Lab exercise Thursday – Recent Developments in Biostatistics: Space-Time Models SoSe 2010
S. Greven, M. Höhle and U. Mansmann Date: 8 July 2010

Exercise 1 (Handling `Spatial*` objects)

The aim of this exercise is to practice the creation, handling and visualization of `Spatial*` objects from the package `sp`.

(a) Create an object named `sppo` of class `SpatialPolygons` which consists of a list containing two `Polygons` objects. The first polygon (`ID="p1"`) is the upper triangle of the unit square with corners $(0,1),(1,1)$ and $(1,0)$, the second polygon (`ID="p2"`) contains the lower triangle with corners $(0,1),(1,0)$ and $(0,0)$.
Note: `Polygons` objects consist of a list of objects of class `Polygon`.

(b) Plot `sppo` such that the first polygon is colored green and the second is colored blue. Use the function `text` to add the name of the respective polygon to the plot.
Hints: The call `coordinates(sppo)` returns the centroid of the respective polygons.

(c) Create a `SpatialPoints` object `p`, which contains the two points $(0.25, 0.5)$ and $(0.75, 0.5)$. Add the two points to the plot. Use the function `overlay` to determine for each point in which polygon it is located.

(d) Sample 20 random points within `sppo` by using the function `spsample` and add the points to the plot. Use the symbol "1" for points located in the first polygon and "2" for points in the second polygon, respectively.

(e) Attach to each point from (d) a random value `value` between zero and one. Name the resulting `SpatialPointsDataFrame` object `p.value`. Calculate the mean of `value` for all points located in the first and second polygon, respectively.

(f) Use the function `maptools::unionSpatialPolygons`, to obtain a `SpatialPolygons` object which combines both polygons of `sppo` into a single polygon.

Exercise 2 (Working with shapefiles)

The folder `Data` available as ZIP file from the webpage contains a WGS84 shapefile `vg2500_geo84-pop` with the administrative borders of the 413 districts in Germany (State: April 2009). The data originate from the German Federal Agency for Cartography and Geodesy. The corresponding attribute table contains two columns which contain the population (in thousands) and population density (people per square km). Theses numbers have been extracted from the "Gemeindeverzeichnis GV2000" of the German Federal Statistics Office.

(a) Read the data in `vg2500_krs-pop.shp` using the function `maptools::readShapeSpatial`. When loading the data use `proj4str=CRS("+proj=longlat +datum=WGS84 +ellps=WGS84")` in order to take the WGS84 georeferencing into account. Plot the resulting `SpatialPolygonsDataFrame` object.
Hint: An alternaive is to use the `rgdal::readOGR` function which automagically determines the projection information.

(b) The column `RS` in the attribute table contains for each district the official municipality key. Select all districts in the federal state of Bavaria (RS key starts with 09) and plot the result. .
Hint: The functions `as.character` and `substr` could be of help.

(c) Use the function `classInt::classIntervals` to assign each district a color according to its population density. Also add a corresponding legend to the graph.

(d) Export the boarders of the administrative districts in Bavaria as KML file and open the generated KML file with Google Earth.

(e) The `spplot` function for a `SpatialPolygonDataFrames` with exactly one column can automatically generate a color legend using the argument `col.regions=pal,colorkey=list(col=pal,at=intervals$brks)`. Here, `pal` is a vector with the colors to be used in the plot and `intervals` is a `classIntervals` object. Generate a plot of the population density of each district with `spplot`.

More on next page!

Exercise 3 (R as SpatialAnalyst)

The file `Data/rain2007-07.asc` contains the amount of precipitation (measured as mm/m^2) in Germany during July 2007 as a 866×654 pixel raster image in the ESRI ASCII format. In this exercise your task is to use R to calculate the average amount of rainfall in each district of Baden-Württemberg.

(a) Read the Shapefile `Data/bw-kreise.shp` containing the the necessary administrative districts. Use the function `readGDAL` to read the ASCII file `Data/regen2007-07.asc` – the resulting `SpatialGRidDataFrame` will contain a column denoted `band1` containing the rainfall values. No additional information on the geo-referencing is necessary. Visualize the shapefile and the ASCII file in one plot.

(b) In the rainfall grid, set the value of each cell not located within Baden-Württemberg to the value `NA`. Convert the resulting object to a `SpatialPixelsObjekt` with the name `rain.bw`. Visualize districts and the rainfall in `rain.bw` in one plot.
Hint: Use the `overlay` function.

(c) Compute the average amount of rainfall in each administrative region of Baden-Würtemmberg and add this information to the attribute table of `bw-kreise`. Visualize the result.