OpenSpat 2017

Geographic data with R: exercises

Lecturer: Manuel Campagnolo (ISA, ULisboa)

1. Consider the data in file `Aragonez.txt`, that describes the locations and yield of vine plants. Columns `rowm` indicates the distance (in meters) along vineyard rows, and column `colm` indicates the distance (in meters) between rows. The origin of the measurements is location LAT=38.4411239°, LONG=-7.5159353° at the SW corner of the set of observations.

   (a) Build an object `SpatialPointsDataFrame` of class `sp` called `AragonezPts` to represent those points in their correct geographic positions. The attribute table should have a single attribute, which is the `yield`. `AragonezPts` should have a cartographic coordinate reference system. To do this you need to download a very high resolution image around the location and identify the direction of the vineyard rows. You should be able to plot `AragonezPts` over the high resolution image and obtain a map similar to the following one:

   Suggestions:

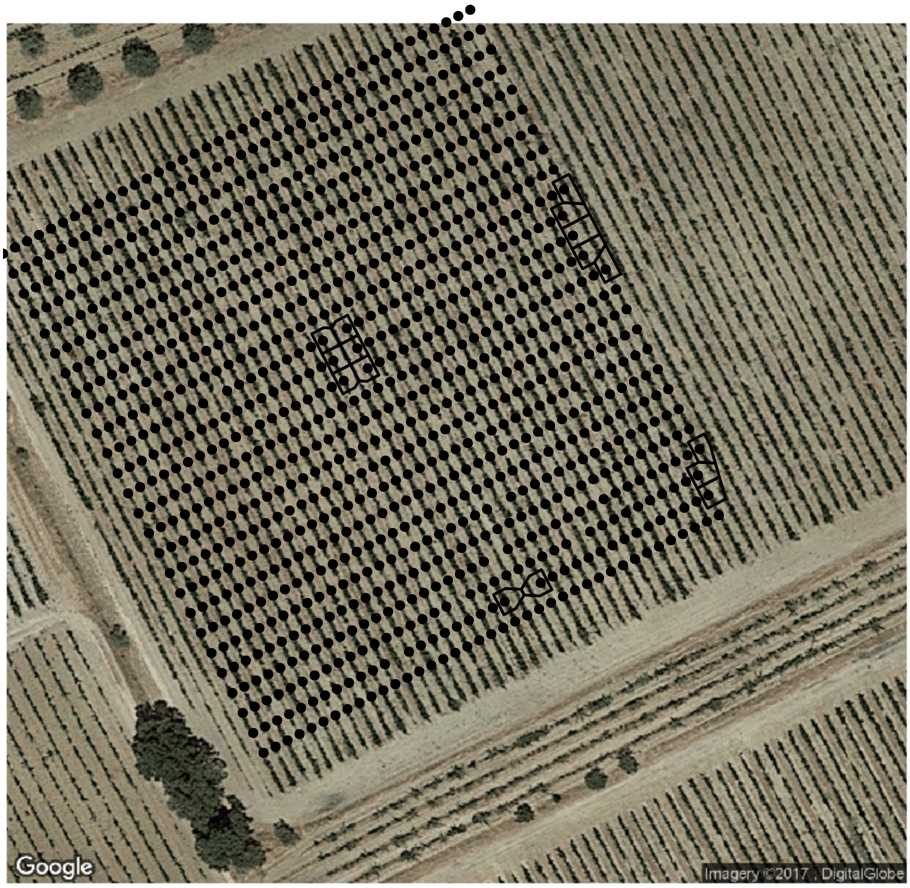
   - The class `SpatialPointsDataFrame` has a simple data structure: it requires an object `SpatialPoints`, which combines a $n \times 2$ matrix of coordinates and a coordinate reference system, and a `data.frame` with n rows and one columns for each attribute;
   - Consider an azimuthal projection for your coordinate reference system;
   - Use `locator(2)` to determine the ends of some vineyard row and define that direction;
   - To rotate the data points, you can simply use matrix multiplication, or you can consider in alternative the R package `rotations`.

(b) Instead of considering just point locations, associate to each observation a polygon which represents its area of influence. The result should be an object of class `SpatialPolygonsDataFrame` called `AragonezVoronoi`. If you plot the results you should get the following (notice that the polygons are not all of the same size since there are locations not occupied by plants).

Suggestions:

- Given a set of points (the locations in `AragonezPts`), one can define **Voronoi polygons**, which are the polygon whose interior consists of all points in the plane which are closer to a particular point than to any other. This is a sensible formalization of the concept of"area of influence";
- To define Voronoi polygons from a 2-columns matrix with coordinates, you can use function `voronoi` from package `dismo` which returns a `SpatialPolygonsDataFrame`; this also requires package `deldir`.
- To limit the extension of the Voronoi polygons on the edge of the set of observations, consider that a maximum distance of 1.875 m around the convex hull of the locations in `AragonezPts`.

(c) Add to the attriibute table of `AragonezVoronoi` the atributes `area`, with the area of each Voronoi polygon, and `yield`, with the yield at that location. Determine the polygons larger than 12 m^2. Your result should correspond to the following figure.



2. The data sets for this exercise are the Landsat 7 multilayer image and the `SpatialPolygonsDataFrame` called `sunflower` used in class. Determine a `SpatialPointsDataFrame` object called `sfsample` with the locations of the centers of the Landsat image pixels, which lie inside sunflower parcels. The points should also lie at least 50 m away from the edge of each parcel so the observations are not too affected by the neighboring crops. The attribute table of `sfsample` should have 6 attributes, that correspond to the reflectance values of the six Landsat bands at the correspond points.

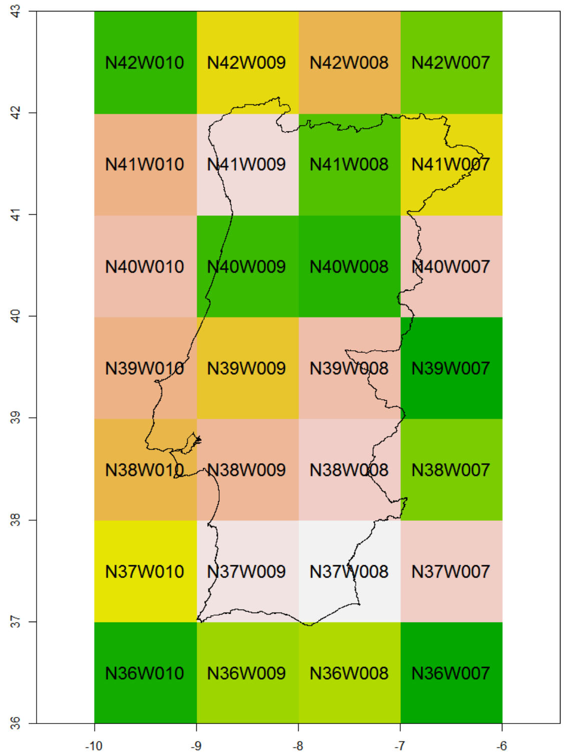
3. Elevation data for most locations on the Earth is readily available to download. For instance, SRTM elevations for Eurasia can be download from site

   `http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/Eurasia/`.

   Those data sets are typically distributed in tiles. For instance, to download and unzip tile `N37W008.hgt` with R, one can use the following set of instructions:

```
urlzip<-"http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/Eurasia/N37W008.hgt.zip"
download.file(url=urlzip,destfile="N37W008.hgt.zip",mode="wb")
unzip(zipfile="N37W008.hgt.zip")
srtm<-raster("N37W008.hgt")
```

   Suppose that you want to create a digital elevation model (DEM) for Continental Portugal, using the relevant tiles depicted in the figure bellow:



(a) Write a R script to download all necessary tiles and merge them together to obtain an elevation model for Portugal;

(b) Determine the location where the slope is steepest according to the DEM, and plot a very high resolution image of that location.

Suggestions:

- If you want to generate automatically a vector of tile names, you may want to consider the function `formatC`. For instance, `formatC(7,width = 3, flag = "0")` returns the string `"007"`;
- To merge two `RasterLayer` objects into one single one, you can use function `merge` from package `raster`;
- To compute the slope, we can use function `terrain` from package `raster`.

4. The command `gBuffer(spgeom, byid=TRUE, width=10)`, where `spgeom` is a `SpatialPolygonsDataFrame`, returns a new `SpatialPolygonsDataFrame` which features are the buffers of the features of `spgeom` determined by the fixed distance `width=10`. For instance, if it is applied to a circular feature with radius 100 m (and area equals to 31415.93 m^2), the output will be a circular feature with radius 110 m (and area equals to 38013.27 m^2).

   Define a new function `myBuffer` which returns a `SpatialPolygonsDataFrame` such that, for each feature `feat` of `spgeom`, `myBuffer` returns a buffer of `feat` which area is <u>twice</u> the area of `feat`.