Homework 8 Time & Space

Brendan Smith

May 13, 2016

Objective Statement: This assignment is an extension of Homework 8, in that we are still analyzing the Cosumnes River, but at a different location. Specifically, we are looking at the Triangle Floodplain north of the Acidental forest. We will be taking a new dataset and create a Shiny web app to easily share and interact with the data.

```
# Load libraries
library(maptools)
```

```
## Loading required package: sp

## Checking rgeos availability: FALSE
##      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpcli
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

```
library(akima)
```

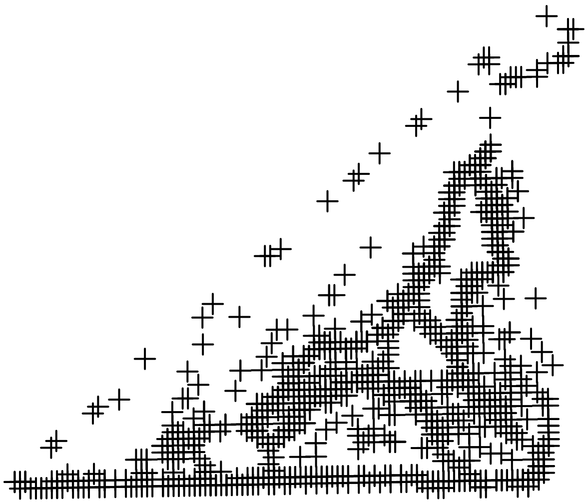
Methods: The methods for data analysis today are an introduction to shape files and importing them. We follow this with interpolation of the data across the entire area of study, and plotting this as heat map image with a contour line overlay. We then touch upon perspective 3D plots, and follow up with the combination of everything we've learned in this assignment to create a Shiny web App.

Data: The data utilized in this assignment was taken at the Triangle Floodplain north of Accidental Forest. We have access to several types of data including: turbidity, chlorophyl, latitude, longitude, elevation, etc. It isn't clear how these data are collected, so not much can be said on the accuracy of the set.

Code: A few new functions are introduced in this assignment. First is `readShapePoints()`, which allows the user to load a shape file into a dataframe. We begin by importing `allobs.shp` and performing a quick EDA:

```
# Read Shapefile
pt.shp = readShapePoints("allobs.shp")

# Perform EDA
plot(pt.shp)
```



```r
head(pt.shp)
```

```
##            coordinates ID  DateTime_  Temp  DO__ SpCond TDS Turbidity_
## 0 (640624.5, 4237716)  1 2005-02-16 10.22 103.7     99  64       11.5
## 1 (639727.4, 4236968)  2 2005-02-16 14.94  55.1    161 105       13.7
## 2 (639753.8, 4236958)  3 2005-02-16 13.89  53.3    147  95        9.5
## 3 (639788.6, 4236969)  4 2005-02-16 13.52  49.6    149  97        9.4
## 4 (639858.6, 4236970)  5 2005-02-16 14.01  60.8    140  91       14.1
## 5 (639928.8, 4236961)  6 2005-02-16 13.89  72.2    137  89        7.5
##   Chlorophyl Latitude Longitude  Elevati  SpineDi direction        X
## 0        2.4  38.2765 -121.3923 4.078768 320.3623       211 640624.5
## 1        6.9  38.2699 -121.4027 3.600000 420.1952        78 639727.4
## 2        7.6  38.2698 -121.4024 3.709091 397.3160        76 639753.8
## 3        7.6  38.2699 -121.4020 3.600000 359.4496        76 639788.6
## 4        7.9  38.2699 -121.4012 3.600000 292.2054        73 639858.6
## 5        7.0  38.2698 -121.4004 3.600000 230.7899        65 639928.8
##         Y
## 0 4237716
## 1 4236968
## 2 4236958
## 3 4236969
## 4 4236970
## 5 4236961
```

```r
summary(pt.shp)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##                  min       max
## coords.x1  639709.9  640633.2
## coords.x2 4236956.9 4237737.9
## Is projected: NA
## proj4string : [NA]
## Number of points: 1108
## Data attributes:
##        ID            DateTime_                Temp             DO__
```

```
##  Min.   :   1.0   Min.   :2005-02-16   Min.   :10.22   Min.   : 38.60
##  1st Qu.: 277.8   1st Qu.:2005-02-17   1st Qu.:12.04   1st Qu.: 85.17
##  Median : 554.5   Median :2005-02-18   Median :13.15   Median : 99.10
##  Mean   : 554.5   Mean   :2005-02-20   Mean   :13.27   Mean   : 97.81
##  3rd Qu.: 831.2   3rd Qu.:2005-02-23   3rd Qu.:14.00   3rd Qu.:112.70
##  Max.   :1108.0   Max.   :2005-02-28   Max.   :21.08   Max.   :197.40
##      SpCond            TDS            Turbidity_      Chlorophyl
##  Min.   : 96.0   Min.   : 62.00   Min.   : 4.60   Min.   :-0.800
##  1st Qu.:101.0   1st Qu.: 65.00   1st Qu.: 7.10   1st Qu.: 3.000
##  Median :126.0   Median : 82.00   Median :11.90   Median : 4.200
##  Mean   :119.8   Mean   : 77.82   Mean   :12.35   Mean   : 5.223
##  3rd Qu.:129.0   3rd Qu.: 84.00   3rd Qu.:17.00   3rd Qu.: 6.800
##  Max.   :205.0   Max.   :133.00   Max.   :34.30   Max.   :23.500
##     Latitude       Longitude         Elevati          SpineDi
##  Min.   :38.27   Min.   :-121.4   Min.   :2.122   Min.   :-3.403e+38
##  1st Qu.:38.27   1st Qu.:-121.4   1st Qu.:2.914   1st Qu.: 3.600e+01
##  Median :38.27   Median :-121.4   Median :3.264   Median : 5.700e+01
##  Mean   :38.27   Mean   :-121.4   Mean   :3.195   Mean   :-3.071e+35
##  3rd Qu.:38.27   3rd Qu.:-121.4   3rd Qu.:3.400   3rd Qu.: 1.150e+02
##  Max.   :38.28   Max.   :-121.4   Max.   :4.600   Max.   : 4.380e+02
##    direction              X                Y
##  Min.   :-32768.0   Min.   :639710   Min.   :4236957
##  1st Qu.:    83.0   1st Qu.:640170   1st Qu.:4237016
##  Median :   175.0   Median :640365   Median :4237115
##  Mean   :   157.4   Mean   :640309   Mean   :4237160
##  3rd Qu.:   295.0   3rd Qu.:640474   3rd Qu.:4237268
##  Max.   :   360.0   Max.   :640633   Max.   :4237738
```

```r
levels(as.factor(pt.shp$DateTime_)) # view individual dates
```

```
## [1] "2005-02-16" "2005-02-17" "2005-02-18" "2005-02-23" "2005-02-28"
```

```r
unique(pt.shp$DateTime_) # TIMTOWTDI
```

```
## [1] "2005-02-16" "2005-02-28" "2005-02-23" "2005-02-18" "2005-02-17"
```

We notice that there are two forms of spatial information, lat/lon and X/Y. This is because lat/lon is truly positional data on a spherical coordinate system, so plotting on a flat plane is not ideal. There is instead another coordinate type, Universal Transverse Mercator coordinate system. This is a projection of the earth onto several flat planes across the earth, essentially breaking the world into a bunch of flat 2D planes. Each plane has a corresponding zone, 60 in total and this is the data we are seeing in the X and Y columns.

Upon further inspection of the EDA, we notice that there are really only five days where data was collected. This makes things simple for us, as we can break the data into several data subsets.

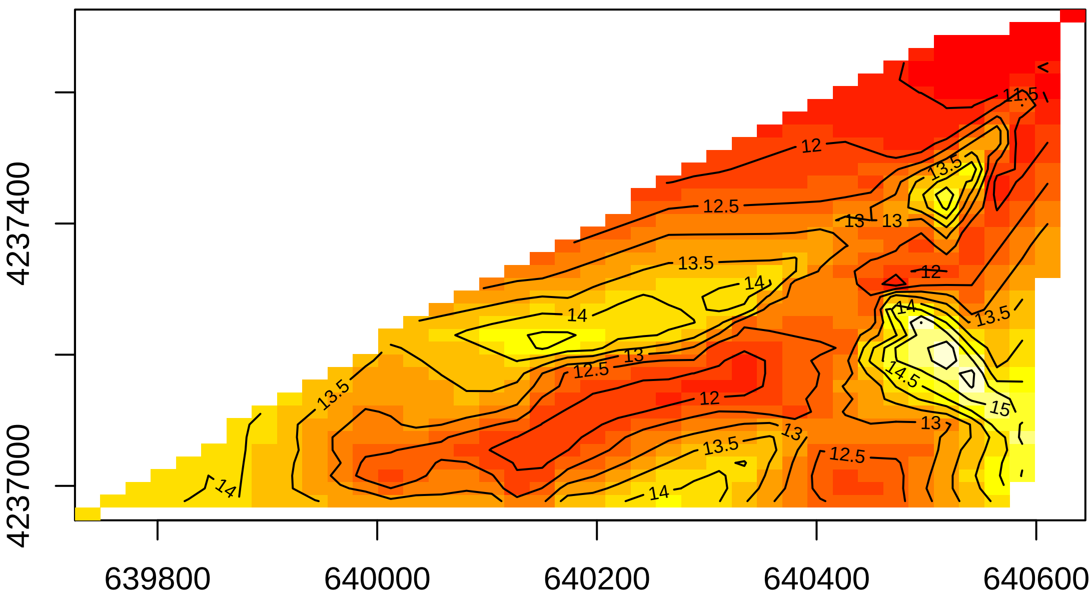
Another function introduced is `interp()`, which allows us to interpolate subsets of data (given there are 2D coordinates in the subset as well) into a raster image.

Results: ##Step 1 - Spatial Interpolation## We start by subsetting the data for two dates:

```r
# Create subets for 2/18 and 2/23
pt.shp.Feb18 = pt.shp[pt.shp$DateTime_ == "2005-02-18",]
pt.shp.Feb23 = pt.shp[pt.shp$DateTime_ == "2005-02-23",]
```

We then apply a spatial interpolation for the February 18 dataset, focusing on temperature, dissolved oxygen, and chlorophyl:

```r
tempinter.Feb18 = interp(pt.shp.Feb18$X,pt.shp.Feb18$Y,pt.shp.Feb18$Temp,duplicate='mean')
dointer.Feb18 = interp(pt.shp.Feb18$X,pt.shp.Feb18$Y,pt.shp.Feb18$DO__,duplicate='mean')
chlinter.Feb18 = interp(pt.shp.Feb18$X,pt.shp.Feb18$Y,pt.shp.Feb18$Chlorophyl,duplicate='mean')
image(tempinter.Feb18)
contour(tempinter.Feb18,add=T)
```



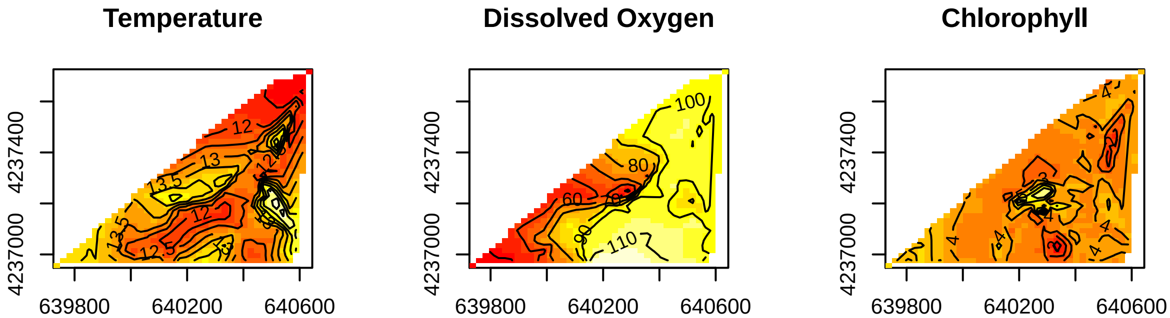

```r
# note: using col=cm.colors(24) makes the contour lines less visible
```

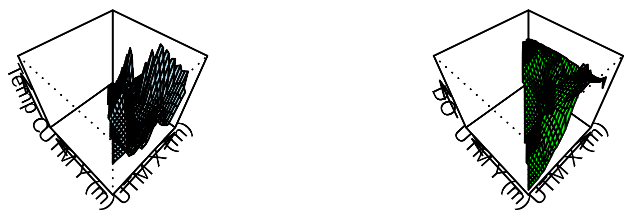
Step 2 - Perspective Plot

Now that we've mastered the `image()` and `contour()` plots, we are introduced to the perspective plots (3D plots). These are extremely straight forward to use, and angles are easily controlled.

```r
# Create a layout matrix to organize the plots
layout(matrix(c(1,2,3,4,5,6),2,3, byrow=TRUE))
# Contour plots
# Temperature
image(tempinter.Feb18,main="Temperature")
contour(tempinter.Feb18,add=T)
# Dissolved Oxygen
image(dointer.Feb18,main="Dissolved Oxygen")
contour(dointer.Feb18,add=T)
# Chlorophyll
image(chlinter.Feb18,main ="Chlorophyll")
contour(chlinter.Feb18,add=T)

persp(tempinter.Feb18, theta=-45, phi=50, col="lightblue", zlab="Temp C", xlab="UTM X (m)", ylab="UTM Y
persp(dointer.Feb18, theta=-45, phi=50, col="green", zlab="DO", xlab="UTM X (m)", ylab="UTM Y (m)")
persp(tempinter.Feb18, theta=-45, phi=50, col = "red", zlab="Chlorophyll", xlab="UTM X (m)", ylab="UTM
```





Step 3 - Shiny Map

See attached `ui.R` and `server.R`.

Discussion: The methods of data analysis introduced in this assignment are, I believe, the most useful for data visualization. The Shiny Map makes it incredibly easy to share and manipulate the data, and I imagine there is a huge possibility to automate the update of server held data.

By examining the image and perspective plots that they differ only in the way three-dimensional data is presented. With the perspective plot, a wire-frame mesh is created and the data is visualized in 3D. The image plot creates a heat map of the values on a 2D plane. Both have their advantages; however, in the case of this dataset, I believe the image plot produces a cleaner visualization

Limitations: I could not get the topo.colors, heat.colors, or cm.colors to work for the perspective plots. Another limitation I encountered was trying to utilize the elevation data from the dataset, it repeatedly crashed my R Studio when trying to interpolate.