R – Hubway Lab – Part II

This lab is about Hubway. It is the "Zipcar" of bikes in Boston. Here is some information about the history of the company from <https://www.thehubway.com/about>:

Soon after the Boston Bikes program was founded in 2007, Boston Mayor Thomas M. Menino and Director of Bicycle Programs, Nicole Freedman, decided to bring bike sharing to the Boston area.

The Hubway system launched July 28, 2011 with 600 bicycles and 60 stations throughout Boston. Following launch Hubway immediately surpassed expectations. Within 10 weeks Hubway bikes had logged more than 100,000 rides, and by the end November had more than 3,600 annual members. During its first season a strong community of users formed that took an active part in the success of the system. Users reported damage, returned lost keys, docked loose bikes and even delivered engagement rings for each other (true story).

Since then, the system has expanded to more than 160 stations with 1,600 shared bikes, with more to come this year. Hubway's 4 millionth and 5 millionth trips are expected to be taken in 2016.

A few years ago, Hubway had a data competition which involved sharing a dataset with more than 1.5 million rows of trip information – we've cut that dataset down to ~350,000 rows. Let's explore this dataset using R and see if we can uncover anything interesting.

**How do we begin? What information do we have?**

Two .csv files: hubway\_trips.csv and hubway\_stations.csv

**A. Merge the trip and station data**

**WALK THROUGH THE FIRST MERGE (INSTRUCTIONS BELOW) WITH THE CLASS, THEN DO THE SECOND MERGE ON YOUR OWN**

1. Use **merge** (twice) to append the appropriate station names to the trips data frame. Add a column for the starting station and the ending station. See ?merge.

Look at the summary of this new data frame. Are all of the stations "existing"? Why is that? Did you merge using your hubway station data frame with only existing stations? Or the data frame with all stations? **Remove all trips to or from a "removed" station if necessary.**

2. Use names() or colnames() to clean up the column names in the merged data frame.

3. Which station is most frequently used as a starting station?

South Station - Id: 22

4. Which station is most frequently used as an ending station?

South Station - 22

5. What is the name of the starting station with the longest average trip duration? Hint: tapply or aggregate

HMS / HSPH - Ave. Louis Pasteur at Longwood Ave.

6. What is the name of the starting station with the shortest average trip duration?

MIT at Mass Ave / Amherst St

7. What is the name of the ending stations with the longest and shortest average trip duration?

End Station with Largest Average: Dorchester Ave. at Gillette Park

End Station with Smallest Average: MIT at Mass Ave / Amherst St

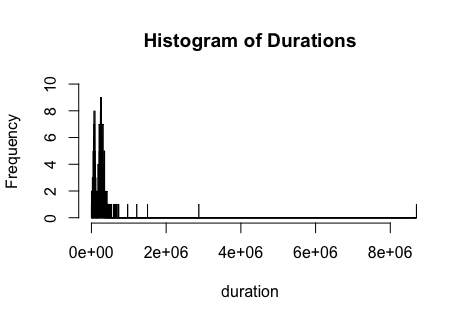
**CHECK IN WITH THE CLASS HERE, CONFIRM THE ABOVE ANSWERS**

8. Which user zipcode is associated most frequently with trips ending at the station with the longest average trip duration?

02127

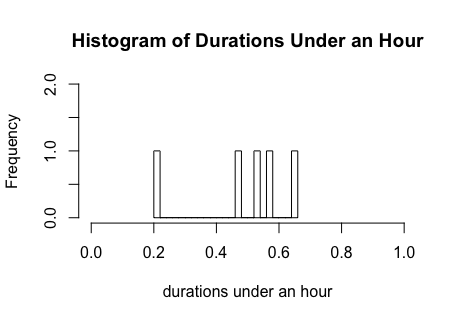
9. Create a table showing the number of trips by gender. Plot this table using **barplot**. The argument for **barplot** is the table itself. What percent of total trips were completed by males?

Male Percentage = 0.7439792

10. Use **hist** with the vector for trip duration as the only argument to plot the histogram of trip durations. What do you notice?

There are 2 humps in the data and a piece of data that lies way beyond most of the data.

11. Use **hist** with the vector for trip duration, considering only trips less than an hour to plot the histogram of trip durations under an hour.



**CHECK IN WITH THE CLASS HERE, CONFIRM THE ABOVE ANSWERS**

**B. Explore the station data on a map**

Consider the station data frame, with only existing stations.

Assume the MIT station, id = 67, is the center of the existing Hubway stations (so that we can plot stations on a map centered here)

**Optional challenge**: Don't assume id = 67 is the center of the existing Hubway stations. Look at the range of latitudes and longitudes for the existing Hubway stations. Pick the values in the center of these ranges, using this (longitude, latitude) as the "center" of the map.

**Optional AT HOME challenge (to practice coding)**: Approximate the "distance" between each station and the center as the hypotenuse of the triangle made up of the difference in longitude and latitude between each station and the center (not a bad approximation at this scale). Find the minimum value. Which station is closest to the center? Return the column information from the "station" and "id" columns.

1. Install the "ggmap" package and load it.

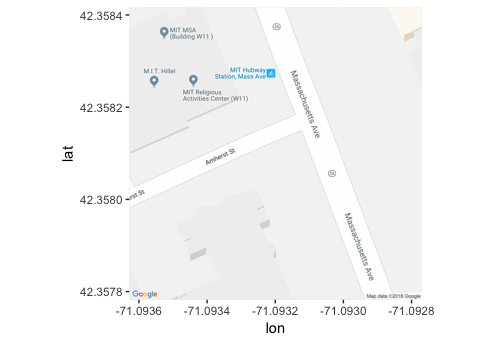
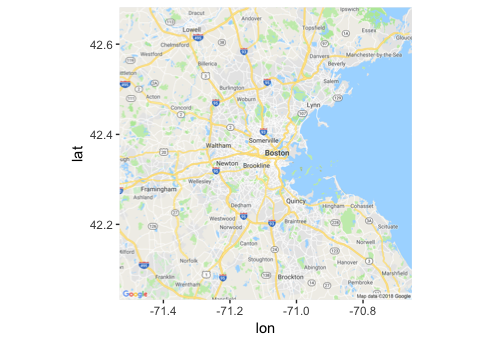
2. Let's get a map from Google maps. Read the help for get\_map to begin.

3. Create a variable to save the map to and assign it with a call to get\_map, specifying the following arguments: location, color, maptype and zoom. Also specify: source = "google". Start with a zoom of 10.

4. Plot the map using:

ggmap(YOURVARIABLE, extent = "device")

Do you see the map?



**CHECK IN WITH NEIGHBOR IF YOU DON'T SEE THE MAP TO GET HELP**

5. Try changing extent back to "normal." Choose which you like best, if you want to label the axes, do so with xlab and ylab arguments.

labels do not change

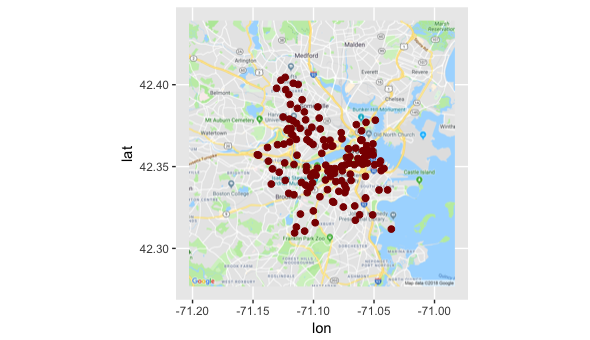
6. Think of ggmap like ggplot, in that you can plot over this map using the addition of geometric objects. Let's use geom\_point(). Try adding geom\_point() to ggmap, specifying the data as the existing bike stations data.

**Hint**: The longitude and latitude may have been read in as factors, and we want them to be numbers. Check if these fields are factors, if so, try plotting them as factors. Note the error you get. Try coercing and saving in a new vector with as.numeric(). Inspect this vector—what is going on here? Try again with as.numeric(as.character())

**Hint**: For geom\_point(): specify x and y within aes(), sample syntax is shown here:

**ggmap(mapImageData, extent = "device") + geom\_point(data = df, aes(x = col1, y = col2), size = 2, color = "darkred")**

7. What do you think of the plot? Is it easy to read? Go back and adjust the zoom in your call to ggmap and find the best option.

zoom = 12:

8. Now is the plot easy to read? Do you want to change the maptype? What about the size of the points? What about their color? Go back and adjust these settings until you have a map you are happy with.

**PAUSE HERE AND CONVERSE WITH YOUR NEIGHBOR AND THE REST OF THE CLASS** – what do you like about each other's maps?

9. Make any final changes and save your map to a Word document for submission. Here is a useful command for saving:

ggsave("FILENAME.png", plot = last\_plot(), width = widthset, height = heightset, units = "in", dpi = dpiset)

We're off to a good start with understanding the Hubway data, we will revisit it in a later lab.

