Uber PickUp Data

Module 04 Assignment 4

Uber Pick Up Data

by Karen Suckling

Business Problem and Goal

In 2015, the Mayor of New York City claimed that Uber was creating 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 including: Uber Is Serving New York's Outer Boroughs More Than Taxis Are, Public Transit Should Be Uber's New Best Friend, Uber Is Taking Millions Of Manhattan Rides Away From Taxis, and Is Uber Making NYC Rush-Hour Traffic Worse?.

Data Retrieval and Information

I downloaded the 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. The data is split between 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.

Importing Data and Installing Packages

To start my project, I had to install the following R packages: ggplot2, ggthemes, lubridate, dplyr, tidyr, DT, and scales. Next, I imported all of the data sets as CSV files into R through my working directory. In addition, I created the vectors of the colors that I will use in my plots.

```
setwd("~/R/Assignment 3/Uber Data/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)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.0.5
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.5
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## 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)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
library(DT)
library(scales)



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

Describing Data Set

The original data sets have 4 columns each as I previously described, Date.Time, Lat, Lon, and Base. In addition, the number of rows in each data set varies from 564516 in apr_data to 1028136 in sep_data. 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(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(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(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(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(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
```

Data Preparation and Errors

Next, 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 of the data in different data visualizations. I did not find any errors 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)))
```

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. This data was captured pre-pandemic, so I am sure it looks much different in 2020 and 2021.

```
hour_data <- data_2014 %>%
  group_by(hour) %>%
  dplyr::summarize(Total = n())
datatable(hour_data)
```

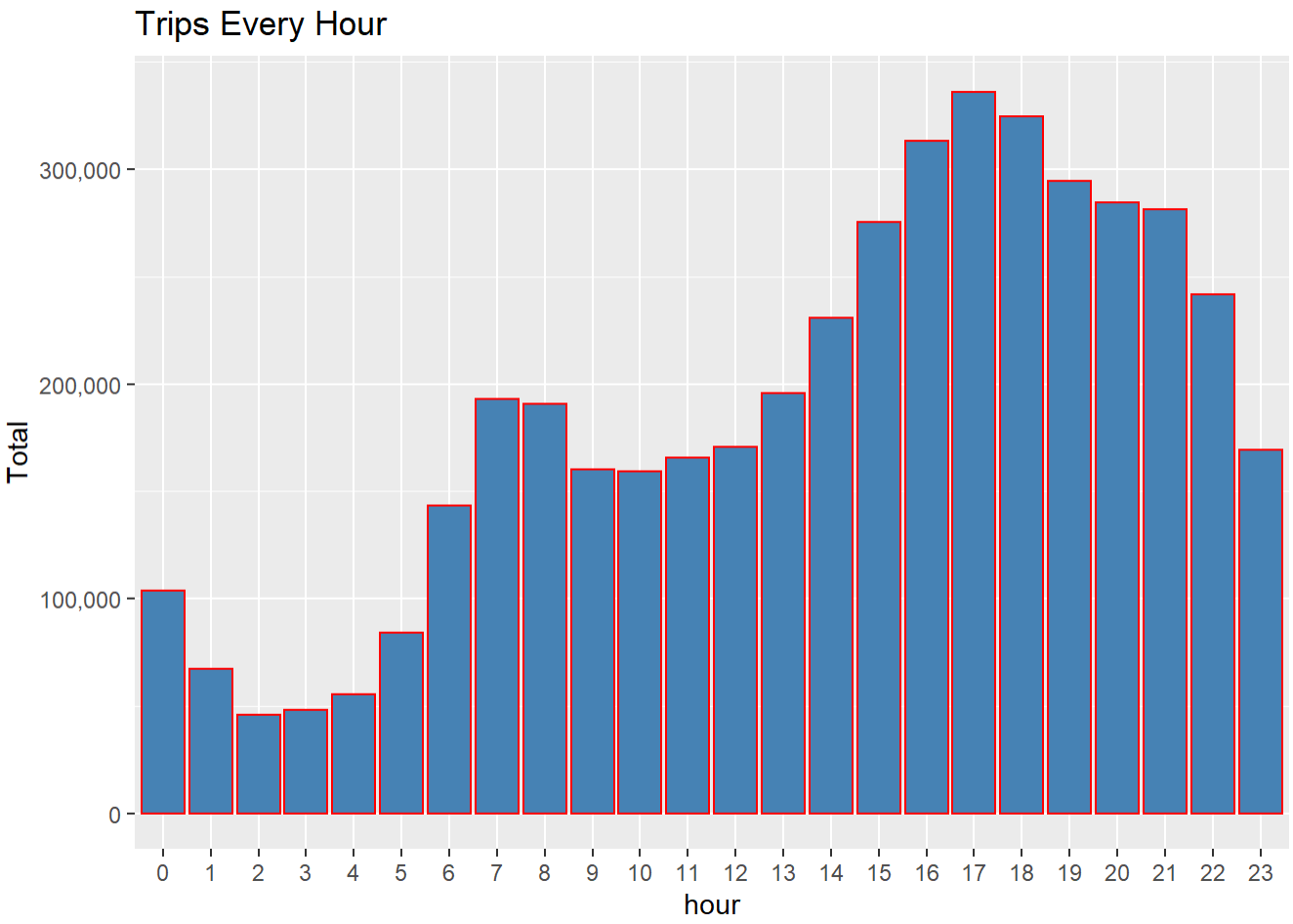
Show 10 **entries** **Search:**

	hour	Total
1	0	103836
2	1	67227
3	2	45865
4	3	48287
5	4	55230
6	5	83939
7	6	143213
8	7	193094
9	8	190504
10	9	159967

Showing 1 to 10 of 24 entries

Previous 1 2 3 Next

```
ggplot(hour_data, aes(hour, Total)) +
  geom_bar( stat = "identity", fill = "steelblue", color = "red") +
  ggtitle("Trips Every Hour") +
  theme(legend.position = "none") +
  scale_y_continuous(labels = comma)
```

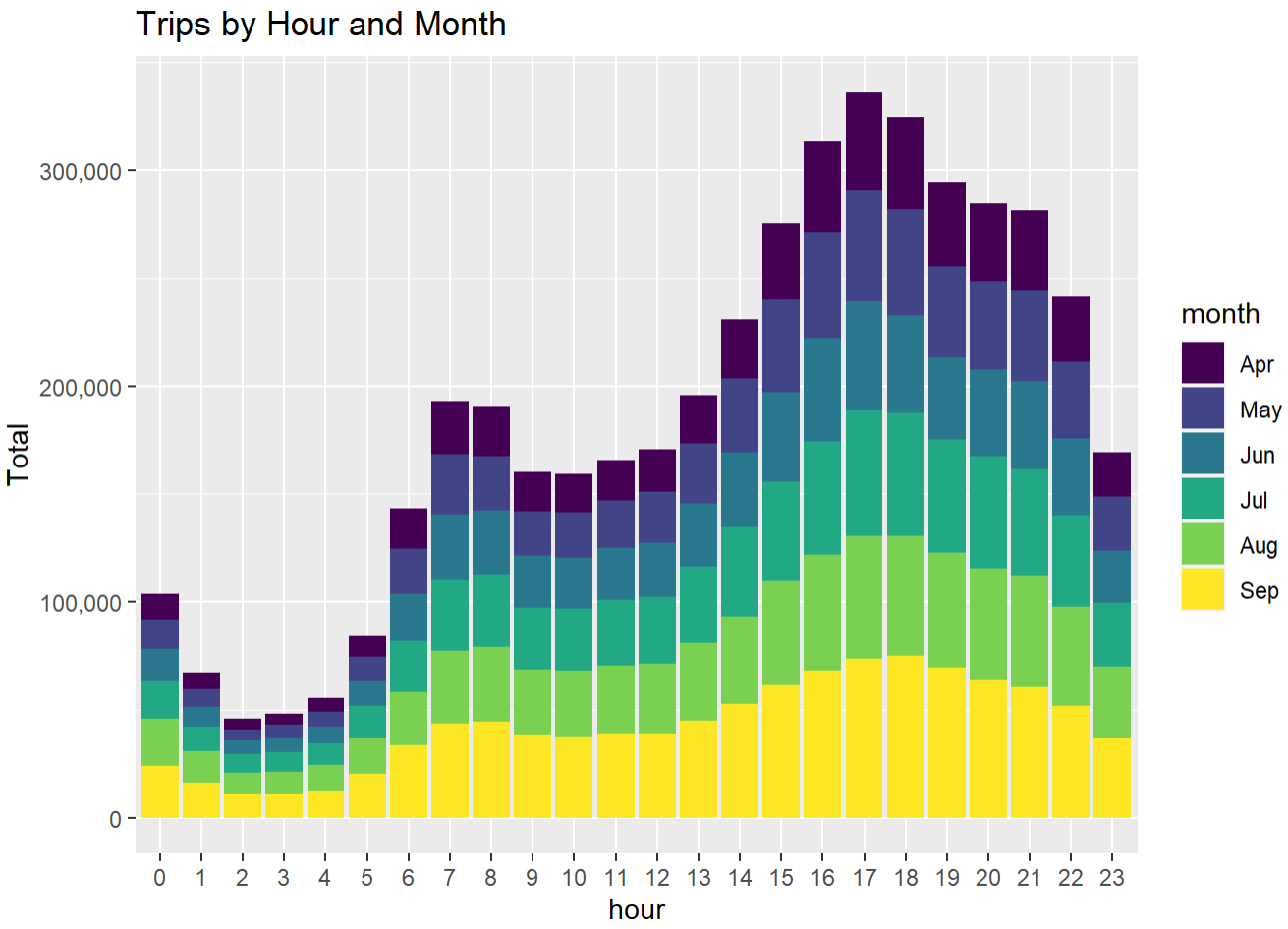


The second plot is similar to the first place, since it shows the number of trips per hour still. However, it breaks up each bar into the number of trips per month at that time. Based on the results, September has more trips at 5:00 PM compared to the other months. In addition, based on the visualization,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



Conclusion

The R script provided by Data Flair has many more data visualizations with the associated code. However, to reduce the time needed to knit, I only provided two of the 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.