INFX 573: Problem Set 4 - Data Analysis

Namrata Kolla

Due: Thursday, November 2, 2017

Collaborators: Tapasvi Bansal

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 recent 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:

About the data:

- Monthly climatology of precipitation (monthly total in cm) and air temperature (monthly mean in degrees C)
- Taken at the surface
- Time series spanning 1900 to 2014
- .nc is a NetCDF file that's a format for creating and distributing arrays of gridded data (often used for climate data)

Load temperature data:

```
# install.packages("ncdf4")
library(ncdf4)
nc <- nc_open("air.mon.mean.v401.nc")

  # Using https://www.r-bloggers.com/a-netcdf-4-in-r-cheatsheet/
airtemp_alltime <- ncvar_get(nc, attributes(nc$var)$names[1])

dim(airtemp_alltime) # 720, 360, 1380
```

```
## [1]  720  360 1380
```

```
  # 1380 / 12 (months) = 115 (years of data)

# Learn about data's attributes
ncatt_get(nc, attributes(nc$var)$names[1])
```

```
## $long_name
## [1] "Monthly mean of surface temperature"
##
## $missing_value
## [1] -9.96921e+36
##
## $units
## [1] "degC"
##
## $var_desc
## [1] "Air Temperature"
##
## $level_desc
## [1] "Surface"
##
## $statistic
## [1] "Mean"
##
## $parent_stat
## [1] "Other"
##
## $standard_name
## [1] "air_temperature"
##
## $cell_methods
## [1] "time: mean"
##
## $valid_range
## [1] -90  50
##
```

```
## $actual_range
## [1] -79.3  43.9
##
## $dataset
## [1] "Univ. of Delaware Precipitation and Air Temp v4.01"
```

Notes:

- Data is monthly mean of surface temperature
- Missing values will be -9.96921e+36
- Units are degC
- Valid range: -90 50
- Actual range: -79.3 43.9

Tidy temperature data:

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
# grabbed one year's air temperature data
airtemp_Jan <- (airtemp_alltime[,,1]) # January will be month 1
dim(airtemp_Jan)
```

```
## [1] 720 360
```

```
  # 720, 360 which confirms this is one layer of data (January 1900)

# turns matrix into a df we can tidy
airtemp_df_Jan <- data.frame(airtemp_Jan)
colnames(airtemp_df_Jan) <- c(1:360)
lon_vals <- c(1:720)
airtemp_df_Jan$lon <- lon_vals

# tidy up data
airtemp_df_Jan <- gather(airtemp_df_Jan,"lat","temp",1:360)
airtemp_df_Jan$lat <- as.integer(airtemp_df_Jan$lat)
```

Plot data, Mean Temp in January:

```
# install.packages('mapproj')
library(mapproj)
```

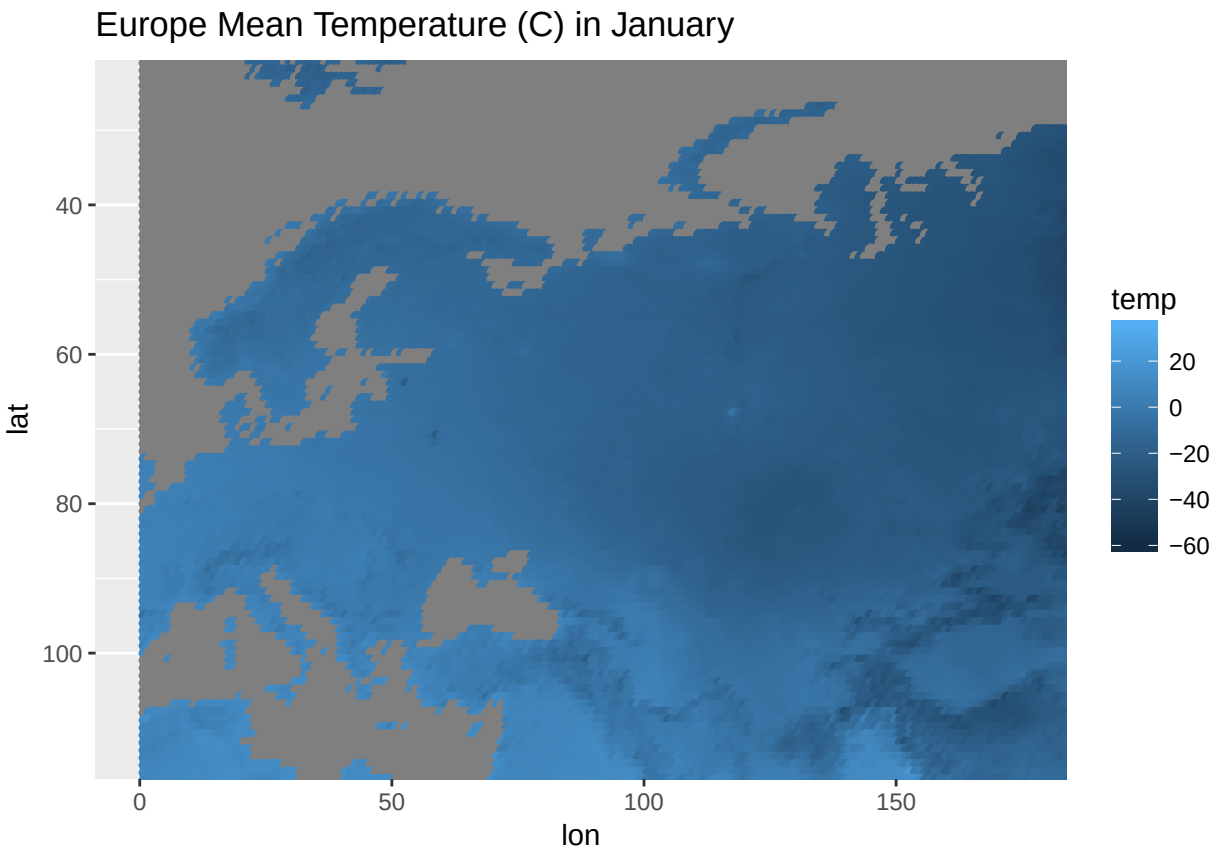
```
## Loading required package: maps
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
  # http://www.milanor.net/blog/maps-in-r-plotting-data-points-on-a-map/

ggplot() +
  geom_point(data = airtemp_df_Jan, aes(x=lon, y = lat, color=temp)) +
  coord_cartesian(xlim=c(0,175),ylim=c(25,112.5)) +
  scale_y_reverse() +
  ggtitle("Europe Mean Temperature (C) in January")
```



Repeat process, Temp for July:

```
airtemp_Jul <- (airtemp_alltime[,,7]) # July will be month 7
dim(airtemp_Jul)
```

```
## [1] 720 360
```

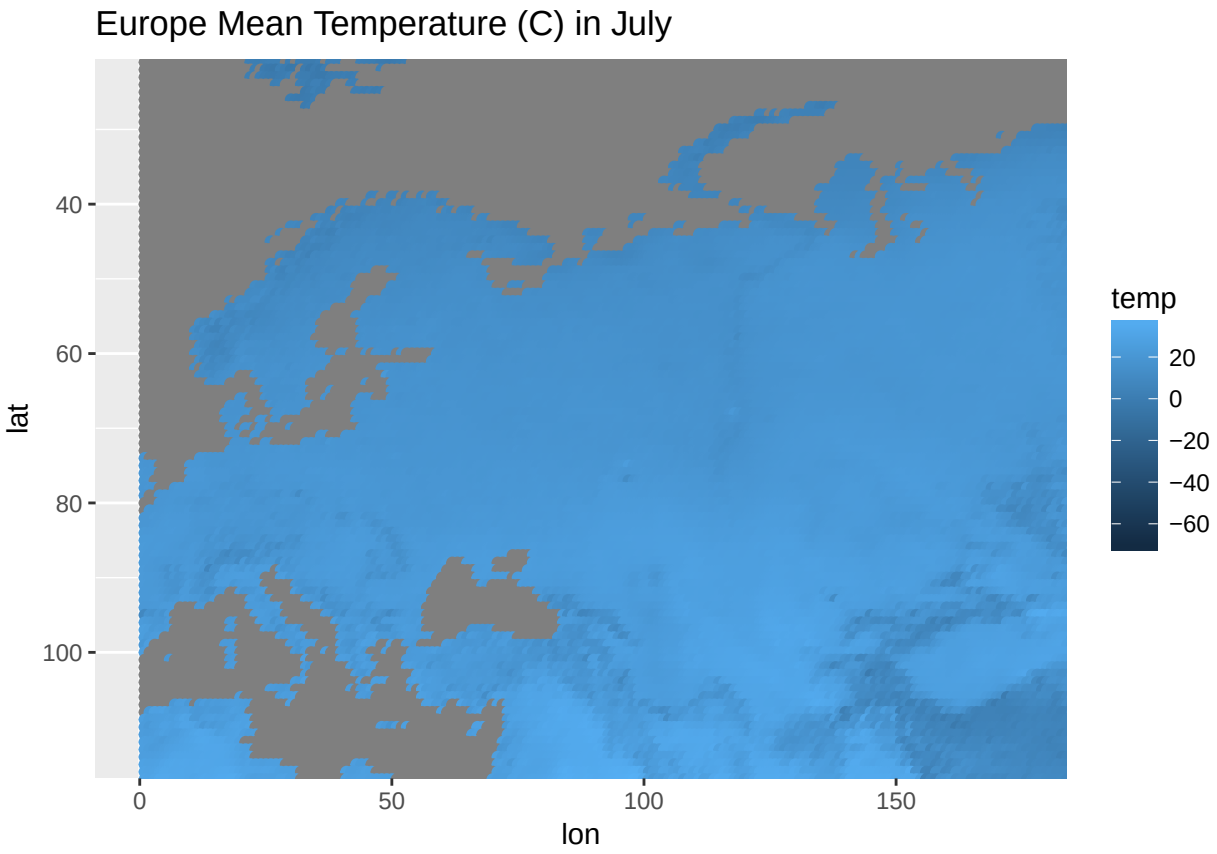
```
  # 720, 360 which confirms this is one layer of data (July 1900)

# turns matrix into a df we can tidy
airtemp_df_Jul <- data.frame(airtemp_Jul)
colnames(airtemp_df_Jul) <- c(1:360)
lon_vals <- c(1:720)
```

```
airtemp_df_Jul$lon <- lon_vals

# tidy up data
airtemp_df_Jul <- gather(airtemp_df_Jul,"lat","temp",1:360)
airtemp_df_Jul$lat <- as.integer(airtemp_df_Jul$lat)

ggplot() +
  geom_point(data = airtemp_df_Jul, aes(x=lon, y = lat, color=temp)) +
  coord_cartesian(xlim=c(0,175),ylim=c(25,112.5)) +
  scale_y_reverse() +
  ggtitle("Europe Mean Temperature (C) in July")
```



Repeat loading data, now for precipitation:

```
nc <- nc_open("precip.mon.total.v401.nc")

precip_alltime <- ncvar_get(nc, attributes(nc$var)$names[1])

dim(precip_alltime) # 720, 360, 1380 just like with air temperatures

## [1]  720  360 1380

  # 1380 / 12 (months) = 115 (years of data)

# Learn about data's attributes
ncatt_get(nc, attributes(nc$var)$names[1])

## $missing_value
```

```
## [1] -9.96921e+36
##
## $units
## [1] "cm"
##
## $var_desc
## [1] "Precipitation"
##
## $level_desc
## [1] "Surface"
##
## $statistic
## [1] "Total"
##
## $parent_stat
## [1] "Other"
##
## $long_name
## [1] "Monthly total of precipitation"
##
## $cell_methods
## [1] "time: sum"
##
## $valid_range
## [1] -90  50
##
## $avg_period
## [1] "0000-01-00 00:00:00"
##
## $actual_range
## [1]   0.00 776.75
##
## $dataset
## [1] "Univ. of Delaware Precipitation and Air Temp v4.01"
```

Notes:

- Data is monthly total of precipitation
- Missing values will be -9.96921e+36
- Units are cm
- Actual range: 0 776.75

Tidy precipitation data:

```
precip_Jan <- (precip_alltime[,,1]) # January will be month 1
dim(precip_Jan)
```

```
## [1] 720 360
```

```
  # 720, 360 which confirms this is one layer of data (January 1900)

# turns matrix into a df we can tidy
precip_df_Jan <- data.frame(precip_Jan)
colnames(precip_df_Jan) <- c(1:360)
lon_vals <- c(1:720)
```

```
precip_df_Jan$lon <- lon_vals

# tidy up data
precip_df_Jan <- gather(precip_df_Jan,"lat","precip",1:360)
precip_df_Jan$lat <- as.integer(precip_df_Jan$lat)
```

Plot data, Total Precip in January:

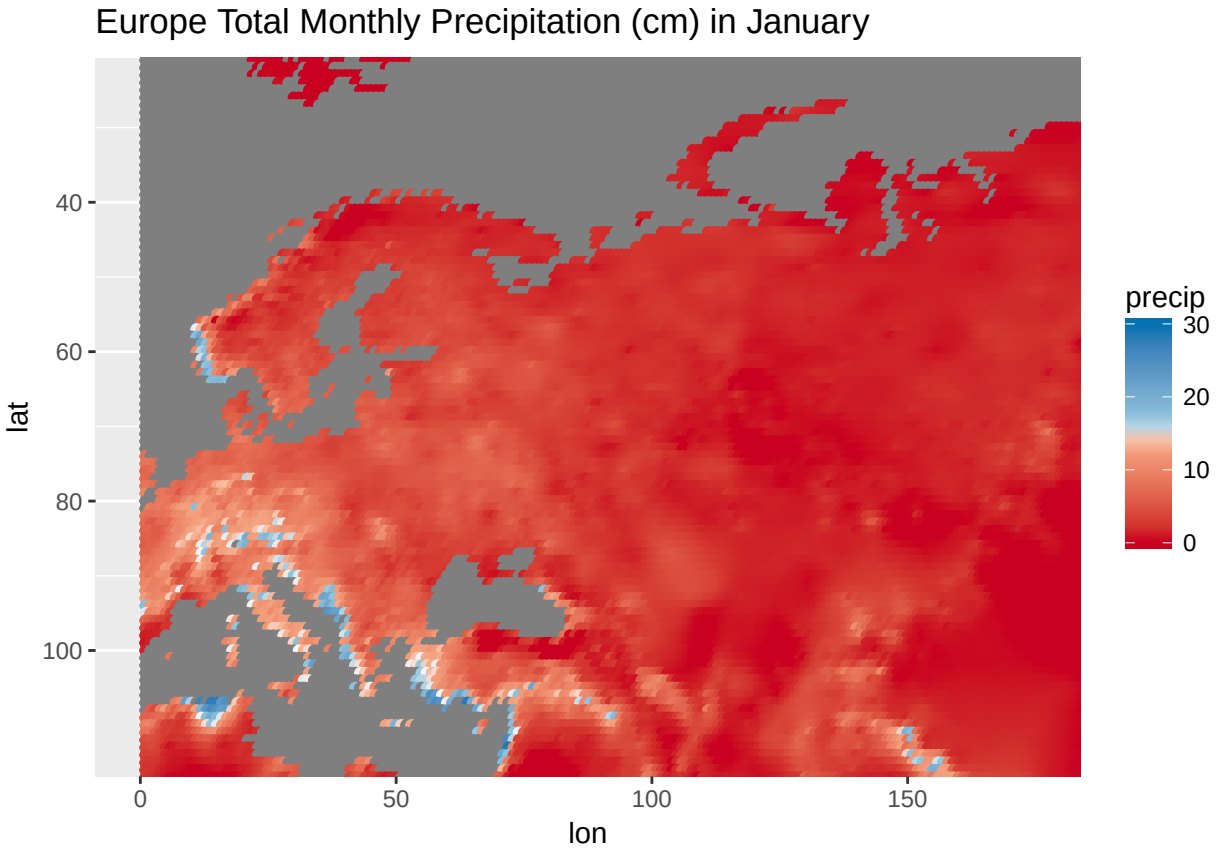
```
  # Download libraries to improve color options
library(RColorBrewer)
cols <- brewer.pal(n = 5, name = "RdBu")
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
ggplot() +
  geom_point(data = precip_df_Jan, aes(x=lon, y = lat, color=precip)) +
  coord_cartesian(xlim=c(0,175),ylim=c(25,112.5)) +
  scale_colour_gradientn(colours = cols,
                         values = rescale(c(-10, -1, 0, 1, 10)),
                         guide = "colorbar", limits=c(0, 30)) +
  scale_y_reverse() +
  ggtitle("Europe Total Monthly Precipitation (cm) in January")
```



Repeat process, Precip for July:

```
precip_Jul <- (precip_alltime[,,7]) # July will be month 7
dim(precip_Jul)
```

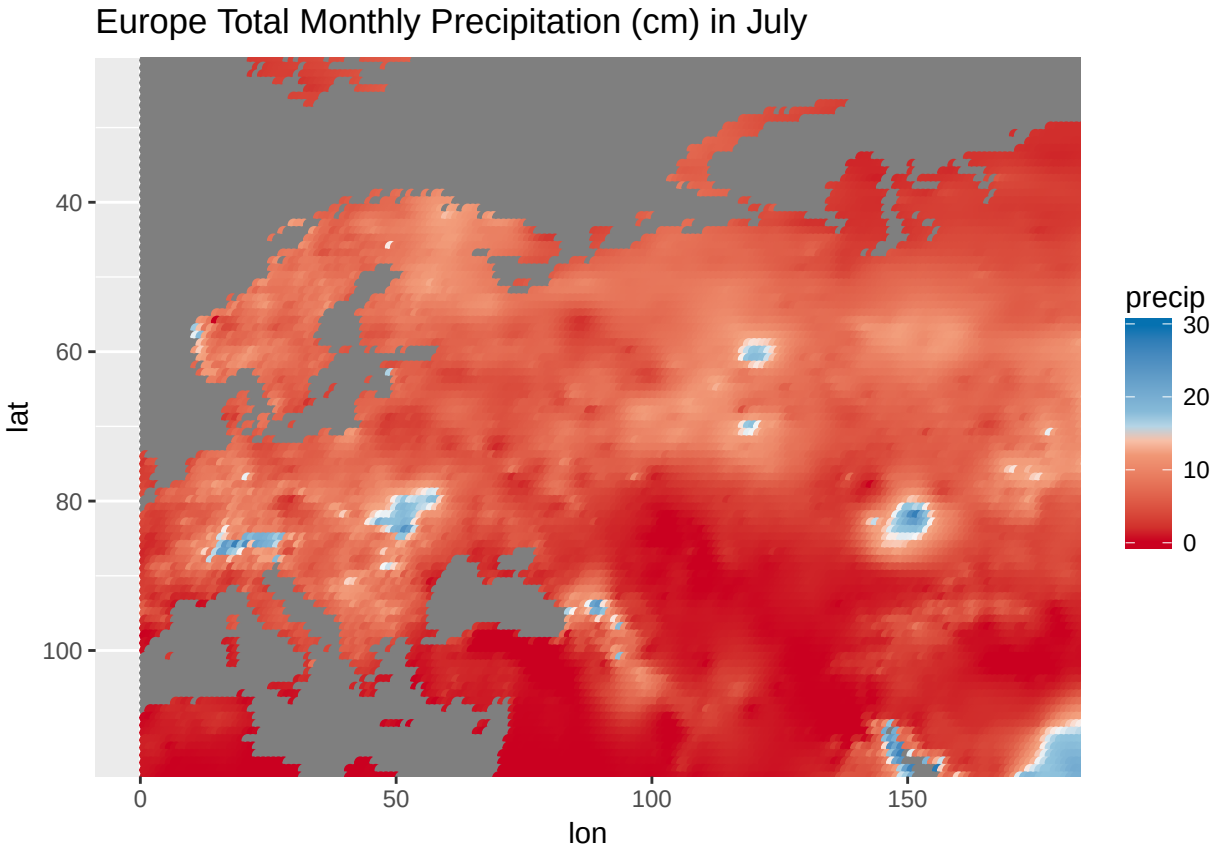
```
## [1] 720 360
```

```
  # 720, 360 which confirms this is one layer of data (July 1900)

# turns matrix into a df we can tidy
precip_df_Jul <- data.frame(precip_Jul)
colnames(precip_df_Jul) <- c(1:360)
lon_vals <- c(1:720)
precip_df_Jul$lon <- lon_vals

# tidy up data
precip_df_Jul <- gather(precip_df_Jul,"lat","precip",1:360)
precip_df_Jul$lat <- as.integer(precip_df_Jul$lat)

ggplot() +
  geom_point(data = precip_df_Jul, aes(x=lon, y = lat, color=precip)) +
  coord_cartesian(xlim=c(0,175),ylim=c(25,112.5)) +
  scale_colour_gradientn(colours = cols,
                         values = rescale(c(-10, -1, 0, 1, 10)),
                         guide = "colorbar", limits=c(0, 30)) +
  scale_y_reverse() +
  ggtitle("Europe Total Monthly Precipitation (cm) in July")
```



Observations:

- Mean temperatures across Europe are generally higher (lighter in color) in July compared to January, which makes sense given the timing of the Northern Hemisphere summer
- Mountain regions are relatively cooler (darker) than their surroundings, which makes sense given temperature generally decreases with elevation
- Monthly total precipitation varies more than temperature across Europe: in some places it rains more in January/the winter time (e.g. Italy, Greek coasts, western edge of Norway) and in other places it rains more in July/the summer time (e.g. spots of Russia, Romania/Ukraine, most of Scandinavia)