# INFX 573: Problem Set 4 - Data Analysis

*Rajendran Seetharaman*

*Due: Thursday, November 2, 2017*

**Collaborators:**

**Instructions:**

1. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
2. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
3. Collaboration on problem sets is fun and useful! However, you must turn in your individual write-up in his or her own words and his or her own work. The names of your collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
4. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly, rename the R Markdown file to `YourLastName_YourFirstName_ps4.Rmd`, knit a PDF and submit both the markdown and the PDF file on Canvas.

**The Task**

The problem set is inspired by a real-world situation and is deliberately somewhat vague. Your task is to understand the data, convert it into a suitable format, and find the tools that produce the desired output. Note: You are asked to produce a map but you don't have to use dedicated mapping tools like *ggmap* and shapefiles, just ordinary plotting will do.

You are working at PredicitiveAnalytics LLC. One day your Most Important Customer comes to you and says:

*I need a temperature and precipitation map of Europe for January and July. It must be based on the most recet NOAA long term means data from NOAA webpage, the v401 format. And I need it by Thursday, November 2nd, 5:30pm. I just need a color map, it does not have to be anything fancy with borders and cities and rivers on it. Just the temperature and rain, plotted in a way I can understand would do.*

Download the data and produce such maps for temperature and precipitation (do not use the tools on the website). Make sure to explain and label your data sources and units of measurement. Try to tune the plot with suitable colors, scales, etc, to impress your Important Customer. Comment, or otherwise explain your code, and briefly discuss the results.

Suggestions:

- If you use *ggplot* for plotting, add coordinate transformation `+ coord_map()` (requires *mapproj* library). This ensures the map will be in a valid map projection. You may experiment with different projections.

## Solution:

Observations:

For the precipitation patterns in europe, the following are my observations. In both January and July, the coastal areas generally have higher precipitation than the interiors.During the month of January, most places in europe have low precipitation with the exception of certain coastal areas like the western coast of Norway. The July month shows higher precipitation for larger number of countries in Europe which are even in the interiors.

For the temperature patterns in europe, the following are my observations. Winters (January) are extremely harsh at a lot of northern countries like Norway, Sweden, Denmark, Finland, with temperatures droppong to -15 degrees celsius. Places in the south like France and Italy are relatively warmer with temperatures in the range 5-10 degrees celsius. Summers are warm with temperatures going as high as 35 degrees in certain places in the south. Places in the north like Sweden, Denmark are still relatively cooler with temperatures in the range 10-15 degrees celsius. This could be due to the fact that they are closer to the poles and are not exposed to the vertical rays of the sun like the equatorial regions.

```
library(ncdf4)
library(ggmap)
```

```
## Loading required package: ggplot2
```

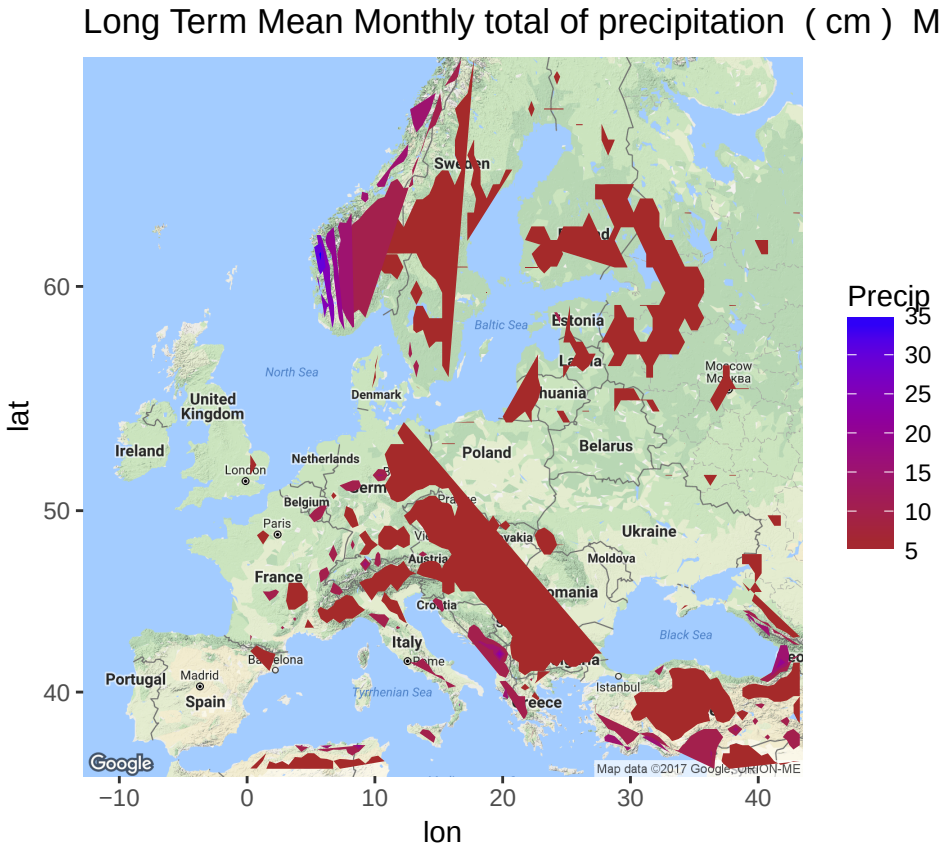
```
#precipitation data
#precip data file path
filepath <- "C:/Users/Rajmohan Seetharaman/Downloads/"
filename <- "precip.mon.ltm.v401"
filesource <- paste(filepath, filename, ".nc", sep="")
var_name <- "precip"
options(warn=-1)
#open file
nc_file <- nc_open(filesource)
#extract the longitude
lon <- ncvar_get(nc_file,"lon")
nlon <- dim(lon)
#extract latitude
lat <- ncvar_get(nc_file,"lat")
nlat <- dim(lat)
#extract time
time <- ncvar_get(nc_file,"time")
nt <- dim(time)
tunits <- ncatt_get(nc_file,"time","units")
#extract precipitation values
tmp_array <- ncvar_get(nc_file,var_name)
#extract data description and precipitation units
dlname <- ncatt_get(nc_file,var_name,"long_name")
dunits <- ncatt_get(nc_file,var_name,"units")
legend <- paste(dlname$value," (",dunits$value,")")
#get data fill value
fillvalue <- ncatt_get(nc_file,var_name,"_FillValue")

#create function for plotting precipitation
precipplot=function(month,legend){
  #extract data slice for particular month
  tmp_slice <- tmp_array[,,month]
#create dataframe from precipitation data
grid <- expand.grid(lon=lon, lat=lat)
lonlat <- as.matrix(expand.grid(lon,lat))
dim(lonlat)
tmp_vector <- as.vector(tmp_slice)
```

```
tmp_dataframe <- data.frame(cbind(lonlat,tmp_vector))
names(tmp_dataframe) <- c("lon","lat",paste(var_name))
#get europe map
euromap <- get_map(location="Europe",zoom=4,maptype = "terrain")
#plot precipitation data pn map as a contour
map <- ggmap( euromap ) +
    stat_contour( data = tmp_dataframe, geom="polygon",
                  aes( x = lon, y = lat, z = round(precip), fill = ..level.. ) ) +
    scale_fill_continuous( name = "Precip", low = "brown", high = "blue" )+
  labs(title=paste(legend," Month=",month))
map
}
#call plot function for months 1 and 7 (Jan and July)
precipplot(1,legend)
```

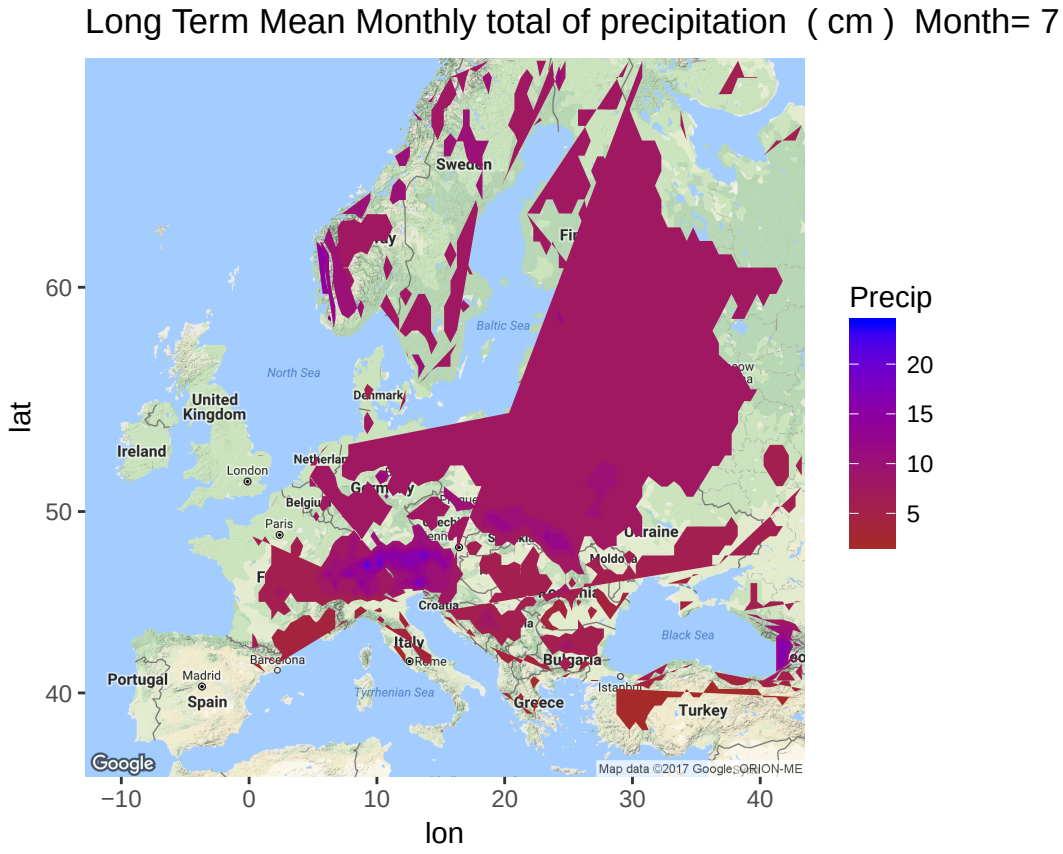
```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Europe&zoom=4&size=640x640&scale=
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Europe&sensor=false
```




```
precipplot(7,legend)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Europe&zoom=4&size=640x640&scale=
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Europe&sensor=false
```

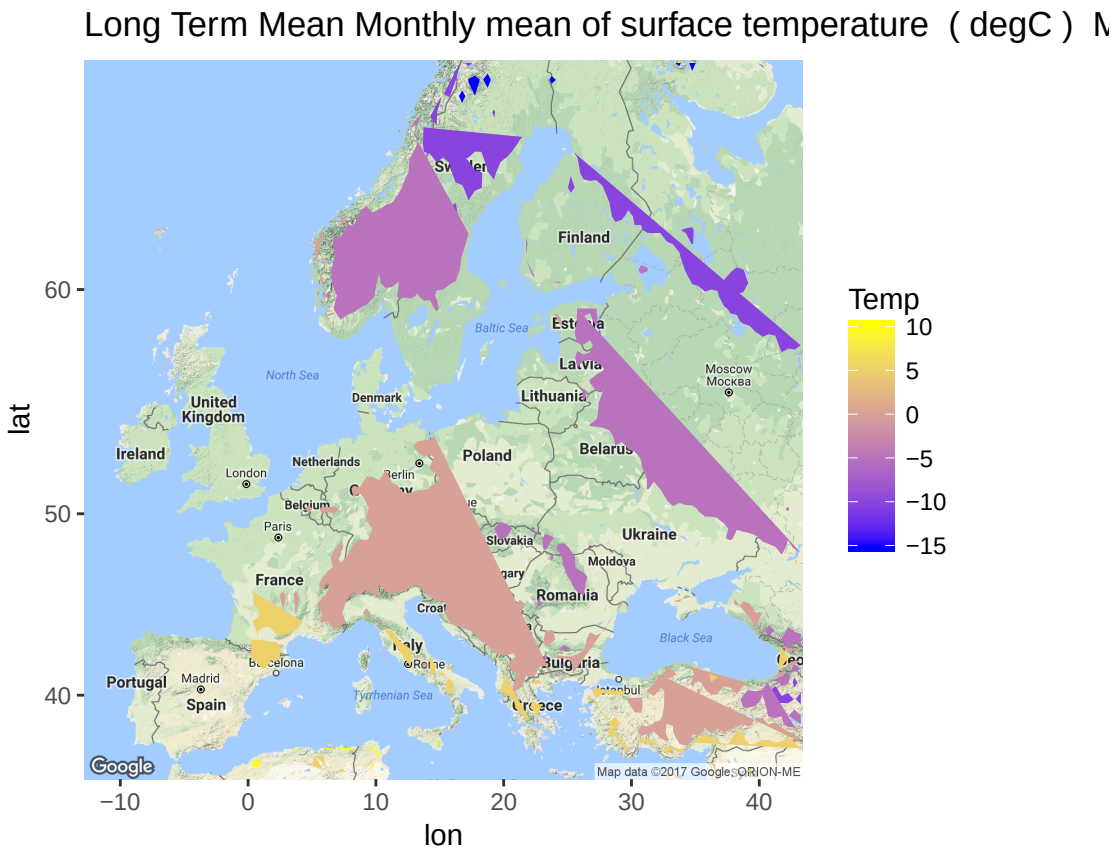



```
# temperature data
#temp data path
ncpath <- "C:/Users/Rajmohan Seetharaman/Downloads/"
ncname <- "air.mon.ltm.v401"
ncfname <- paste(ncpath, ncname, ".nc", sep="")
var_name <- "air"
#open temp data file
#extract longitude, latitude data
nc_file <- nc_open(ncfname)
lon <- ncvar_get(nc_file,"lon")
nlon <- dim(lon)
lat <- ncvar_get(nc_file,"lat")
nlat <- dim(lat)
time <- ncvar_get(nc_file,"time")
nt <- dim(time)
tunits <- ncatt_get(nc_file,"time","units")
tmp_array <- ncvar_get(nc_file,var_name)
#extract description and temperature units
dlname <- ncatt_get(nc_file,var_name,"long_name")
dunits <- ncatt_get(nc_file,var_name,"units")
legend <- paste(dlname$value," (",dunits$value,")")
#extract fill value
fillvalue <- ncatt_get(nc_file,var_name,"_FillValue")
tmp_array[tmp_array==fillvalue$value] <- NA

#temperature map plotting function
```

```
tempplot=function(month,legend){
  #extract temperature data for specified month
    tmp_slice <- tmp_array[,,month]
#create temperature data frame
grid <- expand.grid(lon=lon, lat=lat)
lonlat <- as.matrix(expand.grid(lon,lat))
dim(lonlat)
tmp_vector <- as.vector(tmp_slice)
tmp_dataframe <- data.frame(cbind(lonlat,tmp_vector))
names(tmp_dataframe) <- c("lon","lat",paste(var_name))
# create europe map
euromap <- get_map(location="Europe",zoom=4,maptype = "terrain")
#plot temperature contour on europe map
map <- ggmap( euromap ) +
    stat_contour( data = tmp_dataframe, geom="polygon",
                  aes( x = lon, y = lat, z = air, fill = ..level.. ) ) +
    scale_fill_continuous( name = "Temp", low = "blue", high = "yellow" )+
  labs(title=paste(legend," Month=",month))
map
}
#plot map for months JAn and July
tempplot(1,legend)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Europe&zoom=4&size=640x640&scale=
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Europe&sensor=false
```




```
tempplot(7,legend)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Europe&zoom=4&size=640x640&scale=
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Europe&sensor=false
```