCOME")

# ... with options
ginimap(eire,"INCOME",criteria=(eire$pale==1),
identify=TRUE, pch=5, col="orange",
axes=TRUE, lablong="X", lablat="Y")
```

---

graphique

*Drawing a graphic*

---

**Description**

The function 'graphique' draws a specific graphic. Possibility to draw one of these graphics : Histogram, Barplot, Boxplot, Polyboxplot, Scatterplot, Densityplot, Neighbourplot, Lorentz, Moran, Angleplot, Variocloud, Cluster and PCA (graphics of individuals and variables). This function is called in most of the functions of GeoXp (this is not an interactive function).

**Usage**

```
graphique(var1, var2, var3, obs, num, graph = "", couleurs = "",
  symbol = 16, labvar = "", nbcoll = 10, alpha1, W,
  Xpoly, Ypoly, F, G, opt1 = 1, opt2 = 1, quantiles = 0,
  labmod = "", direct, inertie, label = 0, kernel, obsq, locmoran = FALSE,
  bin = NULL, cex.lab=1, bubble=FALSE, cbubble=NULL, legmap=NULL,
  legends=list(FALSE,FALSE), xlim, ylim)
```

**Arguments**

var1	1st variable of size n
var2	2nd variable of size n
var3	3rd variable of size n (used for variocloudmap)
obs	a vector of boolean of size n with sites selected
num	number of windows which must be activated (3 ou 4)
graph	name of graphic which must be drawn : Histogram, Barplot, Boxplot, Polyboxplot, Scatterplot, Densityplot1, Densityplot2, Neighbourplot, Lorentz, Moran, Quadrant, Angleplot, Variocloud, Cluster, Acp1, Acp2
couleurs	Possibility to change colors on the graphic
symbol	choice of representation of selected sites
labvar	name(s) of variable(s) studied
nbcoll	number of cells if the graphic chosen is the histogram (10 by default)
alpha1	regression smoothing parameter
W	Spatial weight matrix
Xpoly	x-coordinates of the vertices of selected polygon
Ypoly	y-coordinates of the vertices of selected polygon
F	Used for Ginimap
G	Used for Ginimap
opt1	Option for adding a curve (regression line or conditional quantile depending on the function)
opt2	Option for drawing robust empirical variogram
quantiles	vector which contains the values of $\alpha$ for conditional quantile
labmod	names of factor if the graphic chosen is a barplot
direct	Used for PCA
inertie	Used for PCA
label	Name of sites
kernel	Name of the kernel chosen in densitymap
obsq	Used for clustermap and barmap
locmoran	Print local moran for each site
bin	The bins chosen to calculate empirical variogram

<code>cex.lab</code>	character size of label
<code>buble</code>	For Local Moran plot only : a boolean with TRUE for drawing bubbles, FALSE otherwise
<code>cbuble</code>	For Local Moran plot only : vector of size n with size of each site depending on variable with which bubbles are constructed
<code>legmap</code>	For Local Moran plot only : a list of (numeric,numeric,numeric,numeric,numeric,numeric,character) with the sizes and the corresponding values of the 3 bubbles represented in the legend and the variable name of variable choosed
<code>legends</code>	For Local Moran plot only : a list of c(boolean,boolean,c(numeric,numeric),c(numeric,numeric)) with TRUE for drawing legends of bubbles (1st argument) or factors (2nd argument) and the coordinates of the location of the upper left corner of the legend box
<code>xlim</code>	the x limits of the plot
<code>ylim</code>	the y limits of the plot

### Details

This function is called by any function which draws a graphic. A lot of options are considered because of the large number of graphics proposed

### Value

No values, only drawing of a graphic

### Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

---

histnbmap	<i>Interactive histogram of the distances between two neighbors of a nb object and map</i>
-----------	--

---

### Description

The function `histnbmap()` draws the histogram of the distances between two neighbors for a neighbourhood structure given by a `nb` object and calculated by `nbdists` function (see `spdep` package), and links the graphic with a map.

**Usage**

```
histnbmap(sp.obj, nb.obj, longlat = NULL, nbc1=10, type = c("count", "percent", "density"),
sup=FALSE, criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightblue3",
xlab="", ylab="count", axes=FALSE, lablong="", lablat="")
```

**Arguments**

sp.obj	object of class extending Spatial-class
nb.obj	object of class nb
longlat	TRUE if point coordinates are longitude-latitude decimal degrees, in which case distances are measured in kilometers; if coords is a SpatialPoints object, the value is taken from the object itself
nbc1	number of cells for histogram (10 by default)
type	Character string indicating type of histogram to be drawn. "percent" and "count" give relative frequency and frequency histograms, "density" produces a density scale histogram.
sup	if TRUE, it keeps only the distance of the neighbor the farrest
criteria	a vector of boolean of size the number of Spatial units, which permit to represent preselected sites with a cross, using the tcltk window
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	16 by default, symbol for selected points
col	"lightblue3" by default, color of bars on the barplot
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

**Details**

For a selected site *j* on the map, are represented on the map its neighbours. For a selected bar on the graph, the corresponding sites are represented on the map with a link which means that two sites are neighbours.

**Value**

No value returned.

**Note**

When user select sites on the graph or on the map, he cannot add a selection by using the other graphic.

**Author(s)**

Aragon Y., Thomas-Agnan C., Ruiz-Gazen A., Laurent T.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

**See Also**

[moranplotmap](#), [makeneighborsw](#), [normw](#), [nonormmoran](#)

**Examples**

```
##
# data columbus
example(columbus)

# a simple use of histnbmap
histnbmap(columbus, col.gal.nb, criteria=(columbus$CP==1),
xlab="distance of the neighbor the farest")

##
# data meuse
data(meuse)

# meuse is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
meuse.sp = SpatialPoints(cbind(meuse$x,meuse$y))
# ... and then by integrating other variables to create SpatialPointsDataFrame
meuse.spdf = SpatialPointsDataFrame(meuse.sp, meuse)

# meuse.riv is used for contour plot
data(meuse.riv)

# creation of a spatial weight matrix (class nb) based
# on the Delaunay triangulation
meuse.nb <- tri2nb(coordinates(meuse.sp))

# a example with some optionswhich shows the limit of
# this kind of spatial weight matrix
```

```
histnbmap(meuse.spdf, meuse.nb, sup=TRUE, nbc=7,
  carte=meuse.riv[c(21:65,110:153),])
```

---

 histobarmap

*Histogram, barplot and map*


---

### Description

The function `histobarmap()` draws a bar plot (vertical bar) of the given variable names `.var[1]`, a histogram of the given variable names `.var[2]` and a map with sites of coordinates `coordinates(sp.obj)`.

### Usage

```
histobarmap(sp.obj, names.var, nbc = 10, type = "count",
  names.arg = "", names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE,
  cex.lab=0.8, pch=16, col="lightblue3", xlab=c("barplot", "histogram"), ylab=rep("count",2),
  axes=FALSE, lablong="", lablat="")
```

### Arguments

<code>sp.obj</code>	object of class extending Spatial-class
<code>names.var</code>	a vector of 2 characters; first character corresponds to the name of a factor and second character corresponds to a numeric value
<code>nbc</code>	number of cells for histogram (10 by default)
<code>type</code>	Character string indicating type of histogram/barplot to be drawn. "percent" and "count" give relative frequency and frequency histogram/barplot.
<code>names.arg</code>	a vector of level names (for factor)
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	a vector of symbol which must be equal to the number of group else all sites are printed in <code>pch[1]</code>
<code>col</code>	a vector of colors which must be equal to the number of group else all sites and all box are printed in <code>col[1]</code>
<code>xlab</code>	a vector of names for x-title of the barplot and histogram
<code>ylab</code>	a vector of names for y-title of the barplot and histogram
<code>axes</code>	a boolean with TRUE for drawing axes on the map
<code>lablong</code>	name of the x-axis that will be printed on the map
<code>lablat</code>	name of the y-axis that will be printed on the map

## Details

Each site is associated to a factor of `names.var[1]` and to a value of `names.var[2]`. There is interactivity between the three windows created: the sites selected by a bar on the bar plot or on the histogram are represented on the map in red and the value and factor of sites selected on the map are represented in red on the bar plot and on the histogram.

## Value

In the case where user click on `save results` button, a vector of integer is created as a global variable in `last.select` object. It corresponds to the number of spatial units selected just before leaving the Tk window.

## Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

## References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

## See Also

[dblehistomap](#), [histobarmap](#), [scattermap](#), [dbledensitymap](#)

## Examples

```
###
# Data Columbus
example(columbus)

# an example of use
histobarmap(columbus,c("CP","HOVAL"),nbc=8, type="percent",
names.arg=c("A","B"), xlab=c("CP","Hoval"), ylab=rep("percent",2))

#####
# data eire
eire <- readShapePoly(system.file("etc/shapes/eire.shp", package="spdep")[1],
ID="names", proj4string=CRS("+proj=utm +zone=30 +units=km"))

# example of use
histobarmap(eire, c("pale","A"),names.arg=c("Outside Pale","Pale"),
xlab=c("Appartenance to the region of Pale","Average number of people with blood A"),
col=colors()[101:102],identify=TRUE)
```

---

histomap	<i>Interactive Histogram and map</i>
----------	--------------------------------------

---

**Description**

The function `histomap()` draws a histogram of a given variable `name.var` and a map with sites of coordinates `coordinates(sp.obj)`. Each site is associated to a value of `name.var` and there is interactivity between the two windows.

**Usage**

```
histomap(sp.obj, name.var, nbccl=10, type = c("count", "percent", "density"),
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightblue3",
xlab="", ylab="", axes=FALSE, lablong="", lablat="")
```

**Arguments**

<code>sp.obj</code>	object of class extending Spatial-class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>nbccl</code>	number of cells for histogram (10 by default)
<code>type</code>	Character string indicating type of histogram to be drawn. "percent" and "count" give relative frequency and frequency histograms, "density" produces a density scale histogram.
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of size n of boolean which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	16 by default, symbol for selected points
<code>col</code>	"lightblue3" by default, color of bars on the histogram
<code>xlab</code>	a title for the graphic x-axis
<code>ylab</code>	a title for the graphic y-axis
<code>axes</code>	a boolean with TRUE for drawing axes on the map
<code>lablong</code>	name of the x-axis that will be printed on the map
<code>lablat</code>	name of the y-axis that will be printed on the map

**Details**

Sites selected by a bar on the histogram are represented on the map in red and the values of sites selected on the map by 'points' or 'polygon' are represented in red as a sub-histogram on the histogram.

**Value**

In the case where user click on save results button, a vector of integer is created as a global variable in last.select object. It corresponds to the number of spatial units selected just before leaving the Tk window.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

**See Also**

[histomap](#), [histobarmap](#), [scattermap](#), [densitymap](#)

**Examples**

```
#####
# data oldcol
example(columbus)

# columbus is included in the Spatial-Class object
# a very simple use of histomap :
histomap(columbus,"CRIME")

#####
# data on price indices of real estate in France
data(immob)

# immob is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
immob.sp = SpatialPoints(cbind(immob$longitude,immob$latitude))
# ... and then by integrating other variables to create SpatialPointsDataFrame
immob.spdf = SpatialPointsDataFrame(immob.sp, immob)
# we just give names to the spatial units...
row.names(immob.spdf)<-immob$Nom
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
midiP <- readShapePoly(system.file("shapes/region.shp", package="GeoXp")[1])
cont_midiP<-spdf2list(midiP[-c(22,23),])$poly
```

```
# A basic call of histomap function
histomap(immob.spdf,"prix.vente", carte= cont_midiP, identify=TRUE, cex.lab=0.6)

# ... with all options
histomap(immob.spdf,7, nbcol=15, type = "percent",
names.attr=names(immob), criteria=immob$rentabilite>5, carte=cont_midiP,
identify=TRUE, cex.lab=0.5, pch=12, col="pink",
xlab="variation price", ylab="percent", axes=TRUE, lablong="x",
lablat="y")
```

---

immob

*Some price indices of real estate from biggest cities in France*


---

## Description

This data frame contains price indices of real estate from biggest cities in France in 2008.

## Usage

```
data(immob)
```

## Format

A data frame with 94 observations on the following 10 variables.

Nom Name of the city

Code.INSEE Code Insee of the city

Code.region Code of the 'region' area of the city

longitude A vector x of size n

latitude A vector y of size n

prix.vente Average selling price at square metres

variation.vente Rates of change in the average selling price for a quarter year

prix.location Average price of rental square metres

variation.location Rates of change in the average price of rent for a quarter

rentabilite Rentability

## Source

Prepared by T. Laurent.

## References

<http://www.fnaim.fr/>

## Examples

```
data(immob)
```

---

makeneighborsw	<i>Spatial weight matrix</i>
----------------	------------------------------

---

### Description

The function `makeneighborsw()` create a spatial weight matrix based on a given number of nearest neighbors (option "neighbor" by default), based on a threshold distance (option `method="distance"`) or both these 2 methods.

### Usage

```
makeneighborsw(coords,method="neighbor",m=1,d,cum=TRUE)
```

### Arguments

<code>coords</code>	a matrix of spatial coordinates
<code>method</code>	"neighbor" by default, "distance" or "both"
<code>m</code>	number of nearest neighbors
<code>d</code>	threshold point
<code>cum</code>	if <code>cum=TRUE</code> , $W$ is the sum of spatial weight matrix based on $k$ nearest neighbours (for $k \leq m$ ; if <code>FALSE</code> $W$ is the spatial weight matrix based only on $m^{th}$ nearest neighbours

### Details

In the case of `method="neighbor"`, for each site, we order the other sites by their distance from this site. If `cum=TRUE`, for  $i$ , if  $j$  is among the  $m^{th}$  nearest sites, then :

$$W_{ij} = 1$$

else

$$W_{ij} = 0$$

If `cum=FALSE`, for

$$s_i$$

, if

$$s_j$$

is the  $m^{th}$  nearest site, then :

$$W_{ij} = 1$$

else

$$W_{ij} = 0$$

In case of ties, the nearest neighbour is randomly chosen.

In the case of method="distance", if site  $i$  is separated from  $j$  by a distance lower or equal to a given threshold :

$$W_{ij} = 1$$

else

$$W_{ij} = 0$$

In the case of method="both"  $W$  must verify the two first conditions.

### Value

A spatial weight matrix of size  $n \times n$

### Note

This function is not optimised for large dataset. User could find similar functions in the package `spdep` (`dnearneigh` and `knearneigh`). However, these functions don't offer the possibility to use the two criteria in the same time. Moreover, an inconvenient of `makeneighborsw` is that the result is included in a matrix object whereas most of functions of `GeoXp` use the `nb` structure for spatial weight matrix. An issue is to use the `mat2listw` function and then selecting the `nb` part, like the in the examples.

### Author(s)

Aragon Y., Thomas-Agnan C., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

### See Also

[moranplotmap,normw](#)

### Examples

```
##
# data auckland
data(auckland)
coords <- cbind(auckland$Easting[1:10],auckland$Northing[1:10])

# matrix based on 5 nearest neighbors
W<-makeneighborsw(coords,method="neighbor",m=3)

# matrix based on a threshold distance
```

```

W1<-makeneighborsw(coords,method="distance",d=20)

# matrix based on the two methods
W2<-makeneighborsw(coords,method="both",m=3,d=20)

# representation of the 3 spatial weight matrices
op<-par(mfrow=c(2,2))
plot(mat2listw(W),coords,col="lightblue1",main="neighbor")
plot(mat2listw(W1),coords,col="lightblue2",main="distance")
plot(mat2listw(W2),coords,col="lightblue3",main="both")
par(op)

```

---

 misolationmap

*Interactive multivariate isolation plot and map*


---

## Description

The function `misolationmap` draws a scatterplot with the pairwise Mahalanobis distances calculated using variables names `.var` between the observations and their neighbors on the y-axis and the "degree of isolation" of the observations on the x-axis and a map

## Usage

```

misolationmap(sp.obj, nb.obj, names.var, propneighb=0.4, chisqu=0.975,
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightb
xlab="degree of isolation", ylab="Pairwise Mahalanobis distances", axes=FALSE, lablong="", lablat="")

```

## Arguments

<code>sp.obj</code>	object of class extending Spatial-class
<code>nb.obj</code>	object of class nb
<code>names.var</code>	a vector of character; attribute names or column numbers in attribute table
<code>propneighb</code>	proportion of neighbors included in ellipsoid
<code>chisqu</code>	value of alpha for the definition of global outliers
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of size n of boolean which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	16 by default, symbol for selected points
<code>col</code>	color of the points on the cloud map

xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

### Details

The pairwise Mahalanobis distances are calculated using the robust Minimum Covariance Determinant (MCD) estimator associated with 75% of observations (function `covMcd` in the `robustbase` package) calculated on the variables `names.var`. For each observation, the degree of isolation is a chi-square quantile of the conditional distribution of the pairwise Mahalanobis distances associated with the ellipsoid containing the proportion `propneighb` of neighbors. The parameter `propneighb` gives the proportion of neighbors that is expected to be quite similar to the observation in order to conclude that the observation is not a local outlier. Under independence and normality conditions, the user can expect a degree of isolation close by the parameter `propneighb` (vertical line on the scatterplot). An observation with a high degree of isolation is suspected to be a local outlier. Users have also the possibility to plot bubbles on the map which size depends on the robust Mahalanobis distance of each observation to the center of the distribution (function `arw` in the package `mvoutlier`).

### Value

In the case where user click on `save results` button, a matrix of integer is created as a global variable in `last.select` object. It corresponds to the numbers of spatial unit corresponding to couple of sites selected just before leaving the Tk window.

### Author(s)

Fizmoser P., Thomas-Agnan C., Ruiz-Gazen A., Laurent T.,

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

### See Also

[mvariocloudmap](#)

### Examples

```
## data kola
data(moss)
data(kola.background)

# transformation of some variables
moss[,c("Ag", "As", "Bi", "Cd", "Co", "Cu", "Ni")]<-log10(moss[, c("Ag", "As", "Bi", "Cd", "Co", "Cu", "Ni")])
```

```
# creation of a SpatialClass object
kola.sp <- SpatialPoints(moss[,c("XC00", "YC00")])
kola.spdf <- SpatialPointsDataFrame(kola.sp, moss)

# creation of a spatial weight matrix nb
kola.nb <- dnearneigh(kola.sp, 0, 50000 )

# example of use of misolationmap
# The statistics are calculated by taking into account variables
# Ag, As, Bi, Cd, Co, Cu, Ni
misolationmap(kola.spdf, kola.nb, names.var=c("Ag", "As", "Bi", "Cd", "Co", "Cu", "Ni"),
propneighb=0.30, chisqu=0.95, carte= kola.background, identify=TRUE, cex.lab=0.5)
```

---

moranplotmap

*Moran scatterplot and map*


---

### Description

The function `moranplotmap()` draws a moran plot, used to detect spatial autocorrelation in the variable `var`. On the x-axis, is represented  $x - \bar{x}$  and on the y-axis  $W(x - \bar{x})$ , where  $W$  is the spatial weight matrix. It also calculates Moran's I statistic (see `nonnormoran`) and give a p-value associated to the autocorrelation test (gaussian version and permutation version).

### Usage

```
moranplotmap(sp.obj, name.var, listw.obj, flower=FALSE, locmoran=FALSE,
names.arg=c("H.-H.", "L.-H.", "L.-L.", "H.-L."), names.attr=names(sp.obj), criteria=NULL,
carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightblue3",
xlab="", ylab="", axes=FALSE, lablong="", lablat="")
```

### Arguments

<code>sp.obj</code>	object of class extending Spatial-class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>listw.obj</code>	object of class listw
<code>flower</code>	if TRUE, link neighbouring sites
<code>locmoran</code>	if TRUE, print local Moran's I statistic on the Moran plot
<code>names.arg</code>	names of the quadrant of the Moran plot
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .

cex.lab	character size of label
pch	16 by default, symbol for selected points
col	"lightblue3" by default, color of bars on the histogram
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

### Details

For the permutation test, for each drawing the values of the variable var are randomly assigned to the sites. We then calculate Moran'I statistic associated to each drawing and we give the frequency of drawings when Moran'I statistic is lower or equal to the observed Moran'I statistic. Moreover, the function gives the opportunity to link neighbouring sites and gives Local Moran's I statistic. For a site i:

$$I_i = (x_i - \bar{x}) \sum_j W_{ij} (x_j - \bar{x})$$

with j not equal to i.

### Value

In the case where user click on save results button, a list is created as a global variable in last.select object. \Sobs, a vector of integer, corresponds to the number of spatial units selected just before leaving the Tk window, \SMORAN, a numeric, corresponds to the value of the Moran'I statistic.

### Note

In the case of the spatial weight matrix is not normalized, the Moran'I statistic is not equal to  $\beta$  used in regression line for model  $W(X - \bar{X}) = \beta(X - \bar{X}) + u$ . That is why the regression line is only drawn in the case of W normalized.

### Author(s)

Aragon Y., Thomas-Agnan C., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Économétrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

Jim Lesage, "Spatial Econometrics Toolbox", <http://www.spatial-econometrics.com/>

**See Also**

[neighbourmap](#),[makeneighborsw](#),[normw](#),[nonormmoran](#)

**Examples**

```
##
# data baltimore
data(baltimore)

# SpatialPoints
baltimore.sp <- SpatialPoints(cbind(baltimore$X,baltimore$Y))
# SpatialPointsDataFrame
baltimore.spdf<-SpatialPointsDataFrame(baltimore.sp,baltimore)

# Spatial Weight Matrix
W.nb <- knn2nb(knearneigh(baltimore.sp, k=4))
# We choose a row standardized spatial weight matrix :
W.listw <- nb2listw(W.nb,style="W")

# moranplotmap with some options
moranplotmap(baltimore.spdf, "PRICE", W.listw ,
flower=TRUE, locmoran=TRUE,criteria=(baltimore.spdf$AC==1),
identify=TRUE)

# comparison with the moran.test function
moran.test(baltimore.spdf$PRICE,W.listw)

##
# data columbus
example(columbus)

# use of moranplotmap with spatial weight matrix col.gal.nb :
# 1. row-standardized
moranplotmap(columbus,"HOVAL",nb2listw(col.gal.nb,style="W"))

# 2. basic binary
moranplotmap(columbus,"HOVAL",nb2listw(col.gal.nb,style="B"))

# 3. globally standardized
moranplotmap(columbus,"HOVAL",nb2listw(col.gal.nb,style="C"))
```

**Description**

This data frame contains some information about schools in Midi-Pyrenees region. The school is the spatial unit level.

**Usage**

```
data(immob)
```

**Format**

A data frame with 226 observations on the following 8 variables.

`longitude` x-coordinate of the pseudo-canton

`latitude` y-coordinate of the pseudo-canton

`name.city` Name of the city

`index.rurality` A factor with levels of rurality

`Nb.students` Number of students

`Occupancy.rate` Rate of occupancy

`Cost.per.student` Average cast per student

`Nb.students.per.class` Average number of students per class

**Note**

The variables `Occupancy.rate`, `Cost.per.student` and `Nb.students.per.class` have been permuted because of the confidentiality of this data set.

**Source**

Prepared by T. Laurent.

**Examples**

```
data(mp.school)
```

---

mvariocloudmap

*Interactive multivariate variocloud and map*

---

**Description**

The function `mvariocloudmap()` draws a scatterplot of pairwise Mahalanobis distances and spatial distances with a map. It is a multivariate version of the `variocloud`. The number of couples of sites plotted can be reduced by considering couples above a quantile regression curve.

**Usage**

```
mvariocloudmap(sp.obj, nb.obj, names.var, quantiles=NULL,  
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightb  
xlab="Pairwise spatial distances", ylab="Pairwise Mahalanobis distances", axes=FALSE, lablong="", labl
```

**Arguments**

sp.obj	object of class extending Spatial-class
nb.obj	object of class nb
names.var	a vector of character; attribute names or column numbers in attribute table
quantiles	the value of alpha for representing alpha-quantile regression
names.attr	names to use in panel (if different from the names of variable used in sp.obj)
criteria	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the tcltk window
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	16 by default, symbol for selected points
col	color of the points on the cloud map
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

**Details**

The pairwise Mahalanobis distances are calculated using the Minimum Covariance Determinant (MCD) estimator associated with 75% of observations (function covMcd in the robustbase package). Users have the possibility to select some couples of sites on the scatterplot that are also highlighted on the map. Selection of observations on the map is also possible and leads to the selection of all the couples which contain the selected observations on the scatterplot.

**Value**

In the case where user click on save results button, a matrix of integer is created as a global variable in last.select object. It corresponds to the numbers of spatial unit corresponding to couple of sites selected just before leaving the Tk window.

**Author(s)**

Fizmoser P., Thomas-Agnan C., Ruiz-Gazen A., Laurent T.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

**See Also**[misolationmap](#)**Examples**

```
## data meuse
data(meuse)

# transformation of explanatory variables
meuse[,3:7]<-log(1+meuse[,3:7])

# creation of a Spatial Points object
meuse.sp<-SpatialPoints(cbind(meuse$x,meuse$y))

# creation of a SpatialPointsDataFrame
meuse.spdf<-SpatialPointsDataFrame(meuse.sp,meuse)

# for the spatial contours
data(meuse.riv)

# Spatial Weight matrix based on the 7th nearest neighbours
meuse.knn <- knearneigh(meuse.sp, k=7)
meuse.nb <- knn2nb(meuse.knn)

# example of use of mvariocloudmap. The statistic are calculated by taking
# into account variables cadmium,copper,lead,zinc,elev
mvariocloudmap(meuse.spdf,meuse.nb,c("cadmium","copper","lead","zinc","elev"),
quantiles=0.95, carte=meuse.riv[-c(1:20,73:98,156:176),],identify=TRUE,
criteria=(meuse.spdf$lime==1))
```

---

**neighbourmap***Neighbour plot and map*

---

**Description**

The function `neighbourmap()` identifies spatial outliers by comparing a variable value for a particular site with these of its neighbouring sites. It draws a scatterplot of the values of the variable at neighbouring sites for a neighbourhood structure given by a binary weight matrix  $W$  and links this scatterplot with a map.

**Usage**

```
neighbourmap(sp.obj, name.var, nb.obj, lin.reg=TRUE,
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightb
xlab="", ylab="", axes=FALSE, lablong="", lablat="")
```

**Arguments**

<code>sp.obj</code>	object of class extending Spatial-class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>nb.obj</code>	object of class nb
<code>lin.reg</code>	If TRUE, drawing the linear curve $y=x$
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of spatial units, which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	16 by default, symbol for selected points
<code>col</code>	"lightblue3" by default, color of bars on the histogram
<code>xlab</code>	a title for the graphic x-axis
<code>ylab</code>	a title for the graphic y-axis
<code>axes</code>	a boolean with TRUE for drawing axes on the map
<code>lablong</code>	name of the x-axis that will be printed on the map
<code>lablat</code>	name of the y-axis that will be printed on the map

**Details**

For a selected site  $j$  on the map, are represented on the map its neighbors, and on the graph, on the x-axis the value of `var` for this site, and in y-axis, the values of `var` for the neighbouring sites of  $j$ . For a selected point on the graph, the corresponding pair of sites is represented on the map with a link.

**Value**

In the case where user click on `save results` button, a matrix of integer is created as a global variable in `last.select` object. It corresponds to the numbers of spatial unit corresponding to couple of sites selected just before leaving the Tk window.

**Note**

When user selects sites on the graph or on the map, he cannot add a selection by using the other graphic.

**Author(s)**

Aragon Y., Thomas-Agnan C., Ruiz-Gazen A., Laurent T., Robidou L.

## References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

## See Also

[moranplotmap](#), [makeneighborsw](#), [normmw](#), [nonormmoran](#)

## Examples

```
###
# columbus
example(columbus)

# example of use of neighbourmap
neighbourmap(columbus, "CRIME", col.gal.nb,
criteria=(columbus@data$CRIME>mean(columbus@data$CRIME)))

###
# data immob
data(immob)

# change names of individuals
row.names(immob) <- immob$Nom

# immob is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
immob.sp = SpatialPoints(cbind(immob$longitude,immob$latitude))
# ... and then by integrating other variables to create SpatialPointsDataFrame
immob.spdf = SpatialPointsDataFrame(immob.sp, immob)
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
midiP <- readShapePoly(system.file("shapes/region.shp", package="GeoXp")[1])
cont_midiP<-spdf2list(midiP[-c(22,23),])$poly

# A spatial weight matrix based on triangulation Delaunay
W.nb<-tri2nb(cbind(immob$longitude,immob$latitude))

# example of use of neighbourmap
neighbourmap(immob.spdf,"prix.vente", W.nb, identify=TRUE, cex.lab=0.5,
carte=cont_midiP)
```

---

 nonormmoran

*Detection of spatial autocorrelation*


---

**Description**

The function nonormmoran is used to detect spatial autocorrelation in the residuals  $u$  from the least squares model  $Y = \beta \times X + u$ . It calculates Moran's I statistic of the residuals based on the gaussian asymptotic distribution and give a p-value associated to the test of spatial autocorrelation (gaussian version).

**Usage**

```
nonormmoran(y, x, W)
```

**Arguments**

**y** vector of size n of dependent variable  
**x** matrix  $n \times p$  containing explanatory variables  
**W** spatial weight matrix

**Details**

W is supposed standartized :

$$I = (n/s) \frac{u'Wu}{u'u}$$

$$I \sim N(E(I), var(I))$$

let  $M = (I - X(X'X)^{-1}X')$

$$E(I) = (n/s) \frac{tr(MW)}{n - k}$$

$$d = \frac{n - p}{n + p + 2}$$

$$V(I) = (n/s)^2 [tr(MWMW') + tr(MW)^2 + (tr(MW))^2] / d - E(I)^2$$

$$Z_I = \frac{I - E(I)}{v(I)^{1/2}}$$

When W is row-normalised,  $s = n$  else  $s$  is the number of nonzero links

**Value**

a list with nobs is the number of observations, nvar, the number of explanatory variables, morani is the Moran's I statistic estimate, imean is E(I), ivar is var(I), istat is the normalized Moran's I statistic (corresponding to  $Z_I$ ), and prob the associated p-value.

**Note**

This function is closed to the function `lm.morantest` included in the `spdep` package. `nonnorm-moran` is less comfortable to use because it didn't take into account S4 classes.

**Author(s)**

Translated into R from Jim Lesage's *Spatial Econometrics Toolbox*, <http://www.spatial-econometrics.com/>

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Économétrie pour données géoréférencées : modèles et études de cas*

**See Also**

[moranplotmap](#), [neighbourmap](#), [makeneighborsw](#), [normw](#)

**Examples**

```
###
# data baltimore
data(baltimore)

# a spatial weight matrix constructed in the matrix format
W <- makeneighborsw(cbind(baltimore$X,baltimore$Y),method="neighbor",4)

# when W is not row-normalised ...
nonnormmoran(baltimore$PRICE,cbind(rep(1,nrow(baltimore)),baltimore[,14:15]),W)
# when W is row_normalised ...
nonnormmoran(baltimore$PRICE,cbind(rep(1,nrow(baltimore)),baltimore[,14:15]),normw(W))

# If we compare to the function lm.morantest
baltimore.lm<-lm(PRICE~LOTSZ+SQFT,data=baltimore)

lm.morantest(baltimore.lm, mat2listw(W))
```

---

normw

*Row-normalize a spatial weight matrix*

---

**Description**

The function `normw()` row-normalizes a spatial weight matrix

**Usage**

```
normw(w)
```

**Arguments**

w                    A matrix of size  $n \times n$

**Details**

$$W_{ij} = \frac{W_{ij}}{\sum_k W_{ik}}$$

**Value**

A matrix of size  $n \times n$

**Author(s)**

Aragon Y., Thomas-Agnan C., Ruiz-Gazen A., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

**See Also**

[neighbourmap](#), [moranplotmap](#), [makeneighborsw](#)

**Examples**

```
###
# data auckland
data(auckland)
x.ext <- auckland$Easting[1:10]
y.ext <- auckland$Northing[1:10]

# matrix based on 3 nearest neighbors
W1<-makeneighborsw(cbind(x.ext,y.ext),method="both",m=3,d=20)
W2<-normw(W1)

apply(W1[1:10,],1,sum)
apply(W2[1:10,],1,sum)
```

**Description**

The function `pcamap()` draws the plots summarizing a generalized Principal Component Analysis (PCA), made with `genpca`. It draws the scatterplot of the individuals projected on a chosen principal component plane (with their percentage of inertia), together with the scatterplot of the variables projected into the same plane with the quality of representation in order to interpret the principal component axes. The individuals scatterplot interacts with the map.

**Usage**

```
pcamap(sp.obj, names.var, direct=c(1,2), weight=rep(1/nrow(sp.obj), length=nrow(sp.obj)),
metric=diag(length(names.var)), center=NULL, reduce=TRUE, qualproj=FALSE,
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8, pch=16, col="lightblue",
xlab=paste(direct[1]), ylab=paste(direct[2]), axes=FALSE, lablong="", lablat="")
```

**Arguments**

<code>sp.obj</code>	object of class extending Spatial-class
<code>names.var</code>	a vector of character; attribute names or column numbers in attribute table
<code>direct</code>	Two-dimension vector containing the numbers of principal axes to plot
<code>weight</code>	vector of size n of weight (by default : $\text{weight} = t(1/n, \dots, 1/n)$ )
<code>metric</code>	matrix $p \times p$ (by default : <code>metric=Identity matrix</code> )
<code>center</code>	A vector of size p of the gravity center which by default is equal to $\bar{X}$
<code>reduce</code>	if TRUE, reduced PCA
<code>qualproj</code>	if TRUE, print the quality of representation of individuals
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of Spatial unit, which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	a vector of symbol which must be equal to the number of group else all sites are printed in <code>pch[1]</code>
<code>col</code>	a vector of colors which must be equal to the number of group else all sites and all bars are printed in <code>col[1]</code>
<code>xlab</code>	a title for the graphic x-axis
<code>ylab</code>	a title for the graphic y-axis

axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

### Details

Let

$$D = \text{diag}(\lambda_1, \dots, \lambda_p)$$

$$1_p = (1, \dots, 1)'$$

Let the coordinates of individuals in the principals components

$$CC = (C'_1, \dots, C'_n)'$$

with  $C_i = (C_i^1, \dots, C_i^p)$

Let the coordinates of variables in the principals components

$$CC = (V'_1, \dots, V'_p)'$$

with  $V_i = (V_i^1, \dots, V_i^p)$

Part of inertia :

$$\left( \frac{\lambda_1}{\sum_i \lambda_i}, \dots, \frac{\lambda_p}{\sum_i \lambda_i} \right)'$$

Quality of representation of individual k projected on plane (i,j):

$$Qu = \sqrt{\frac{(C_k^i)^2 + (C_k^j)^2}{\sum_l (C_k^l)^2}}$$

Quality of representation of variable k projected on plane (i,j):

$$VQu = \sqrt{\frac{(V_k^i)^2 + (V_k^j)^2}{\sum_l (V_k^l)^2}}$$

### Value

In the case where user click on save results button, a list is created as a global variable in last.select object. obs, corresponds to the number of spatial units selected just before leaving the Tk window, inertia vector of size p with percent of inertia of each component, casecoord matrix  $n \times p$  of individuals, varcoord matrix  $n \times p$  of principal components.

### Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Caussinus H., Fekri M., Hakam S., Ruiz-Gazen A. (2003) , *A monitoring display of Multivariate Outliers*, Computational Statistics and Data Analysis, vol. 44, 1-2, 237-252.

**See Also**

[clustermap, genpca](#)

**Examples**

```
###
# Data Columbus
example(columbus)

# a basic PCA on 7 variables
pcamap(columbus,c(6:12), qualproj=TRUE, identify=TRUE)

###
# data boston
data(boston)

# SpatialPoints object
boston.sp<-SpatialPoints(cbind(boston.utm[,1],boston.utm[,2]))

# SpatialPointsDataFrame object
boston.spdf <- SpatialPointsDataFrame(boston.sp, boston.c)

# a basic PCA on 7 variables
pcamap(boston.spdf, c(7:8,10,12:15), identify=TRUE, cex.lab=0.5)

# generalized PCA : user have to construct a new metric and a vector
# of gravity center, by using for exampe covMcd
cov.boston<-covMcd(boston.c[,c(7:8,10,12:15)],alpha=.75)
b.center<-cov.boston$center
b.cov<-cov.boston$cov

# example of use of pcamap
pcamap(boston.spdf, c(7:8,10,12:15), metric=b.cov, center=b.center,identify=TRUE,
cex.lab=0.5)
```

---

plot3dmap

*Interactive Plot3d and map*

---

**Description**

The function `plot3dmap()` draws a 3d-plot of three given variables `$names.var$` and a map with sites of coordinates `coordinates(sp.obj)`.

**Usage**

```
plot3dmap(sp.obj, names.var, box=TRUE,
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8,
pch=16, col="lightblue3",xlab="", ylab="", zlab="", axes=FALSE, lablong="", lablat="")
```

**Arguments**

sp.obj	object of class extending Spatial-class
names.var	a vector of three characters; attribute names or column numbers in attribute table
box	a boolean with TRUE for drawing a box on the scatterplot 3d
names.attr	names to use in panel (if different from the names of variable used in sp.obj)
criteria	a vector of boolean of size the number of Spatial units, which permit to represent preselected sites with a cross, using the tcltk window
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	16 by default, symbol for selected points
col	"lightblue3" by default, color of bars on the histogram
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
zlab	a title for the graphic z-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

**Details**

Sites selected on the map by 'points' or 'polygon' are represented in red in the 3-d plot.

**Value**

In the case where user click on save results button, a vector of integer is created as a global variable in last.select object. It corresponds to the number of spatial units selected just before leaving the Tk window.

**Note**

This function uses the rgl package and open a rgl device.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

**See Also**[scattermap](#)**Examples**

```

# data on price indices of real estate in France
#####
# data on price indices of real estate in France
data(immob)
row.names(immob)<-immob$Nom

# immob is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
immob.sp = SpatialPoints(cbind(immob$longitude,immob$latitude))
# ... and then by integrating other variables to create SpatialPointsDataFrame
immob.spdf = SpatialPointsDataFrame(immob.sp, immob)
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
midiP <- readShapePoly(system.file("shapes/region.shp", package="GeoXp")[1])
cont_midiP<-spdf2list(midiP[-c(22,23),])$poly

# an example of plot3dmap
plot3dmap(immob.spdf, c("prix.vente", "prix.location", "variation.vente"),
box=FALSE, carte=cont_midiP, identify=TRUE, cex.lab=0.5, xlab="prix.vente",
ylab="prix.location", zlab="variation.vente")

#####
# data eire
eire <- readShapePoly(system.file("etc/shapes/eire.shp", package="spdep")[1],
ID="names", proj4string=CRS("+proj=utm +zone=30 +units=km"))

# an example of use
plot3dmap(eire, c("A", "RETSALE", "INCOME"), xlab="A", ylab="RETSALE", zlab="INCOME")

```

---

polyboxplotmap

*Interactive polyboxplot and map*

---

**Description**

Parallel Boxplots of a numerical variable by levels of a factor. It interacts with a map.

**Usage**

```
polyboxplotmap(sp.obj, names.var, varwidth=FALSE, names.arg = "",
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8,
pch=16, col="lightblue3",xlab="", ylab="count", axes=FALSE, lablong="", lablat="")
```

**Arguments**

sp.obj	object of class extending Spatial-class
names.var	a vector of 2 characters; first character corresponds to the name of a factor and second character corresponds to a numeric value
varwidth	if varwidth is TRUE, the boxes are drawn with widths proportional to the square-roots of the number of observations in the groups
names.arg	a vector of level names (for factor)
names.attr	names to use in panel (if different from the names of variable used in sp.obj)
criteria	a vector of boolean of size the number of Spatial units, which permit to represent preselected sites with a cross, using the tcltk window
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	a vector of symbol which must be equal to the number of group else all sites are printed in pch[1]
col	a vector of colors which must be equal to the number of group else all sites and all box are printed in col[1]
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

**Details**

Each site is associated to a value of a level of \$names.var[1]\$ and of a value of \$names.var[2]\$. There is an interactivity only in one direction : the sites selected by quartile-intervals on one of the boxplots are then represented on the map in red (or colors according to the options).

**Value**

In the case where user click on save results button, a vector of integer is created as a global variable in last.select object. It corresponds to the number of spatial units selected just before leaving the Tk window.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

**See Also**

[dblehistomap](#), [histobarmap](#), [scattermap](#), [dbledensitymap](#)

**Examples**

```
#####  
# data eire  
eire <- readShapePoly(system.file("etc/shapes/eire.shp", package="spdep")[1],  
ID="names", proj4string=CRS("+proj=utm +zone=30 +units=km"))  
  
# example of use of polyboxplotmap  
polyboxplotmap(eire,c("pale","POPCHG"), names.arg=c("Outside Pale","Pale"),  
xlab="Appartenance to the region of Pale", col=c("pink","violet"))
```

---

polylist2list

*Extract from a polylist object the vertices of the polygons*

---

**Description**

The function `polylist2list()` extracts the vertices of the polygons from a `polylist` object, separating polygons from each other by 3 rows of `NaN`.

**Usage**

```
polylist2list(data)
```

**Arguments**

`data`            A `polylist` object

**Details**

The user can then represent the coordinates of sites of a `polylist` object as background map, using the option `carte` included in all interactive functions of `GeoXp`.

**Value**

It returns a matrix of numeric values with 2 columns (x and y coordinates of the vertices of the polygons) where polygons are separated from each other by 3 rows of `NaN`.

**Author(s)**

T. Laurent

**References**Bivand R. et al. (2009), *Applied Spatial Data Analysis with R*, Springer.**See Also**[spdf2list](#)**Examples**

```
data(eire)
eire.contours<-polylist2list(eire.polys.utm)
```

---

rotation

*Transform coordinates of sites using a rotation*


---

**Description**

The function `rotation()` is used to modify coordinates of sites by a rotation with an angle equal to `angle`. This function is ude in `derifmap`

**Usage**

```
rotation(coords, angle)
```

**Arguments**

<code>coords</code>	matrix $n \times 2$ of coordinates
<code>angle</code>	value of angle to use in rotation in degree

**Details**

Let:

$$x = (\cos(\theta), -\sin(\theta))$$

$$y = (\sin(\theta), \cos(\theta))$$

$$nlecoord = coords \times cbind(x, y)$$

**Value**matrix  $n \times 2$  of new coordinates.**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

## References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

## See Also

[angleplotmap](#), [variocloudmap](#)

---

scattermap	<i>Interactive scatterplot and map</i>
------------	--

---

## Description

The function scattermap draws a scatterplot of the given variables indicated in names.var and a map with sites of coordinates coordinates(sp.obj). Boxplots of each variable names.var[1] and names.var[2] are represented below the x-axis and y-axis.

## Usage

```
scattermap(sp.obj, names.var, lin.reg=TRUE, quantiles=NULL,
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8,
pch=16, col="lightblue3", xlab="", ylab="", axes=FALSE, lablong="", lablat="")
```

## Arguments

sp.obj	object of class extending Spatial-class
names.var	a vector of two characters : 1st name corresponds to the x-variable, 2nd name corresponds to the y-variable
lin.reg	If TRUE, drawing of the linear predictor for 'response' in linear model
quantiles	a vector of numeric corresponding to order of conditionnal quantile regression spline
names.attr	names to use in panel (if different from the names of variable used in sp.obj)
criteria	a vector of boolean of size the number of Spatial Units, which permit to represent preselected sites with a cross, using the tcltk window
carte	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
identify	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table row.names(as.data.frame(sp.obj)).
cex.lab	character size of label
pch	16 by default, symbol for selected points
col	"lightblue3" by default, color of bars on the histogram
xlab	a title for the graphic x-axis
ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map

## Details

Each site is associated to a value of `names.var[1]` and a value of `names.var[2]`. There is an interactivity between the two windows: the sites selected by ‘point’ or ‘polygon’ on the scatterplot are represented on the map in red; sites selected on the map are then represented in red on the scatterplot. Users have the possibility to draw linear predictor for ‘response’ in linear model (option `lin.reg`) or conditionnal quantile regression spline (option `quantiles`).

## Value

In the case where user click on `save results` button, a vector of integer is created as a global variable in `last.select` object. It corresponds to the number of spatial units selected just before leaving the Tk window.

## Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

## References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

## See Also

[dblehistomap](#), [histobarmap](#), [scattermap](#), [densitymap](#)

## Examples

```
#####
# data on price indices of real estate in France
data(immob)
row.names(immob)<-immob$Nom

# immob is a data.frame object. We have to create
# a Spatial object, by using first the longitude and latitude
# to create Spatial Points object ...
immob.sp = SpatialPoints(cbind(immob$longitude,immob$latitude))
# ... and then by integrating other variables to create SpatialPointsDataFrame
immob.spdf = SpatialPointsDataFrame(immob.sp, immob)
# For more details, see vignette('sp', package="sp")

# optional : we add some contours that don't correspond to the spatial unit
# but are nice for mapping
midiP <- readShapePoly(system.file("shapes/region.shp", package="GeoXp")[1])
cont_midiP<-spdf2list(midiP[-c(22,23),])$poly
```

```
# a example of use
scattermap(immob.spdf,c("prix.vente","prix.location"),quantiles=0.5,
carte= cont_midiP, xlab="Average sell price",ylab="Average rent price",
identify=TRUE, cex.lab=0.6)

#####
# data eire
eire <- readShapePoly(system.file("etc/shapes/eire.shp", package="spdep")[1],
ID="names", proj4string=CRS("+proj=utm +zone=30 +units=km"))

# example of use of scattermap
scattermap(eire, c("ROADACC","OWNCONS"),lin.reg=TRUE,quantiles=0.95,
xlab="Roads",ylab="Consomation Rate",col="purple")
```

---

selectgraph

*Selection of an additionnal graph*

---

### Description

The function `selectgraph()` is used in most of the GeoXp functions to choose an additional graph and the variables associated to this graph.

### Usage

```
selectgraph(listnomvar, listgraph)
```

### Arguments

<code>listnomvar</code>	list of names of variables given
<code>listgraph</code>	kind of graphics

### Details

This function is not an interactive function.

### Value

Names of variables and graph chosen.

### Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

---

selectmap	<i>Selection of a point or polygon on a scatterplot</i>
-----------	---

---

**Description**

The function `selectmap()` is used to select a point or a polygon on the map or on a scatterplot. Called by any function which draws a scatterplot.

**Usage**

```
selectmap(var1, var2, obs, Xpoly, Ypoly, method = "")
```

**Arguments**

var1	a vector x of size n
var2	a vector y of size n
obs	a boolean vector of size n : TRUE if a site is already selectioned, FALSE otherwise.
Xpoly	X-coordinates of the vertices of selected polygon
Ypoly	Y-coordinates of the vertices of selected polygon
method	'point' if the selected area is a point; 'poly' if the selected area is a polygon

**Details**

This function is called by all the functions which draw a scatterplot such as `scattermap`, `moranplotmap`,... This is not an interactive function.

**Value**

A vector of boolean of size n. TRUE if a site has been selected, FALSE otherwise.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

---

selectstat	<i>Selection of values on a graphic</i>
------------	---

---

### Description

The function `selectstat()` is used to select sites on a graphic such as Histogram, Barplot, Boxplot, Polyboxplot, Scatterplot, Densityplot, Neighbourplot, Lorentz, Moran, Angleplot, Variocloud, Cluster, or PCA. Most of the GeoXp functions call `selectstat.r`.

### Usage

```
selectstat(var1, var2, obs, Xpoly, Ypoly, method,
           nbc1, W, F, long, lat)
```

### Arguments

var1	1st variable of size n
var2	2nd variable of size n
obs	a boolean vector of size n : TRUE if a site is already selected, FALSE otherwise.
Xpoly	X-coordinates of the vertices of selected polygon
Ypoly	Y-coordinates of the vertices of selected polygon
method	name of the graph wich must be drawn among Histogram, Barplot, Boxplot, Polyboxplot, Densityplot, Neighbourplot, Lorentz, Anglepoint, Variopoint
nbc1	nbc1 : number of cells for the histogram (10 by default)
W	A matrix of size $n \times n$ (for spatial weight)
F	a vector of numeric
long	a vector of x-axis of size n
lat	a vector of y-axis of size n

### Details

This function is not an interactive function.

### Value

Return a vector of boolean of size n with TRUE if sites have been selected on a graph

### Author(s)

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

---

slider1	<i>Scroll bar</i>
---------	-------------------

---

**Description**

Creates a scroll bar for modifying the value of a parameter.

**Usage**

```
slider1(fenetre, refresh.code, names.slide, minima, maxima, resolutions,  
starts, no=0)
```

**Arguments**

fenetre	number of windows
refresh.code	name of function called after modifying the value of a parameter
names.slide	title for scroll bar
minima	minimum value of parameter
maxima	maximum value of parameter
resolutions	scale
starts	Initial Value
no	number of scroll bar

**Details**

This function is used in the functions which draw a curve to modify the smoothing parameter. This is not an interactive function

**Value**

Draws a Tk window.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Économétrie pour données géoréférencées : modèles et études de cas*

**See Also**

[densitymap](#), [dbledensitymap](#)

---

spdf2list	<i>Extract from a SpatialPolygonsDataFrame object the middle coordinates of boundary box and the vertices of the polygons (in the case where polygons are given)</i>
-----------	--

---

### Description

The function `spdf2list()` calculates the middle coordinates of boundary box and in the case where polygons are included into a `SpatialPolygonsDataFrame` object, it extracts the vertices of the polygons, separating polygons from each other by 3 rows of `NaN`.

### Usage

```
spdf2list(data)
```

### Arguments

`data`            A `SpatialPolygonsDataFrame` object

### Details

The user can then represent the coordinates of sites of a `SpatialPolygonsDataFrame` object as background map, using the option ‘`carte`’ included in all interactive functions.

### Value

It returns two vectors of middle coordinates for x-axis and y-axis corresponding to middle of each boundary box. It returns a matrix of numeric values with 2 columns (x and y coordinates of the vertices of the polygons) where polygons are separated from each other by 3 rows of `NaN`.

### Note

The data of a `SpatialPolygonsDataFrame` object can be directly extract using `@data`

### Author(s)

Thibault Laurent

### References

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Roger S.Bivand, Edzer J.Pebesma, Virgilio Gomez-Rubio (2009), *Applied Spatial Data Analysis with R*, Springer.

**See Also**[polylist2list](#)**Examples**

```
africa <- readShapePoly(system.file("shapes/Africa.shp", package = "GeoXp")[1])
africa.contour<-spdf2list(africa)$poly
```

---

 variocloudmap

*Interactive variocloud and map*


---

**Description**

The function `variocloudmap()` draws a semi-variocloud (directional or omnidirectional) and a map. It is used to detect spatial autocorrelation. Possibility to draw the empirical semi-variogram and a robust empirical semi-variogram.

**Usage**

```
variocloudmap(sp.obj, name.var, bin=NULL, quantiles=NULL,
names.attr=names(sp.obj), criteria=NULL, carte=NULL, identify=FALSE, cex.lab=0.8,
pch=16, col="lightblue3", xlab="", ylab="", axes=FALSE, lablong="", lablat="",
xlim=NULL, ylim=NULL)
```

**Arguments**

<code>sp.obj</code>	object of class extending Spatial-class
<code>name.var</code>	a character; attribute name or column number in attribute table
<code>bin</code>	a vector of numeric values where empirical variogram will be evaluated
<code>quantiles</code>	a vector of numeric values of quantile orders (the regression quantile is obtained by spline smoothing)
<code>names.attr</code>	names to use in panel (if different from the names of variable used in <code>sp.obj</code> )
<code>criteria</code>	a vector of boolean of size the number of Spatial Units, which permit to represent preselected sites with a cross, using the tcltk window
<code>carte</code>	matrix with 2 columns for drawing spatial polygonal contours : x and y coordinates of the vertices of the polygon
<code>identify</code>	if not FALSE, identify plotted objects (currently only working for points plots). Labels for identification are the row.names of the attribute table <code>row.names(as.data.frame(sp.obj))</code> .
<code>cex.lab</code>	character size of label
<code>pch</code>	16 by default, symbol for selected points
<code>col</code>	"lightblue3" by default, color of bars on the cloud map
<code>xlab</code>	a title for the graphic x-axis

ylab	a title for the graphic y-axis
axes	a boolean with TRUE for drawing axes on the map
lablong	name of the x-axis that will be printed on the map
lablat	name of the y-axis that will be printed on the map
xlim	the x limits of the plot
ylim	the y limits of the plot

**Details**

For some couple of sites  $(s_i, s_j)$ , the graph represents on the y-axis the semi squared difference between  $var_i$  and  $var_j$  :

$$\gamma_{ij} = \frac{1}{2}(var_i - var_j)^2$$

and on the x-axis the distance  $h_{ij}$  between  $s_i$  and  $s_j$ . The semi Empirical variogram has been calculated as :

$$\gamma(h) = \frac{1}{2|N(h)|} \sum_{N(h)} (Z(s_i) - Z(s_j))^2$$

where

$$N(h) = \{(s_i, s_j) : s_i - s_j = h; i, j = 1, \dots, n\}$$

and the robust version :

$$\gamma(h) = \frac{1}{2(0.457 + \frac{0.494}{|N(h)|})} \left( \frac{1}{|N(h)|} \sum_{N(h)} |Z(s_i) - Z(s_j)|^{1/2} \right)^4$$

The number N of points to evaluate the empirical variogram and the distance  $\epsilon$  between points are set as follows :

$$N = \frac{1}{\max(30/n^2, 0.08, d/D)}$$

and :

$$\epsilon = \frac{D}{N}$$

with :

$$D = \max(h_{ij}) - \min(h_{ij})$$

and :

$$d = \max(h_{ij}^{(l)} - h_{ij}^{(l+1)}),$$

where  $h^{(l)}$  is the vector of sorted distances. In options, possibility to represent a regression quantile smoothing spline  $g_\alpha$  (in that case the points below this quantile curve are not drawn).

**Value**

In the case where user click on save results button, a matrix of integer is created as a global variable in last.select object. It corresponds to the numbers of spatial unit corresponding to couple of sites selected just before leaving the Tk window.

**Author(s)**

Thomas-Agnan C., Aragon Y., Ruiz-Gazen A., Laurent T., Robidou L.

**References**

Aragon Yves, Perrin Olivier, Ruiz-Gazen Anne, Thomas-Agnan Christine (2012), *Statistique et Econométrie pour données géoréférencées : modèles et études de cas*

Cressie N. and Hawkins D. (1980), *Robust estimation of the variogram*, in Journal of the international association for mathematical geology, 13, 115-125.

**See Also**

[angleplotmap](#), [driftmap](#)

**Examples**

```
#####  
# Data Meuse  
data(meuse)  
  
# meuse is a data.frame object. We have to create  
# a Spatial object, by using first the longitude and latitude  
# to create Spatial Points object ...  
meuse.sp = SpatialPoints(cbind(meuse$x,meuse$y))  
# ... and then by integrating other variables to create SpatialPointsDataFrame  
meuse.spdf = SpatialPointsDataFrame(meuse.sp, meuse)  
  
# meuse.riv is used for contour plot  
data(meuse.riv)  
  
# example of use of variocloudmap  
variocloudmap(meuse.spdf, "zinc", quantiles=0.75, bin=seq(0,2000,100),  
xlim=c(0,2000),ylim=c(0,500000),pch=2,carte=meuse.riv[c(21:65,110:153),],  
criteria=(meuse$lime==1))
```

# Index

## \*Topic **datasets**

immob, 41  
mp.school, 48

## \*Topic **manip**

polylist2list, 63  
spdf2list, 71

## \*Topic **multivariate**

clustermap, 16  
dblehistomap, 21  
densitymap, 23  
genpca, 27  
histobarmap, 37  
misolationmap, 44  
mvariocloudmap, 49  
pcamap, 57  
plot3dmap, 59  
polyboxplotmap, 61  
scattermap, 65

## \*Topic **regression**

scattermap, 65

## \*Topic **smooth**

dbledensitymap, 19  
densitymap, 23

## \*Topic **spatial**

angleplotmap, 4  
barmap, 7  
barnbmap, 9  
boxplotmap, 11  
carte, 12  
choix.bubble, 14  
choix.couleur, 15  
choixvarfunc, 16  
clustermap, 16  
dbledensitymap, 19  
dblehistomap, 21  
densitymap, 23  
driftmap, 25  
genpca, 27  
GeoXp-package, 3

gini, 30

ginimap, 31

graphique, 32

histnbmap, 34

histobarmap, 37

histomap, 39

makeneighborsw, 42

misolationmap, 44

moranplotmap, 46

mvariocloudmap, 49

neighbourmap, 51

nonormmoran, 54

normw, 55

pcamap, 57

plot3dmap, 59

polyboxplotmap, 61

rotation, 64

scattermap, 65

selectgraph, 67

selectmap, 68

selectstat, 69

slider1, 70

variocloudmap, 72

## \*Topic **univar**

barmap, 7

boxplotmap, 11

densitymap, 23

ginimap, 31

histomap, 39

## \*Topic **utilities**

carte, 12

choix.bubble, 14

choix.couleur, 15

choixvarfunc, 16

graphique, 32

rotation, 64

selectgraph, 67

selectmap, 68

selectstat, 69

- angleplotmap, 4, 27, 65, 74
- barmap, 7, 18
- barnbmap, 9
- boxplotmap, 11
  
- carte, 12
- choix.bubble, 14
- choix.couleur, 15
- choix.couleur (choix.couleur), 15
- choixvarfunc, 16
- clustermap, 16, 29, 59
  
- dbledensitymap, 19, 38, 63, 70
- dblehistomap, 21, 22, 38, 63, 66
- densitymap, 8, 12, 20, 22, 23, 25, 40, 66, 70
- driftmap, 6, 25, 74
  
- genpca, 27, 59
- GeoXp (GeoXp-package), 3
- GeoXp-package, 3
- gini, 30, 32
- ginimap, 31, 31
- graphique, 32
  
- histnbmap, 34
- histobarmap, 8, 12, 20, 22, 25, 37, 38, 40, 63, 66
- histomap, 8, 12, 20, 25, 39, 40
  
- immob, 41
  
- makeneighborsw, 10, 36, 42, 48, 53, 55, 56
- misolationmap, 44, 51
- moranplotmap, 10, 36, 43, 46, 53, 55, 56
- mp.school, 48
- mvariocloudmap, 45, 49
  
- neighbourmap, 48, 51, 55, 56
- nonormoran, 10, 36, 48, 53, 54
- normw, 10, 36, 43, 48, 53, 55, 55
  
- pcamap, 18, 29, 57
- plot3dmap, 59
- polyboxplotmap, 61
- polylist2list, 63, 72
  
- rotation, 27, 64
  
- scattermap, 8, 12, 20, 22, 25, 38, 40, 61, 63, 65, 66
  
- selectgraph, 67
- selectmap, 68
- selectstat, 69
- slider1, 70
- spdf2list, 64, 71
  
- variocloudmap, 6, 27, 65, 72