Spatial data in R: using R as a GIS (old version)

posted 9 Dec 2011 16:05 by Paco Rodriguez **[updated 18 Dec 2013 16:51]**

NOTE: This is an old version. The tutorial has been updated. You can find the new version HERE.

In a previous post I pointed out several free alternatives for Geographical Information Systems (GIS). Besides purely GIS software, such as SAGA, GRASS, gvSIG, DIVA-GIS, or QGIS, the open-source statistical software R has increasingly gained GIS capabilities, and is now able to perform most (if not all) the operations we typically do with traditional GIS software.
In our group meeting this week I made a short tutorial on how to perform basic GIS operations in R, such as importing and exporting data (both vectorial and raster), plotting, analysing and making maps. I paste the code used below, in the hope that it will be useful to GIS and R users currrently learning how to deal with spatial data in R. Please note that the code is very introductory, far from comprehensive, and there might be some errors or better ways of performing a task. But I think most basic GIS operations are described here, so the code may be used as a reference for occassional users. I´ll try to keep this updated as new functionalities appear (see the links).

Enjoy your mapping!

```
##########################################################

### Using R as a GIS ###

##########################################################

# Basic GIS operations in R
# v 1.1
# 09/12/2011
# Francisco Rodriguez-Sanchez
# Look for the latest version at
# http://sites.google.com/site/rodriguezsanchezf
```

```
# Note this introductory code is far from comprehensive
# and focussed on ecological-biogeographical analyses


setwd("~/UsingR-GIS")


### Basic packages ###

library(sp)              # classes for spatial data
library(raster)          # grids, rasters
library(rasterVis)       # raster visualisation
library(maptools)
# and their dependencies


#########################################################

### VISUALISATION OF GEOGRAPHICAL DATA ###


### RWORLDMAP ###

library(rworldmap)    # visualising (global) spatial dat

  # examples:
  newmap <- getMap(resolution="medium", projection="non
  plot(newmap)

  mapCountryData()
  mapCountryData(mapRegion="europe")
  mapGriddedData()
  mapGriddedData(mapRegion="europe")


### GOOGLEVIS ###

library(googleVis)     # visualise data in a web browser
Visualisation API

  # demo(googleVis)    # run this demo to see all the pc

  # Example: plot country-level data
  data(Exports)
  View(Exports)
  Geo <- gvisGeoMap(Exports, locationvar="Country", num
                    options=list(height=400, dataMode='
  plot(Geo)
  print(Geo)
  # this HTML code can be embedded in a web page (and b

  # Example: Plotting point data onto a google map (int
  data(Andrew)
  M1 <- gvisMap(Andrew, "LatLong" , "Tip", options=list
showLine=F, enableScrollWheel=TRUE,
                              mapType='satellite', useMapT
width=800,height=400))
  plot(M1)
```

```
### RGOOGLEMAPS ###

library(RgoogleMaps)

  # get maps from Google
  newmap <- GetMap(center=c(36.7,-5.9), zoom =10, destf
maptype = "satellite")
  # View file in your wd
  # now using bounding box instead of center coordinate
  newmap2 <- GetMap.bbox(lonR=c(-5, -6), latR=c(36, 37)
"newmap2.png", maptype="terrain")     # try different ma
  newmap3 <- GetMap.bbox(lonR=c(-5, -6), latR=c(36, 37)
"newmap3.png", maptype="satellite")

  # and plot data onto these maps, e.g. these 3 points
   PlotOnStaticMap(lat = c(36.3, 35.8, 36.4), lon = c(-
10, cex=2, pch= 19, col="red", FUN = points, add=F)


### GMAP (DISMO) ###

library(dismo)

  # Some examples
  # Getting maps for countries
  mymap <- gmap("France")    # choose whatever country
  plot(mymap)
  mymap <- gmap("Spain", type="satellite")    # choose n
  plot(mymap)
  mymap <- gmap("Spain", type="satellite", exp=3)   # ch
  plot(mymap)
  mymap <- gmap("Spain", type="satellite", exp=8)
  plot(mymap)

  mymap <- gmap("Spain", type="satellite", filename="Sp
map as a file in your wd for future use

  # Now get a map for a region drawn at hand
  mymap <- gmap("Europe")
  plot(mymap)
  select.area <- drawExtent()    # now click on the map
  mymap <- gmap(select.area)
  plot(mymap)
  # See ?gmap for many other possibilities


#########################################################

### SPATIAL STATISTICS ###

## Point pattern analysis
  library(spatial)
  library(spatstat)
  library(spatgraphs)
  library(ecespa)     # ecological focus
  # etc (see Spatial Task View)

  # example
  data(fig1)
```

```
  plot(fig1)     # point pattern
  data(Helianthemum)
  cosa12 <- K1K2(Helianthemum, j="deadpl", i="survpl",
         nsim=99, nrank=1, correction="isotropic")
  plot(cosa12$k1k2, lty=c(2, 1, 2), col=c(2, 1, 2), xli
         main= "survival- death",ylab=expression(K[1]-K

### Geostatistics ###
  library(gstat)
  library(geoR)
  library(akima)    # for spline interpolation
  # etc (see Spatial Task View)


  library(spdep)    # dealing with spatial dependence


#######################################################

### INTERACTING AND COMMUNICATING WITH OTHER GIS ###

library(spgrass6)    # GRASS
library(RPyGeo)      # ArcGis (Python)
library(RSAGA)       # SAGA
library(spsextante)  # Sextante


#######################################################

## Other useful packages ##

library(Metadata)     # automatically collates data from
(land cover, pop density, etc) for a given set of coord

#library(GeoXp)     # Interactive exploratory spatial da
  example(columbus)
  histomap(columbus,"CRIME")

library(maptools)
# readGPS

library(rangeMapper)     # plotting species distributior


# Species Distribution Modelling
library(dismo)
library(BIOMOD)
library(SDMTools)

library(BioCalc)    # computes 19 bioclimatic variables
values (tmin, tmax, prec)



#######################################################

### Examples ###
```

```
### SPATIAL VECTOR DATA (POINTS, POLYGONS, ETC) ###


# Example dataset: Get "Laurus nobilis" coordinates frc
laurus <- gbif("Laurus", "nobilis")
# get data frame with spatial coordinates (points)
locs <- subset(laurus, select=c("country", "lat", "lon"

# Making it 'spatial'
coordinates(locs) <- c("lon", "lat")    # set spatial c
plot(locs)

# Define geographical projection
# to look for the appropriate PROJ.4 description look h
http://www.spatialreference.org/

crs.geo <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84
datum WGS84
proj4string(locs) <- crs.geo     # define projection sy
summary(locs)


# Simple plotting
data(wrld_simpl)
summary(wrld_simpl)     # Spatial Polygons Data Frame w
plot(locs, pch=20, col="steelblue")
plot(wrld_simpl, add=T)


### Subsetting
table(locs@data$country)     # see localities by countr

locs.gr <- subset(locs, locs$country=="GR")    # select
plot(locs.gr, pch=20, cex=2, col="steelblue")
plot(wrld_simpl, add=T)
summary(locs.gr)

locs.gb <- subset(locs, locs$country=="GB")     # locs i
plot(locs.gb, pch=20, cex=2, col="steelblue")
plot(wrld_simpl, add=T)


### MAKING MAPS ###

# Plotting onto a Google Map using RGoogleMaps
PlotOnStaticMap(lat = locs.gb$lat, lon = locs.gb$lon, z
19, col="red", FUN = points, add=F)


# Downloading map from Google Maps and plotting onto it
map.lim <- qbbox (locs.gb$lat, locs.gb$lon, TYPE="all")
mymap <- GetMap.bbox(map.lim$lonR, map.lim$latR, destfi
maptype="satellite")
# see the file in the wd
PlotOnStaticMap(mymap, lat = locs.gb$lat, lon = locs.gb
cex=1.3, pch= 19, col="red", FUN = points, add=F)

# using different background
mymap <- GetMap.bbox(map.lim$lonR, map.lim$latR, destfi
maptype="hybrid")
```

```
PlotOnStaticMap(mymap, lat = locs.gb$lat, lon = locs.gb
cex=1.3, pch= 19, col="red", FUN = points, add=F)

# you could also use function gmap in "dismo"
gbmap <- gmap(locs.gb, type="satellite")
locs.gb.merc <- Mercator(locs.gb)     # Google Maps are
This function projects the points to that projection tc
plot(gbmap)
points(locs.gb.merc, pch=20, col="red")


### Plotting onto a Google Map using googleVis (interne
points.gb <- as.data.frame(locs.gb)
points.gb$latlon <- paste(points.gb$lat, points.gb$lon,
map.gb <- gvisMap(points.gb, locationvar="latlon", tipv
                  options = list(showTip=T, showLine=F,
                                 useMapTypeControl=T, width=1
plot(map.gb)
print(map.gb)     # HTML suitable for a web page

##########


# drawing polygons and polylines
mypolygon <- drawPoly()     # click on the map to draw a
when finished
summary(mypolygon)     # now you have a spatial polygon!



### READING AND SAVING DATA

### Exporting KML
writeOGR(locs.gb, dsn="locsgb.kml", layer="locs.gb", dr

### Reading kml
newmap <- readOGR("locsgb.kml", layer="locs.gb")

### Saving as a Shapefile
writePointsShape(locs.gb, "locsgb")

### Reading (point) shapefiles
gb.shape <- readShapePoints("locsgb.shp")
plot(gb.shape)

# readShapePoly   # polygon shapefiles
# readShapeLines  # polylines
# see also shapefile in "raster"



### PROJECTING ###

summary(locs)
# define new projection; look parameters at spatialrefe
crs.laea <- CRS("+proj=laea +lat_0=52 +lon_0=10 +x_0=43
+ellps=GRS80 +units=m +no_defs")
locs.laea <- spTransform(locs, crs.laea)
```

```
plot(locs.laea)

# Projecting shapefile of countries
country <- readShapePoly("ne_110m_admin_0_countries", I
proj4string=crs.geo)     # downloaded from Natural Eart
plot(country)    # in geographical projection
country.laea <- spTransform(country, crs.laea)  # proje

# Plotting
plot(locs.laea, pch=20, col="steelblue")
plot(country.laea, add=T)
# define spatial limits for plotting
plot(locs.laea, pch=20, col="steelblue", xlim=c(1800000
ylim=c(1000000, 3000000))
plot(country.laea, add=T)

####################

### Overlay

ov <- overlay(locs.laea, country.laea)
countr <- country.laea@data$NAME[ov]
summary(countr)



#########################################################

### USING RASTER (GRID) DATA ####


### DOWNLOADING DATA
tmin <- getData("worldclim", var="tmin", res=10)    # th
data on minimum temperature at 10 min resolution
  # can also get other climatic data, elevation, admini

### LOADING A RASTER LAYER
tmin1 <- raster("~/UsingR-GIS/wc10/tmin1.bil")    # Tmin
fromDisk(tmin1)  # values are stored on disk instead of
large rasters)
tmin1 <- tmin1/10    # Worldclim temperature data come
tmin1    # look at the info
plot(tmin1)

?raster    # raster reads many different formats, inclu
netcdf files



### CREATING A RASTER STACK (collection of many raster
projection, spatial extent and resolution)
library(gtools)
list.ras <- mixedsort(list.files("~/UsingR-GIS/wc10/",
pattern=".bil"))
list.ras   # I have just collected a list of the files
temperature values
tmin.all <- stack(list.ras)
tmin.all
```

```
tmin.all <- tmin.all/10
plot(tmin.all)

# BRICKS
tmin.brick <- brick(tmin.all)    # a rasterbrick is simi
(i.e. multiple layers with the same extent and resoluti
must be stored in a single file



### CROP RASTERS
plot(tmin1)
newext <- drawExtent()     # click on the map
tmin1.c <- crop(tmin1, newext)
plot(tmin1.c)

newext2 <- c(-10, 10, 30, 50)    # alternatively, provic
tmin1.c2 <- crop(tmin1, newext2)
plot(tmin1.c2)

tmin.all.c <- crop(tmin.all, newext)
plot(tmin.all.c)



### DEFINE PROJECTION
crs.geo     # defined above
projection(tmin1.c) <- crs.geo
projection(tmin.all.c) <- crs.geo
tmin1.c     # notice info info at coord.ref.

### CHANGING PROJECTION
tmin1.proj <- projectRaster(tmin1.c, crs="+proj=merc +l
+y_0=0 +a=6378137 +b=6378137 +units=m +no_defs")
tmin1.proj    # notice info info at coord.ref.
plot(tmin1.proj)
# can also use a template raster, see ?projectRaster



### PLOTTING
histogram(tmin1.c)
pairs(tmin.all.c)
persp(tmin1.c)
contour(tmin1.c)
contourplot(tmin1.c)
levelplot(tmin1.c)
plot3D(tmin1.c)
bwplot(tmin.all.c)
densityplot(tmin1.c)



### Spatial autocorrelation
Moran(tmin1.c)     # global Moran's I
tmin1.Moran <- MoranLocal(tmin1.c)
plot(tmin1.Moran)


### EXTRACT VALUES FROM RASTER
```

```
View(locs)     # we'll obtain tmin values for our points
locs$tmin1 <- extract(tmin1, locs)     # values are incc
dataframe
View(locs)

# extract values for a given region
plot(tmin1.c)
reg.clim <- extract(tmin1.c, drawExtent())
summary(reg.clim)

# rasterToPoints
tminvals <- rasterToPoints(tmin1.c)
View(tminvals)

## CLICK: get values from particular locations in the m
plot(tmin1.c)
click(tmin1.c, n=3)    # click n times in the map



### RASTERIZE POINTS, LINES OR POLYGONS
locs2ras <- rasterize(locs.gb, tmin1)
locs2ras
plot(locs2ras, xlim=c(-10,10), ylim=c(45, 60), legend=F
plot(wrld_simpl, add=T)


### CHANGING RESOLUTION (aggregate)
tmin1.lowres <- aggregate(tmin1.c, fact=2, fun=mean)
tmin1.lowres
tmin1.c     # compare
par(mfcol=c(1,2))
plot(tmin1.c, main="original")
plot(tmin1.lowres, main="low resolution")
dev.off()



### SPLINE INTERPOLATION
xy <- data.frame(xyFromCell(tmin1.lowres, 1:ncell(tmin1
raster cell coordinates
View(xy)
vals <- getValues(tmin1.lowres)
require(fields)
spline <- Tps(xy, vals)     # thin plate spline
intras <- interpolate(tmin1.c, spline)
intras
plot(intras)
intras <- mask(intras, tmin1.c)
plot(intras)


# SETTING ALL RASTERS TO THE SAME EXTENT, PROJECTION AN
library(climstats)
?spatial_sync_raster



### ELEVATIONS: Getting slope, aspect, etc
elevation <- getData('alt', country='ESP')
```

```
x <- terrain(elevation, opt=c('slope', 'aspect'), unit=
plot(x)

slope <- terrain(elevation, opt='slope')
aspect <- terrain(elevation, opt='aspect')
hill <- hillShade(slope, aspect, 40, 270)
plot(hill, col=grey(0:100/100), legend=FALSE, main='Spa
plot(elevation, col=rainbow(25, alpha=0.35), add=TRUE)


### SAVING AND EXPORTING DATA

# writeraster
writeRaster(tmin1.c, filename="tmin1.c.grd")    # can ex
file types
writeRaster(tmin.all.c, filename="tmin.all.grd")

# exporting to KML (Google Earth)
tmin1.c <- raster(tmin.all.c, 1)
KML(tmin1.c, file="tmin1.kml")
KML(tmin.all.c)      # can export multiple layers



#####################################################

### To learn more ###

# Packages help and vignettes, especially
http://cran.r-project.org/web/packages/raster/vignettes
http://cran.r-project.org/web/packages/dismo/vignettes/
http://cran.r-project.org/web/packages/sp/vignettes/sp.

# CRAN Task View: Analysis of Spatial Data
http://cran.r-project.org/web/views/Spatial.html

# R-SIG-Geo mailing list
https://stat.ethz.ch/mailman/listinfo/R-SIG-Geo

# R wiki: tips for spatial data
http://rwiki.sciviews.org/doku.php?id=tips:spatial-data

# book
http://www.asdar-book.org/

#####################################################
```

Created by Pretty R at inside-R.org