ANA 515 Assignment 4

William Halim

2022-10-16

#1 Business problem and goal In 2015, the Mayor of NYC claimed that Uber was creating an additional traffic and congestion in Manhattan. To evaluate the validity of the argument, the city did a study of Uber’s trip data to determine the number of rides per day, time and date of the rides, trips per month, etc. Eventually, heat maps were created using the data to see where Uber rides were picking up its riders in NYC. This information was used in various articles published by FiveThirtyEight.

#2 Data retreival and information I got the data of Uber Pickups Dataset from Data Flair. However, the data was originally obtained by FiveThirtyEight on July 20, 2015 through the submittal of a Freedom of Information Law request to the NYC Taxi and Limousine Commission (TLC). The dataset is composed of information from 4.5 million Uber pickups in New York City from April 2014 to September 2014. In addition, the dataset includes data from 14.3 million Uber pickups from January 2015 to June 2015. All in all, the data is split to 6 different CSV files. Each file has four columns – Data and Time, Latitude, Longitude, and Base of the pickup. The Base is the TLC base company code affiliated with the Uber pickup.

#3 Importing data and installing packages To start the project, I installed the following R packages: ggplot2, dplyr, tidyr, ggthemes, scales, lubridate, and DT. Next, I imported all of the data sets as CSV files into R through the working directory. Then, I created the vectors of the colors that I will use in my plots.

setwd("~/Downloads/archive/Uber Dataset")  
Apr\_data <- read.csv("uber-raw-data-apr14.csv")  
May\_data <- read.csv("uber-raw-data-may14.csv")  
Jun\_data <- read.csv("uber-raw-data-jun14.csv")  
Jul\_data <- read.csv("uber-raw-data-jul14.csv")  
Aug\_data <- read.csv("uber-raw-data-aug14.csv")  
Sep\_data <- read.csv("uber-raw-data-sep14.csv")  
library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(ggthemes)  
library(scales)  
library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(DT)  
colors = c("#CC1011", "#665555", "#05a399", "#cfcaca", "#f5e840", "#0683c9", "#e075b0")

#4 Describing the data set The original data sets have 4 columns each as described earlier. It consists of: Date.Time, Lat, Lon, and Base. In addition, the number of rows in each data set varies from 564516 (April) to 1028136 (September). The summary function shows the min, median, mean, and max of each latitude and longitude variable of each data set.

colnames(Apr\_data)

## [1] "Date.Time" "Lat" "Lon" "Base"

summary(Apr\_data)

## Date.Time Lat Lon Base   
## Length:564516 Min. :40.07 Min. :-74.77 Length:564516   
## Class :character 1st Qu.:40.72 1st Qu.:-74.00 Class :character   
## Mode :character Median :40.74 Median :-73.98 Mode :character   
## Mean :40.74 Mean :-73.98   
## 3rd Qu.:40.76 3rd Qu.:-73.97   
## Max. :42.12 Max. :-72.07

summary(May\_data)

## Date.Time Lat Lon Base   
## Length:652435 Min. :40.11 Min. :-74.93 Length:652435   
## Class :character 1st Qu.:40.72 1st Qu.:-74.00 Class :character   
## Mode :character Median :40.74 Median :-73.98 Mode :character   
## Mean :40.74 Mean :-73.97   
## 3rd Qu.:40.76 3rd Qu.:-73.97   
## Max. :41.32 Max. :-72.18

summary(Jun\_data)

## Date.Time Lat Lon Base   
## Length:663844 Min. :39.96 Min. :-74.86 Length:663844   
## Class :character 1st Qu.:40.72 1st Qu.:-74.00 Class :character   
## Mode :character Median :40.74 Median :-73.98 Mode :character   
## Mean :40.74 Mean :-73.97   
## 3rd Qu.:40.76 3rd Qu.:-73.97   
## Max. :41.32 Max. :-72.70

summary(Jul\_data)

## Date.Time Lat Lon Base   
## Length:796121 Min. :39.72 Min. :-74.83 Length:796121   
## Class :character 1st Qu.:40.72 1st Qu.:-74.00 Class :character   
## Mode :character Median :40.74 Median :-73.98 Mode :character   
## Mean :40.74 Mean :-73.97   
## 3rd Qu.:40.76 3rd Qu.:-73.97   
## Max. :41.34 Max. :-72.31

summary(Aug\_data)

## Date.Time Lat Lon Base   
## Length:829275 Min. :39.66 Min. :-74.77 Length:829275   
## Class :character 1st Qu.:40.72 1st Qu.:-74.00 Class :character   
## Mode :character Median :40.74 Median :-73.98 Mode :character   
## Mean :40.74 Mean :-73.97   
## 3rd Qu.:40.76 3rd Qu.:-73.96   
## Max. :41.32 Max. :-72.34

summary(Sep\_data)

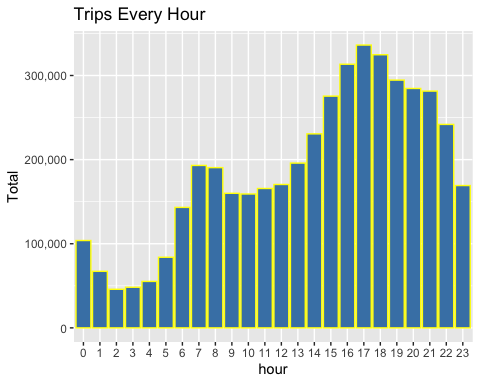
## Date.Time Lat Lon Base   
## Length:1028136 Min. :39.99 Min. :-74.77 Length:1028136   
## Class :character 1st Qu.:40.72 1st Qu.:-74.00 Class :character   
## Mode :character Median :40.74 Median :-73.98 Mode :character   
## Mean :40.74 Mean :-73.97   
## 3rd Qu.:40.76 3rd Qu.:-73.96   
## Max. :41.35 Max. :-72.72

#5 Data prep and errors In this step, I combined all of the month data sets into one 2014 file and added columns for the day, month, year, day of the week, hour, minute, and second. This will help graph and plot all data in different data visualizations. No errors were found with the data sets.

Data\_2014 <- rbind(Apr\_data, May\_data, Jun\_data, Jul\_data, Aug\_data, Sep\_data)  
Data\_2014$Date.Time <- as.POSIXct(Data\_2014$Date.Time, format = "%m/%d/%Y %H:%M:%S")  
Data\_2014$Time <- format(as.POSIXct(Data\_2014$Date.Time, format = "%m/%d/%Y %H:%M:%S"), format="%H:%M:%S")  
Data\_2014$Date.Time <- ymd\_hms(Data\_2014$Date.Time)  
Data\_2014$day <- factor(day(Data\_2014$Date.Time))  
Data\_2014$month <- factor(month(Data\_2014$Date.Time, label = TRUE))  
Data\_2014$year <- factor(year(Data\_2014$Date.Time))  
Data\_2014$dayofweek <- factor(wday(Data\_2014$Date.Time, label = TRUE))  
Data\_2014$hour <- factor(hour(hms(Data\_2014$Time)))  
Data\_2014$minute <- factor(minute(hms(Data\_2014$Time)))  
Data\_2014$second <- factor(second(hms(Data\_2014$Time)))

#6 Modeling and plots The first plot shows the total number of Uber trips per hour. Based on the bar graph, 5:00 PM is the most popular time to take an Uber. 4:00 PM and 3:00 PM are close behind with the number of trips. This data captures the evening rush hour home from work. However, this data was captured pre-pandemic, so I believe that it looks very different in 2020 and 2021.

hour\_data <- Data\_2014 %>%  
 group\_by(hour) %>%  
 dplyr::summarize(Total = n())   
ggplot(hour\_data, aes(hour, Total)) +   
 geom\_bar( stat = "identity", fill = "steelblue", color = "yellow") +  
 ggtitle("Trips Every Hour") +  
 theme(legend.position = "none") +  
 scale\_y\_continuous(labels = comma)

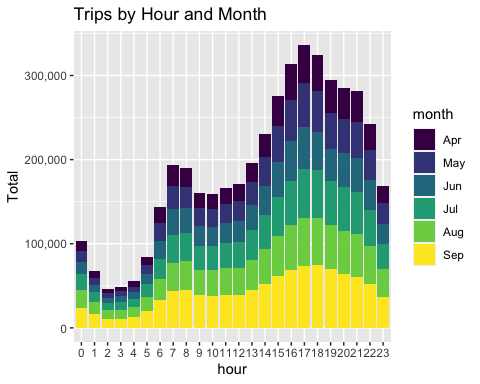


The second plot is similar to the first since it still shows the number of trips per hour. However, it breaks up each bar into the number of trips per month during that time. Based on the results, September has more trips at 5:00 PM compared to the other months. In addition, based on the visualization, September looks to have more trips at each hour compared to the other months provided by the data.

month\_hour <- Data\_2014 %>%  
 group\_by(month, hour) %>%  
 dplyr::summarize(Total = n())

## `summarise()` has grouped output by 'month'. You can override using the  
## `.groups` argument.

ggplot(month\_hour, aes(hour, Total, fill = month)) +   
 geom\_bar( stat = "identity") +  
 ggtitle("Trips by Hour and Month") +  
 scale\_y\_continuous(labels = comma)



#9 Conclusion The R script provided by Data Flair has many other data visualizations with the associated code. However, to reduce the time needed to knit, I only provided these two plots. The plots I included with my code show that most of the Uber trips are taken during the evening rush hour portion of the day in September.