Uber Analysis Project

Lunhan Zhang

2022-10-15

Uber Analysis with different Visualization in R

1. Discuss the business problem/goal (5 points) The dataset has information of about 4.5 million uber pickups in New York City from April 2014 to September 2014 and 14million more from January 2015 to June 2015. Users can perform data analysis and gather insights from the data.
2. identify where the dataset was retrieved from (2 points) The data set called Uber Rides in NYC in 2014. It was created by FiveThirtyEight, one of the few organizations to have acquired valuable trip data from Uber. This data is available to download on FiveThirtyEight's Kaggle Page (https://www.kaggle.com/datasets/fivethirtyeight/fivethirtyeight).
3. identify the code that imported and saved your dataset in R (3 points)

Installing Packages And Prep

```
library(tidyverse)
library(ggplot2)
library(dplyr)
library(tidyr)
library(lubridate)
library(ggthemes)
library(DT)
library(sf)
library(scales)
```

The next step is to read the Uber raw data files and assign them to make it easier for our next steps.

```
apr_raw <- read.csv("C:/Users/lunha/Downloads/uber-raw-data-apr14.csv")
may_raw <- read.csv("C:/Users/lunha/Downloads/uber-raw-data-may14.csv")
jun_raw <- read.csv("C:/Users/lunha/Downloads/uber-raw-data-jun14.csv")
jul_raw <- read.csv("C:/Users/lunha/Downloads/uber-raw-data-jul14.csv")
aug_raw <- read.csv("C:/Users/lunha/Downloads/uber-raw-data-aug14.csv")
sep_raw <- read.csv("C:/Users/lunha/Downloads/uber-raw-data-sep14.csv")
```

We now need to combine the data

```
uber_data <- rbind(apr_raw, may_raw, jun_raw, jul_raw, aug_raw, sep_raw)
#To check if this worked we run this
cat("The dimenions the data contains:", dim(uber_data))
```

```
## The dimenions the data contains: 4534327 4
```

Now let us read these data by showing the top 6 rows.

```
head(uber_data)
```

```
##          Date.Time     Lat      Lon   Base
## 1 4/1/2014 0:11:00 40.7690 -73.9549 B02512
## 2 4/1/2014 0:17:00 40.7267 -74.0345 B02512
## 3 4/1/2014 0:21:00 40.7316 -73.9873 B02512
## 4 4/1/2014 0:28:00 40.7588 -73.9776 B02512
## 5 4/1/2014 0:33:00 40.7594 -73.9722 B02512
## 6 4/1/2014 0:33:00 40.7383 -74.0403 B02512
```

We notice we have 4 columns with Date.Time and Base are factors and Lat and long are 2 doubles. We also notice that our Date, and time are quite confusing to read. We will now change the format of the day, month, year, and time, and put them in a column of their own to make it easier to read and conduct analysis.

```
# First we create the formats to M, D, Y
uber_data$Date.Time <- as.POSIXct(uber_data$Date.Time,
                                  format = "%m/%d/%Y %H:%M:%S")
uber_data$Time <- format(as.POSIXct(uber_data$Date.Time,
                                    format = "%m/%d/%Y %H:%M:%S"),
                         format="%H:%M:%S")
uber_data$Date.Time <- ymd_hms(uber_data$Date.Time)

# Next we create individual columns for each
uber_data$Day <- factor(day(uber_data$Date.Time))
uber_data$Month <- factor(month(uber_data$Date.Time, label = TRUE))
uber_data$Year <- factor(year(uber_data$Date.Time))
uber_data$Hour <- factor(hour(hms(uber_data$Time)))
uber_data$Minute <- factor(minute(hms(uber_data$Time)))
uber_data$Second <- factor(second(hms(uber_data$Time)))

# We can also add what day of the week it is
uber_data$Day_of_week <- factor(wday(uber_data$Date.Time, label = TRUE))
```

4. describe your data set (using the common attributes such as #rows, #columns, variable names, types, means, SD, min/max, NAs, etc…) (10 points)

```
# Lets look at the top 6 rows now.
head(uber_data)
```

```
##             Date.Time     Lat      Lon   Base     Time Day Month Year Hour
## 1 2014-04-01 00:11:00 40.7690 -73.9549 B02512 00:11:00   1   Apr 2014    0
## 2 2014-04-01 00:17:00 40.7267 -74.0345 B02512 00:17:00   1   Apr 2014    0
## 3 2014-04-01 00:21:00 40.7316 -73.9873 B02512 00:21:00   1   Apr 2014    0
## 4 2014-04-01 00:28:00 40.7588 -73.9776 B02512 00:28:00   1   Apr 2014    0
## 5 2014-04-01 00:33:00 40.7594 -73.9722 B02512 00:33:00   1   Apr 2014    0
## 6 2014-04-01 00:33:00 40.7383 -74.0403 B02512 00:33:00   1   Apr 2014    0
##   Minute Second Day_of_week
## 1     11      0         Tue
## 2     17      0         Tue
## 3     21      0         Tue
## 4     28      0         Tue
## 5     33      0         Tue
## 6     33      0         Tue
```

```
summary(uber_data)
```

```
##    Date.Time                          Lat             Lon
##  Min.   :2014-04-01 00:00:00.00   Min.   :39.66   Min.   :-74.93
##  1st Qu.:2014-05-28 15:18:00.00   1st Qu.:40.72   1st Qu.:-74.00
##  Median :2014-07-17 14:45:00.00   Median :40.74   Median :-73.98
##  Mean   :2014-07-11 18:50:50.57   Mean   :40.74   Mean   :-73.97
##  3rd Qu.:2014-08-27 21:55:00.00   3rd Qu.:40.76   3rd Qu.:-73.97
##  Max.   :2014-09-30 22:59:00.00   Max.   :42.12   Max.   :-72.07
##
##      Base               Time                Day          Month
##  Length:4534327     Length:4534327     30     : 167160   Apr: 564516
##  Class :character   Class :character   12     : 160606   May: 652435
##  Mode  :character   Mode  :character   16     : 158921   Jun: 663844
##                                        13     : 156892   Jul: 796121
##                                        23     : 156032   Aug: 829275
##                                        9      : 155135   Sep:1028136
##                                        (Other):3579581
##    Year              Hour             Minute        Second      Day_of_week
##  2014:4534327   17     : 336190   10     :  77757   0:4534327   Sun:490180
##                 18     : 324679   14     :  77161               Mon:541472
##                 16     : 313400   15     :  77124               Tue:663789
##                 19     : 294513   13     :  76957               Wed:696488
##                 20     : 284604   12     :  76849               Thu:755145
##                 21     : 281460   8      :  76719               Fri:741139
##                 (Other):2699481   (Other):4071760               Sat:646114
```

Analysis

5. discuss any data preparation, missing values and errors (10 points) (if the dataset was clean and there is no prep in the code, include a comment that explains what likely data preparation was done. What are the common issues with raw data?)

transfered date column to a proper format. THen we group them by category and also we gonna pick tge peak time fir uber rides a day data. Usually, the raw data should be cleanned by filtered some outliers or wrong data. outliers can be detected by cookD or any other ways,

Uber Trips Per Hour

```
# To do that let us first create a new table called trips_per_hour
#To do that let us first create a new table called trips_per_hour
trips_per_hour <-
  uber_data %>%
  group_by(Hour) %>%
  summarize(Total = n())
datatable(trips_per_hour)
```

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

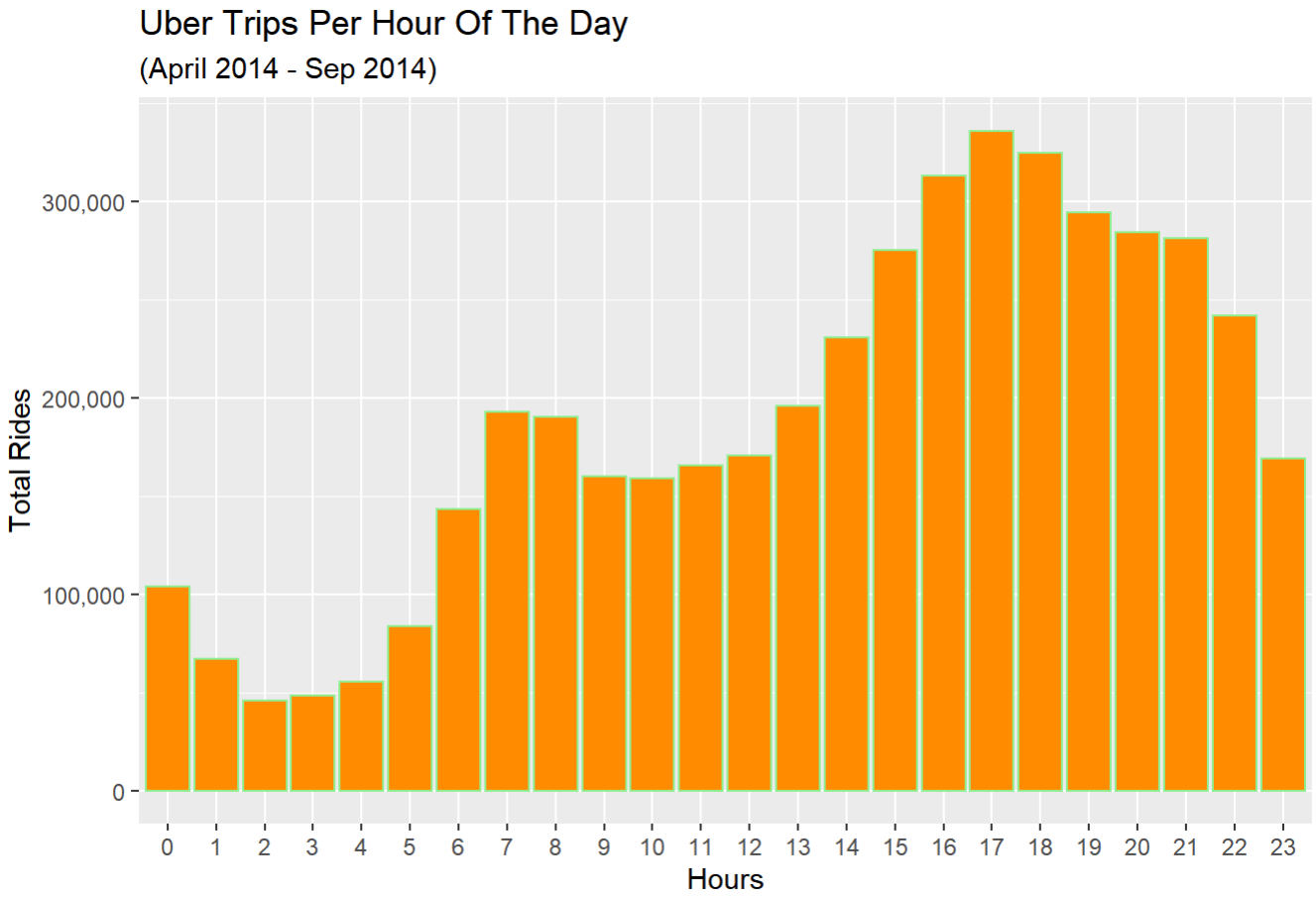
```
# Lets have a look at it in descending order to see the peak times

arrange(trips_per_hour, - Total)
```

```
## # A tibble: 24 × 2
##    Hour   Total
##    <fct>  <int>
##  1 17    336190
##  2 18    324679
##  3 16    313400
##  4 19    294513
##  5 20    284604
##  6 21    281460
##  7 15    275466
##  8 22    241858
##  9 14    230625
## 10 13    195877
## # … with 14 more rows
```

This gives us an idea of Uber rides peak times. Let us plot it into a graph to look at it even better.

```
ggplot(trips_per_hour, aes(Hour, Total)) +
  geom_bar(stat = "identity", fill="DarkOrange", color="LightGreen") +
  labs(
    title = "Uber Trips Per Hour Of The Day",
    subtitle = "(April 2014 - Sep 2014)",
    caption = "Data from Uber Rides in NYC dataset",
    x = "Hours",
    y = "Total Rides"
  ) +
  scale_y_continuous(labels = comma)
```



We can make our **First Hypothesis**. **H1**: **Peak times for Uber Rides in NYC during 2014 were mostly in the evening with 5 PM being the busiest.**

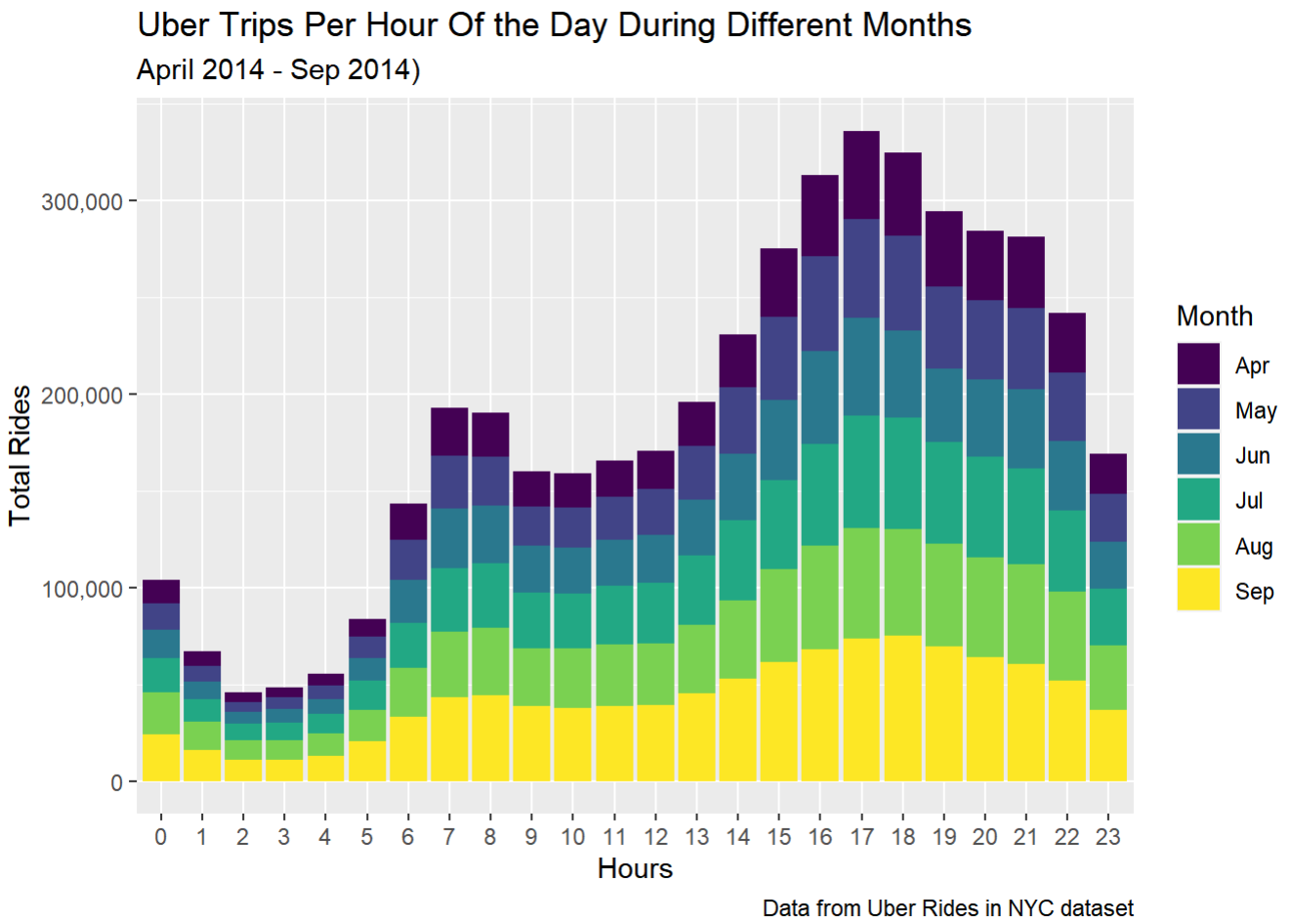
Uber Trips Per Hour with Months

Let us add months to the table as well, to see how the months affect the data.

```
Month_Hour <-
  uber_data %>%
  group_by(Month, Hour) %>%
  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")+
  labs(
    title = "Uber Trips Per Hour Of the Day During Different Months",
    subtitle = "April 2014 - Sep 2014)",
    caption = "Data from Uber Rides in NYC dataset",
    x = "Hours",
    y = "Total Rides"
  ) +
  scale_y_continuous(labels = comma)
```



6. discuss the modeling (10 points)

When we add months to the graph, we can say that the month to contribute the greatest number of trips at 5 pm was on September. We can further this research to understand why that was the case.

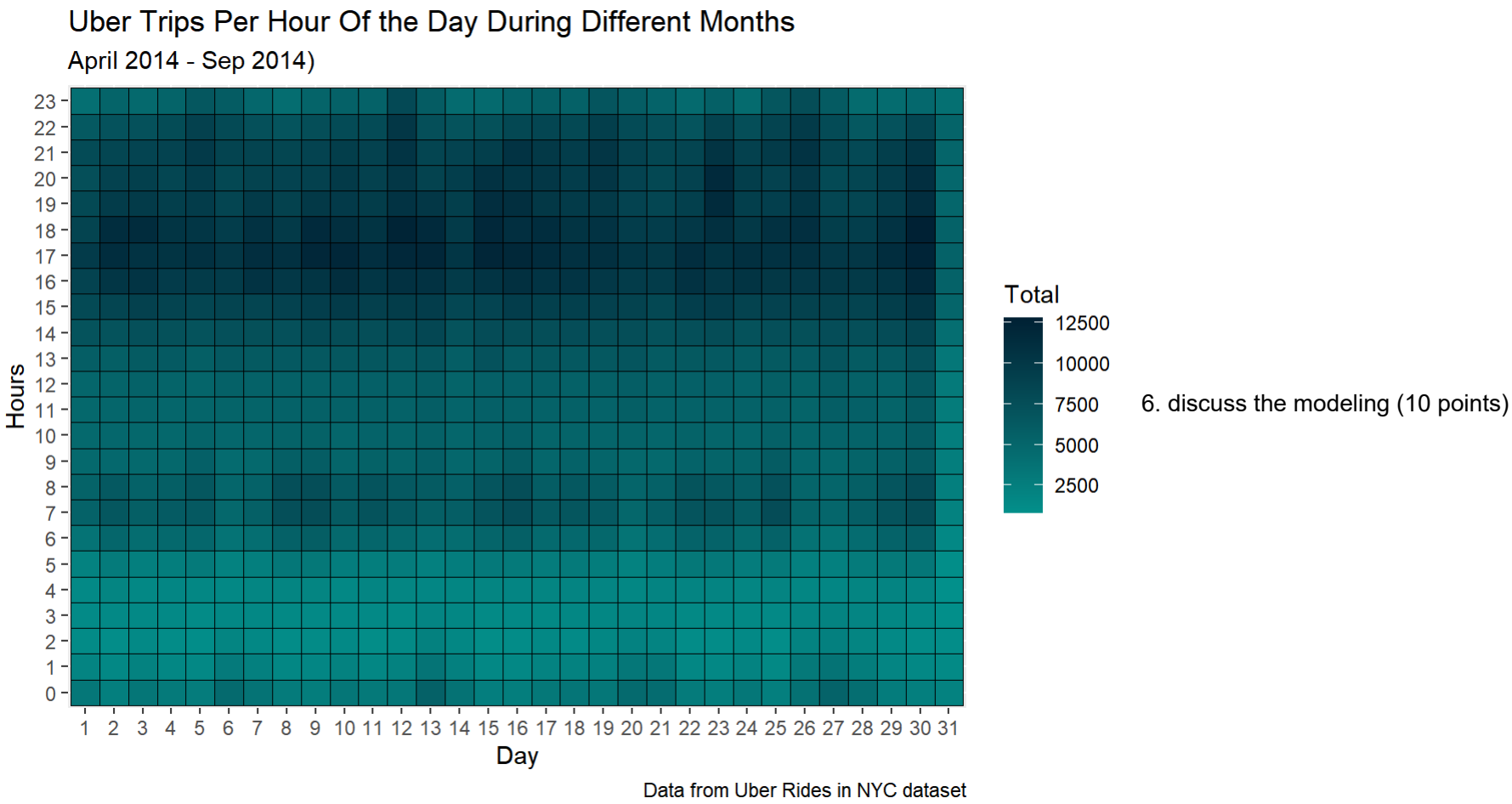
Heat Map

Moving on, let us back it up with a heat map to make sure our hypothesis is correct.

```
# Heat map by the Hours and Days
Day_Hour <-
  uber_data %>%
  group_by(Day, Hour) %>%
  summarize(Total = n())
```

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

```
ggplot(Day_Hour, aes(Day, Hour, fill = Total)) +
  geom_tile(color = "Black") +
  scale_fill_gradient(low = "#02908b",
                      high = "#002134",
                      guide = "colorbar") +
  labs(
    title = "Uber Trips Per Hour Of the Day During Different Months",
    subtitle = "April 2014 - Sep 2014)",
    caption = "Data from Uber Rides in NYC dataset",
    x = "Day",
    y = "Hours"
  )
```



We can conclude that on all the days of the month the peak time for Uber rides starts from 3 pm onwards. With this information, Uber can make sure there are enough drivers during those times to cater to all their customers.

Peak Day Of The Week

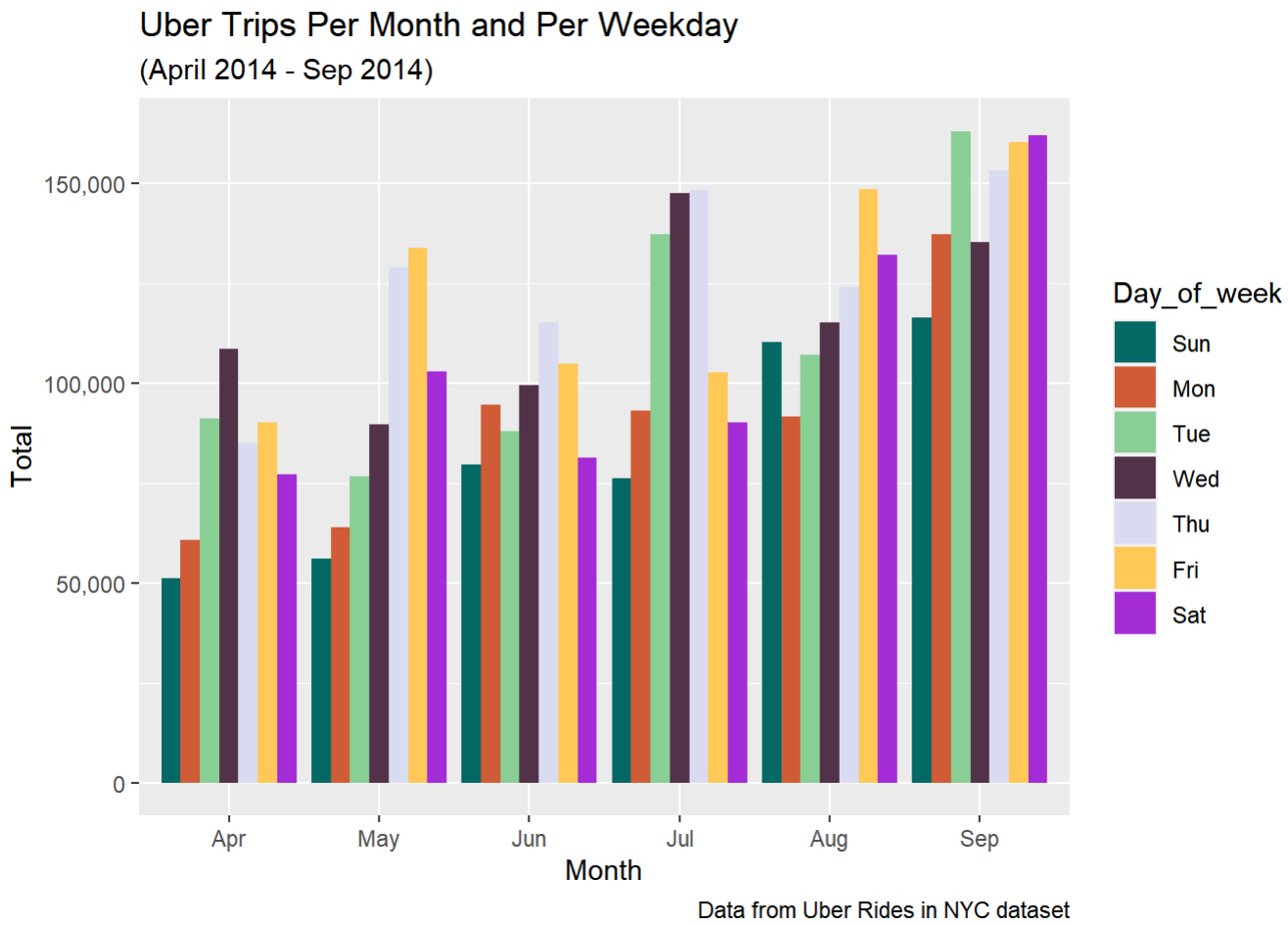
Now that we know our peak times, let’s find out which days of the week have the greatest number of trips from April to September.

```
# Lets add the colors first for the graph
colors = c("#046865","#CF5C36","#89CE94","#523249","#D9DBF1","#FFC857","#A42CD6")

month_dayofweek <-
  uber_data %>%
  group_by(Month, Day_of_week) %>%
  summarize(Total = n())
```

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

```
ggplot(month_dayofweek, aes(Month, Total, fill = Day_of_week)) +
  geom_bar(stat = "identity", position = "dodge")+
  scale_fill_manual(values = colors)+
   labs(
    title = "Uber Trips Per Month and Per Weekday",
    subtitle = "(April 2014 - Sep 2014)",
    caption = "Data from Uber Rides in NYC dataset",
    x = "Month",
    y = "Total"
  )+
  scale_y_continuous(labels = comma)
```



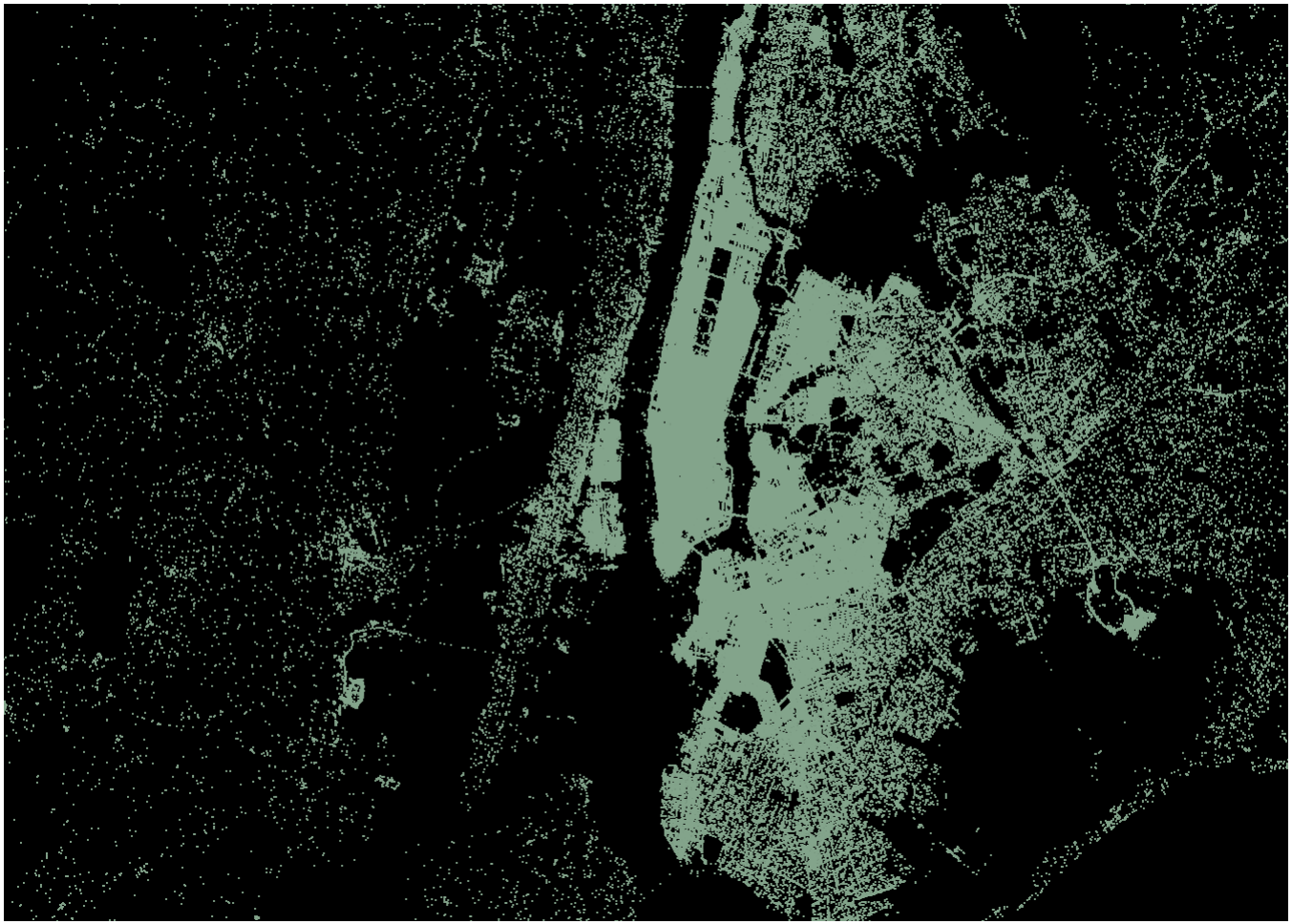
If we would have created a graph without adding months, we would have gotten a result of Thursday being the peak day however that is not the case as the peak days changes depending on the month. But from this graph, we can tell that September had the greatest number of trips with Tuesday as the peak day.

Creating a map visualization of the rides in NYC

Now that we have all the visualization mentioned earlier, let's look at the map of NYC using our Lat, Long, and Base data.

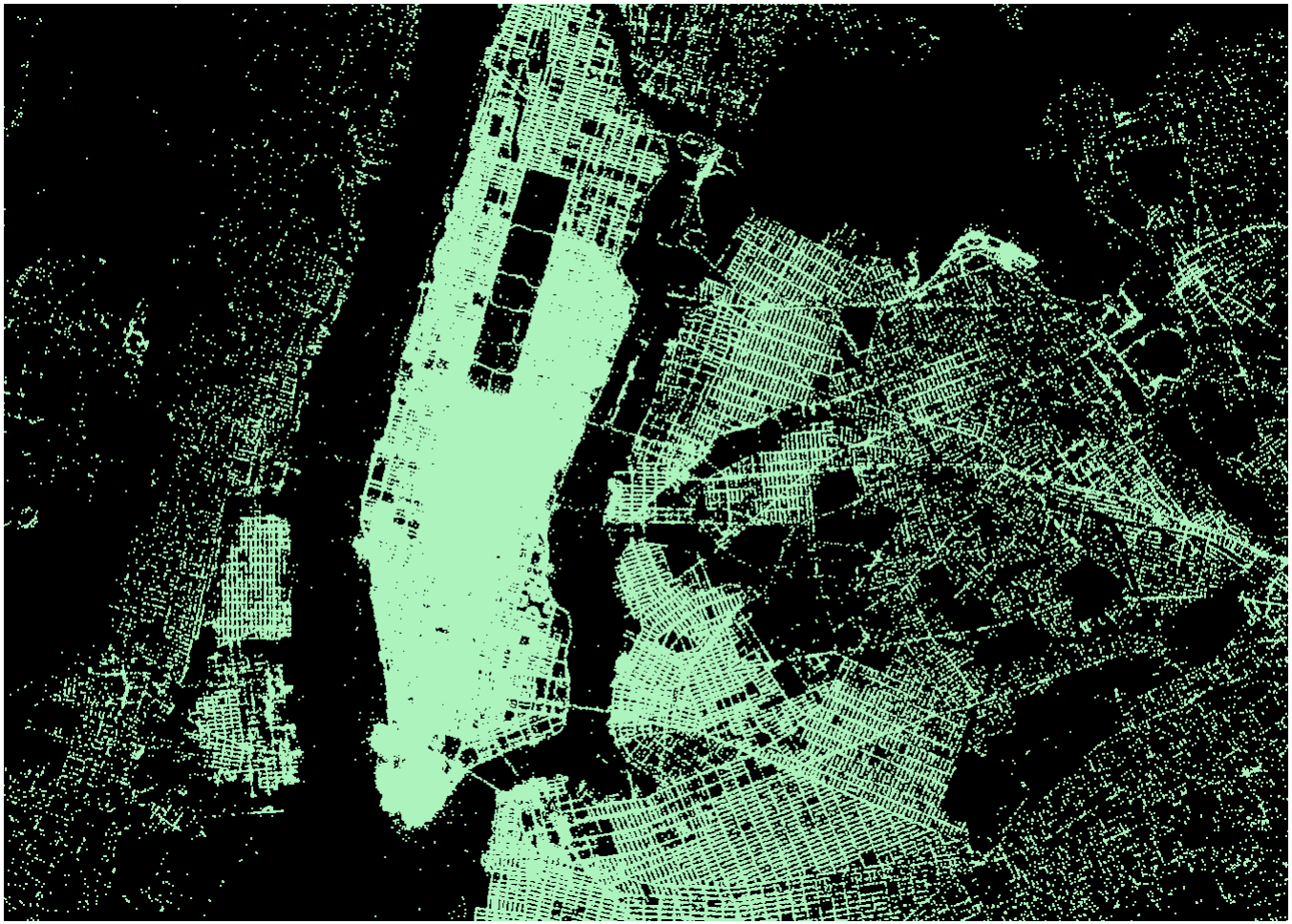
```
# First thing we got to do here is to set CRS (coordinate reference system) to make the mapping. We then transform the CRS t
o transfer them to a common CRS so they align with one another.
uber_map_data <-
  uber_data %>%
  select(-`Date.Time`, -Base) %>%
  st_as_sf(coords = c("Lon", "Lat"), crs = 4326) %>%
  st_transform(crs = "ESRI:102003") %>%
  st_coordinates() %>%
  as.data.frame()

# Next, we plot them in the map
uber_map_data %>%
  ggplot() +
  geom_point(aes(X, Y), size = 1e-12, color = "#83a48b", alpha = .9) +
  coord_cartesian(xlim = c(1800000, 1850000), ylim = c(560000, 590000)) +
  theme_void() +
  theme(panel.background = element_rect(fill = "black"))
```



We can also make a zoomed-in version of this to show the whole of Manhattan.

```
uber_map_data %>%
  ggplot() +
  geom_point(aes(X, Y), size = 1e-12, color = "#acf3bd") +
  coord_cartesian(xlim = c(1820000, 1840000), ylim = c(570000, 585000)) +
  theme_void() +
  theme(panel.background = element_rect(fill = "black"))
```



With the help of the Jpeg function and dev.off function, we can make a much better resolution image of the map mentioned above. I used #jpeg(“manh_4M.jpeg”, units = “in”, width = 75, height = 50, res = 300) and #dev.off() at the end to upload it as a imguR.

```
jpeg("manh_4M.jpeg", units = "in", width = 75, height = 50, res = 300)
uber_map_data %>%
  ggplot() +
  geom_point(aes(X, Y), size = 1e-12, color = "#acf3bd") +
  coord_cartesian(xlim = c(1820000, 1840000), ylim = c(570000, 585000)) +
  theme_void() +
  theme(panel.background = element_rect(fill = "black"))
dev.off()
```

These maps contain more than 4 million points of data, We can see from the map that, the midtown area shows the majority of the uber trips in NYC. This is most likely because the visitors, tourists, and commuters fill the city during the day. There are various attractions in these regions as well as businesses, retail, and service jobs that bring people around this area. With this analysis, Uber could do various changes to the business to improve pick-up and drop rates in this region.

- Uber could increase the number of drives in these areas during peak times and peak days to cater to everyone. Users tend to change applications when it takes too long to find a driver.
- Uber could do marketing projects in these regions to create customer exposure to the business.
- Uber could start a new system called a Uber Stand for places with a high number of trips to make it more efficient for travelers.