FOR/STT 875 Exercise 2

Part 1: Working with some spatial-temporal climate data

Today we're going to work with some air temperature data measured at 356 weather stations across the Northeastern United States between 1/1/2000 and 1/1/2005 drawn from the National Climatic Data Center (https://www.ncdc.noaa.gov/cdo-web) (NCDC). I aggregated the NCDC hourly temperature data to mean monthly values of degrees Celsius. Additionally, each station temperature record is associated with its latitude, longitude, and elevation (meters above sea level). As the semester progresses we'll make use of this set of data in a variety of ways. Today we'll perform some simple but insightful analyses using these data.

```
con <- url("http://blue.for.msu.edu/FOR875/data/NE-temp.RData")
load(con)
close(con)
rm(con)
ls()
```

```
## [1] "ne.station.1" "ne.station.2" "ne.stations"
```

Two useful R functions are `rm` and `ls`. The first removes an object (in this case, `con`) from the current R workspace. The second lists all the objects in the current R workspace. If you had created other objects in an earlier R session, they might be listed along with the objects read in from the external file.

Initially we're only interested in the temperature records for stations 1 and 2 that are in objects `ne.station.1` and `ne.station.2`, respectively. These objects are R vectors of numeric values. The code below uses the functions `is.vector` and `is.numeric` to check if I'm telling the truth.

```
is.vector(ne.station.1)
```

```
## [1] TRUE
```

```
is.numeric(ne.station.1)
```

```
## [1] TRUE
```

First let's look at the station 1 temperature record.

```
head(ne.station.1)
```

```
##   2000-01   2000-02   2000-03   2000-04   2000-05   2000-06
## -5.277778 -1.666667  4.833333  7.833333 14.277778 18.444444
```

```
mean(ne.station.1)
```

```
## [1] 8.905685
```

```
max(ne.station.1)
```

```
## [1] 22.72222
```

```
summary(ne.station.1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -8.8333  0.5556  9.1111  8.9057 17.5000 22.7222
```

The hash marks that show up in front of R output are useful if we want to copy and paste from a document into R, since R will treat the output as comments and won't get confused. But the hash marks are somewhat confusing and irritating in a written document, and we will use the option `comment = NA` to remove these in the next code chunk.

The `which.max` and `which.min` functions return the vector position (or index) where the maximum and minimum values occur, respectively. Below, I'm using `which.max` to return the vector position where the maximum temperature occurs and any name associated with that position (in this case the name is the date). Next, using this position to reference the `ne.station.1` vector value, returns the maximum temperature value (notice the use of the `[]` for referencing the value at the given position). This maximum temperature should, and does, match the temperature returned by `max(ne.station.1)` (although the `max` function strips off the name associated with the value).

```
which.max(ne.station.1)
```

```
2006-07
     79
```

```
ne.station.1[which.max(ne.station.1)]
```

```
 2006-07
22.72222
```

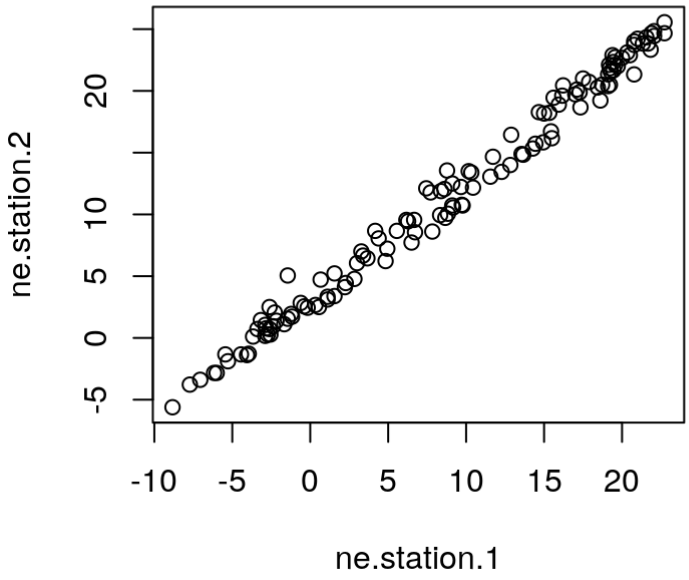
```
max(ne.station.1)
```

```
[1] 22.72222
```

Next we look at the relationship between mean monthly temperature collected at station 1 and 2. Specifically we'll calculate the correlation coefficient and associated scatter plot.

```
cor(ne.station.1, ne.station.2)
```

```
## [1] 0.9938151
```



Make sure your figure is the same dimension and centered in the document. I used `fig.width=4, fig.height=4, fig.align="center"` in the figure's chunk options.

Note, the horizontal rule (i.e., line) below. It is useful for delineating different parts of a document.

Part 2: Some questions for you to answer

Write R code that answers the following questions for the station 2 data, i.e., `ne.station.2` vector. Write your answers in an R code chunk immediately after each question.

1. How many temperature readings were taken at station 2?

```
# Question 1 answer here
```

2. What date was the:
    a. highest temperature recorded?

```
# Question 2a answer here
```

    b. lowest temperature recorded?

```
# Question 2b answer here
```

3. What was the mean temperature over the record?

```
# Question 3 answer here
```

Part 3: Some other useful R Markdown features

Here are some other useful features of R Markdown.

a. It is possible to display subscripts and superscripts such as x_2 or $5^3 = 125$.
b. It is possible to use LaTeX to display more complex mathematics such as $\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$ or $SD(\bar{X}) = \frac{\sigma}{\sqrt{n}}$.
    - However, if a document needs substantial mathematics or substantial control over formatting, using `knitr` along with LaTeX is an option. Another option is using the package `boookdown` that builds off of R Markdown to enable writing of larger more complex documents. This is how we created the course book.
c. As you saw in Part 1, URLs can be included in R Markdown, e.g., the webcomic xkcd (http://xkcd.com/) sometimes has funny stuff.

Part 4: Now just some fun and foreshadowing of topics to come

Try out the resulting web map by clicking on the station markers, radio buttons, and check boxes. Zoom into the coldest location, the Mount Washington Observatory (https://www.mountwashington.org/experience-the-weather/current-summit-conditions.aspx). It's a lovely view from there.

The `leaflet` package used to create this dynamic web map will be covered later in the course. Don't worry about understanding the code used to create this map. It is an example of how R Markdown can work with packages like `leaflet` to create dynamic, interactive documents like this HTML page. Please note, you will likley need to install the leaflet package via `install.packages("leaflet")`.

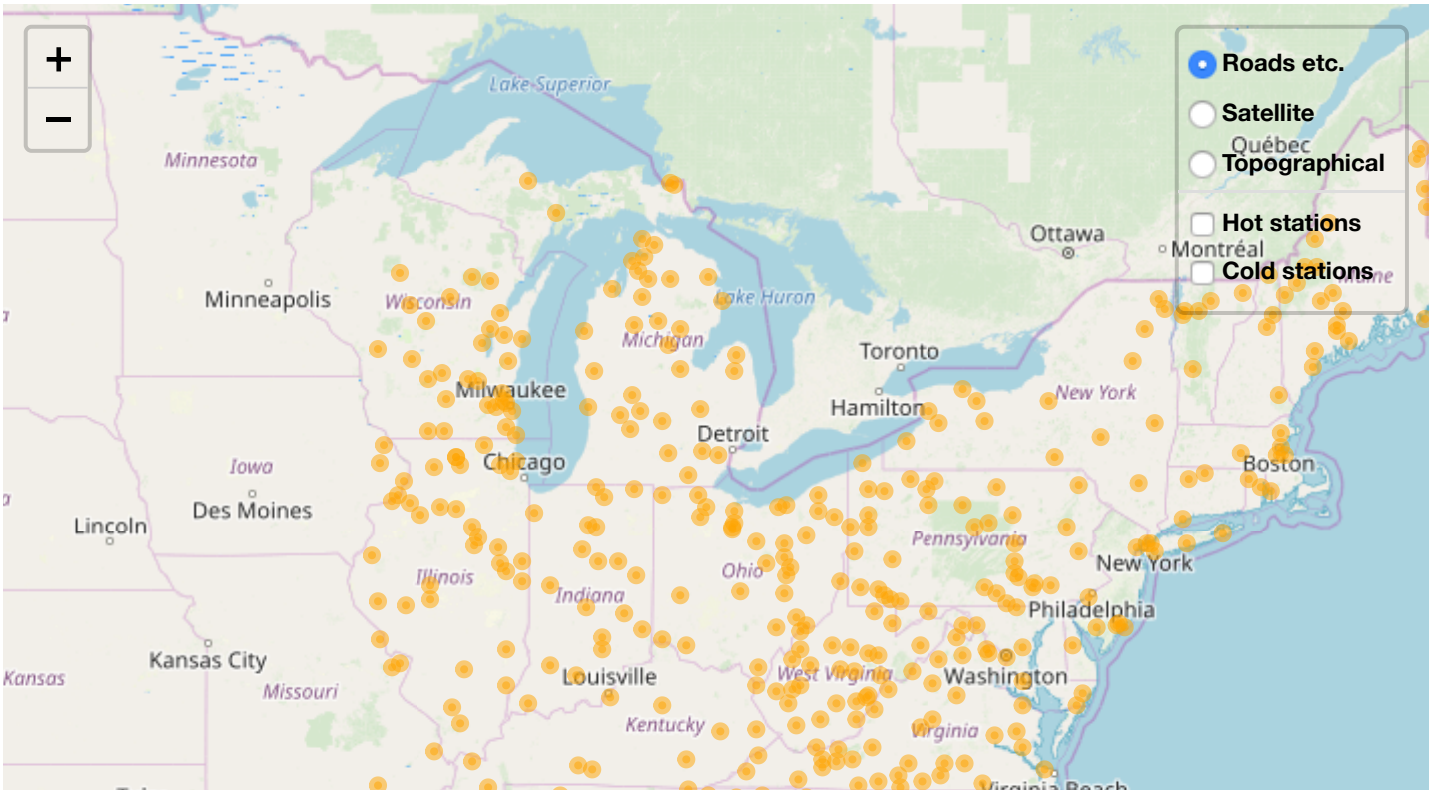
```
library(leaflet)

##find some really hot and cold stations
hot.stations <- subset(ne.stations, yr2010.m07 > 22 & elev >= 700)
cold.stations <- subset(ne.stations, yr2010.m01 < -12 & elev >= 700)

##make a pretty map with labels
all.popup <-  paste("Station ID: ",ne.stations[,"stationID"],"<br>July 2010 temp (C): ",
round(ne.stations[,"yr2010.m07"],2),sep="")
hot.popup <- paste("Station ID: ",hot.stations[,"stationID"],"<br>July 2010 temp (C): ",
round(hot.stations[,"yr2010.m07"],2),sep="")
cold.popup <- paste("Station ID: ",cold.stations[,"stationID"],"<br>January 2010 temp
 (C): ",round(cold.stations[,"yr2010.m01"],2),sep="")

m <- leaflet(ne.stations, width="100%")%>%
    addCircleMarkers(~lon, ~lat, radius=2, fillOpacity=0.5, color="orange", popup=all.po
pup)%>%
    addMarkers(~lon, ~lat, data=hot.stations, group="Hot stations", popup= ~hot.popup) %
>%
    addMarkers(~lon, ~lat, data=cold.stations, group="Cold stations", popup= ~cold.popu
p) %>%
    addTiles(group = "Roads etc.") %>%
    addProviderTiles("Esri.WorldImagery", group = "Satellite") %>%
    addProviderTiles("Thunderforest.Landscape", group = "Topographical") %>%
    addLayersControl(baseGroup = c("Roads etc.", "Satellite", "Topographical"),
                     overlayGroups = c("Hot stations", "Cold stations"),
                     options = layersControlOptions(collapsed = FALSE)
                     ) %>%
    hideGroup(c("Hot stations", "Cold stations"))

m
```




Tulsa
Oklahoma
Memphis
Arkansas
Tennessee
Charlotte
North Carolina
South Carolina
Atlanta
Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/)