Lab 8 SPATIAL INTERPOLATION

NEW SKILLS
- In this lab we will study various methods of spatial interpolation.

**QGIS**

1. Create a folder for your output called Lab 8. Click the Start QGIS. Interpolation in QGIS GRASS can be done directly the Interpolation plugin. Use Plugins->Manage and Install Plugins to make sure this is installed. Add the vector layer field5samples.shp from the wheat directory.

3. Interpolation can be carried out using either a triangulated irregular network (TIN) or inverse distance weighting (IDW). We will first carry out the IDW interpolation. Select Raster -> Interpolation -> Interpolation. Select the Vector layer field5samples and the Interpolation attribute YIELD. Click the Add button. This will give us the values of crop yield at the selected sample points. We can then interpolate them and compare the results with the actual yield dat. Choose the Interpolation method Inverse Distance Weighting (IDW) and leave all the other settings as they are. Place the output in a file called field5interp in the Lab8 folder. It will automatically be given the extension asc.

4. Load the raster yield file field5.asc and compare the two files. The result is not a particularly good interpolation. Experiment with TIN interpolation and with the interpolation parameter, adjusted by clicking on the wrench icon in the upper right corner of the plugin window.

**R**

5. It is also possible to carry out interpolation in R, with better results. There are a number of packages that include some form of interpolation; we will use the package gstat. Install this package in your R system. First we will read in the data to interpolate. Execute the following lines:

```r
library(gstat)
library(maptools)
f5.dat <- readShapePoints("c:\qgislab\wheat\field5samples.shp")
```

It is possible to produce a (bubble map) that shows the distribution of yield at each point. Type

`bubble(f5.dat, "YIELD")`

and observe the result.

6. The next step is to create the coordinates of the grid on which we will do the interpolation. We will use a 4m cell size running from an Easting of 591994m to 592494m and a Northing of 4270307 to 4271187m. This is a grid with 220 rows and 125 columns. First we will create a pair of vectors of the appropriate size. Type:

```r
Left <- 591994
Right <- 592490
Top <- 4271187
Bottom <- 4270311
```
cell.size <- 4

Next we will create the grid in which to interpolate the data. We can use the R function expand.grid() to do this. Because the grid is so large, first try this to see how it works:

expand.grid(x = seq(from = 1, to = 7, by = 2), y = seq(from = 1, to = 10, by = 3))

You don’t actually need the from, by and to, so we can get the grid we want by typing (note the minus sign in from of the second cell.size, because y is decreasing).

grid.xy <- expand.grid(x = seq(Left, Right, cell.size), y = seq(Top, Bottom, -cell.size))

The next step is to tell the system that this is gridded spatial data:

coordinates(grid.xy) <- ~x + y
gridded(grid.xy) = TRUE

Finally we can do an IDW interpolation. This is done by using the function krige but not specifying a model:

yield.idw <- krige(YIELD ~ 1, f5.dat, grid.xy)
spplot(yield.idw["var1.pred"])

7. We can also do a kriged interpolation using the function krige. The steps are to compute the variogram, fit the variogram, and use this in the function krige:

yield.vgm <- variogram(YIELD ~ 1, f5.dat)
yield.fit <- fit.variogram(yield.vgm, model = vgm(1, "Sph", 700, 1))
yield.krig <- krige(YIELD ~ 1, f5.dat, grid.xy, model = yield.fit)
spplot(yield.krig["var1.pred"])