Data Analysis Report

Aseem Malik

05/04/2020

Description

Over past few years, I have been hearing from a lot my friends that taxi business is getting slow in city. During my undergraduate degree in NYC, I had few friends who simultaneously drove taxis as a part time job and made some living out of it. I always used to hear from them that Taxi business is not as strong as it used to be 6-7 years back. In today's time, they had to give 30% more time to work to earn the same amount of revenue as they used to do 6 years back. Hence, many times, they had to give up a lot of fun and recreational time to their work and cancel all their college plans. Therefore, listening to this, I always felt that why is that the case? Through this project, I think I got answers to many questions and many of facts got verified too. Moreover, the results from this project can also prove helpful to my friends as they can change a bit of their routine to get maximum benefit. In taxi business, time and weather plays and important role. Therefore, using these two important components, there will be a data analysis done using Uber data.

This project is a data analysis project that tells a story through visuals and tables. I will be majorly focusing on Uber Pickup data and get some insights from the data. In addition, I will then compare Uber's performance with its competitor Lyft to see who has a big market share. Data analysis storytelling is an important element in the field of data science through which many audiencea are able to understand the background of many things. In the next part of my project, I will bring in the weather data set and connect it to Uber data and see how weather affects Uber Pickups in NYC. The results from this project will be helpful to all the Taxi drivers in NYC. In addition, some parts will also be helpful to small taxi companies as they can adjust their operations according to data results and gain benefits from the data insights. Visualization is an important tool in data storytelling. Hence, this project makes use of many visualization techniques to understand the complex data and get meaningful insights that would help a lot drivers and companies to make important business decisions. I have implemented the ggplot2 library in R on the Uber Pickup data, Lyft Pick up data, and finally weather data to get some beautiful meaningful plots that can be easily comprehended by anyone. Some other packages used which are also a backbone of this project are tidyverse, lubridate, dplyr etc.

Getting the data in R.

This step invovlves reading the csv's, wrangling and formatting the columns of data. The dataset contains, roughly, six files: Uber trip data from 2014 (April - September), separated by month, with detailed location information.

CODE FOR QUESTION 1.A

```
##             Date.Time     Lat      Lon   Base       Date year month day hour
## 1 2014-04-01 00:11:00 40.7690 -73.9549 B02512 2014-04-01 2014     4   1    0
## 2 2014-04-01 00:17:00 40.7267 -74.0345 B02512 2014-04-01 2014     4   1    0
## 3 2014-04-01 00:21:00 40.7316 -73.9873 B02512 2014-04-01 2014     4   1    0
## 4 2014-04-01 00:28:00 40.7588 -73.9776 B02512 2014-04-01 2014     4   1    0
## 5 2014-04-01 00:33:00 40.7594 -73.9722 B02512 2014-04-01 2014     4   1    0
## 6 2014-04-01 00:33:00 40.7383 -74.0403 B02512 2014-04-01 2014     4   1    0
##   minute
## 1     11
## 2     17
## 3     21
## 4     28
## 5     33
## 6     33
```

In the above data frame (df_uber), which a cleaned version of the raw data, various steps have been performed to arrive at the cleaned and formatted df_uber data frame. Firstly, I downloaded six month Uber Pick up data files from Kaggle which was not cleaned and formatted. Using rbind in R, I concatenated all the rows of the six files that contained data of six months from April to September. Moreover, one of the important component in my analysis is date and time. Therefore, after rbinding, I made sure that my datetime column was formatted in a correct way. The next step involved, making columns of from date-time column, and I generated six columns from the data-time column using the lubridate and tidyverse package, namely: Date, Month, Year, Day, Hours, Minute and Second and all these columns will be thoroughly used in the coming analyisis.

Now, in the next part, after wrangling and formatting my Uber data, I will start the analysis part. Firstly, I will begin with analyzing the month level information provided.

Month Level Analysis

Getting the trips by the months in year

```
## # A tibble: 6 x 2
##   month   Count
##   <dbl>   <int>
## 1     9 1028136
## 2     8  829275
## 3     7  796121
## 4     6  663844
## 5     5  652435
## 6     4  564516
```

In the above, data frame (month_uber), I have utilized dplyr package to arrive at the final result. Firstly, I grouped the data by month and then summarized the data by getting the counts of monthly Uber trips and then arranged it in a descending order. Looking at the above data frame, it can be clearly seen that September and August are busiest months with trip counts of 1028136, 829275 respectively in a six month time frame and April is the month with lowest trip count. Institutively, September and August are back-to-school months and also summer vacation season in NYC. Hence for these reasons, we can say that August and September are busy months and drivers across the entire city should give full and maximum time to their work in these to months to get the full benefit.

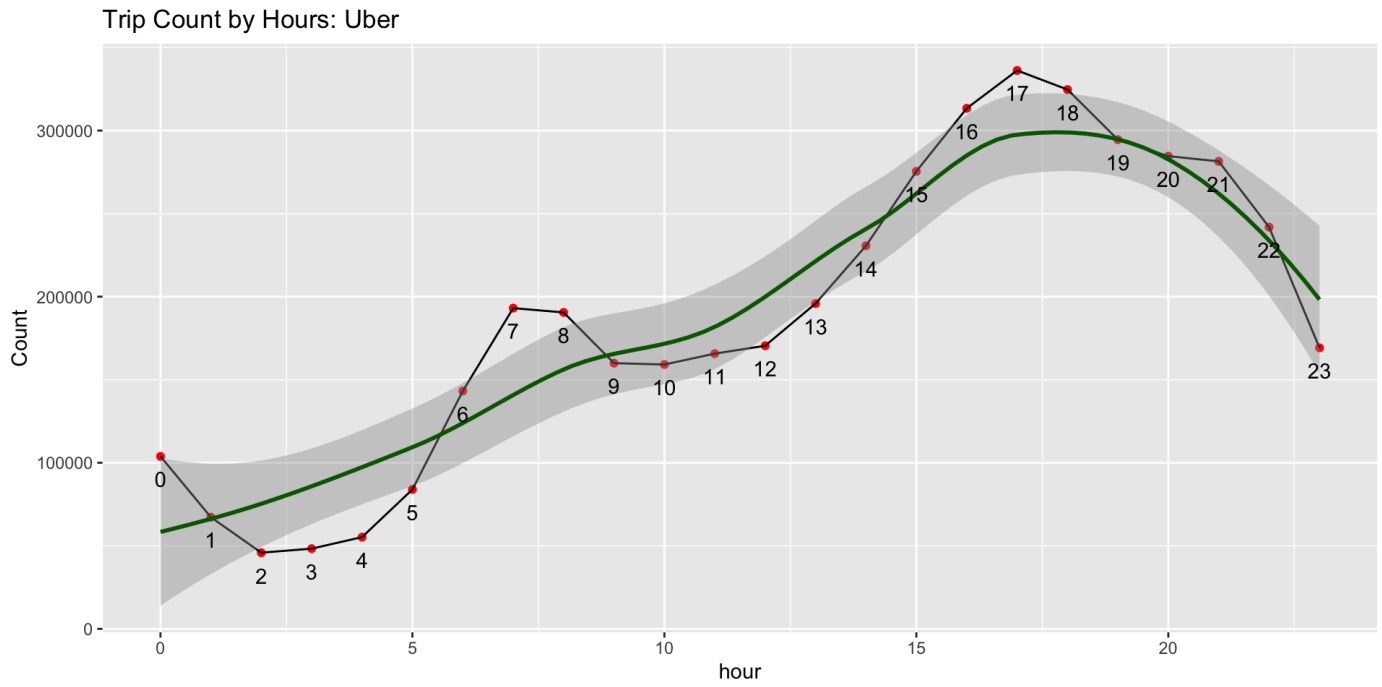
Hour Level Analysis

In the following data frame, I had again used the same methodology, where I had grouped data by hours of day and got the trip count and plotted it using the ggplot2.

```
## # A tibble: 6 x 2
##    hour  Count
##   <int>  <int>
## 1    17 336190
## 2    18 324679
## 3    16 313400
## 4    19 294513
## 5    20 284604
## 6    21 281460
```

Line Plot for the Hours Uber Trip Counts

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Looking at the above plot, we have a very interesting insight from the data. It shows that during every day over a six month course of time, 5pm is the busiest hour of Uber pick up in the New York City with count of 336190 and from my understanding this makes sense too as most of the offices, colleges and schools in NYC get dispersed at this hour. In addition, subways are extremely crowded during 5pm which also supports my result that taxi business also gets busy during this time. Hence all drivers should focus during on driving this time and take full advantage of this hour. Moreover, between 4pm-9pm, all driver should give maximum time to their services as these times have high pickup counts. Therefore, drivers have more chances of more getting rides during these hours than rest of the hours in the day. In addition, 1am-5am are the slowest hours in NYC as most of the peole are asleep during this period of time. Hence, the counts too are really low. Reading through google, I also found that Uber increases its prices when there is a high demand for vehicles.

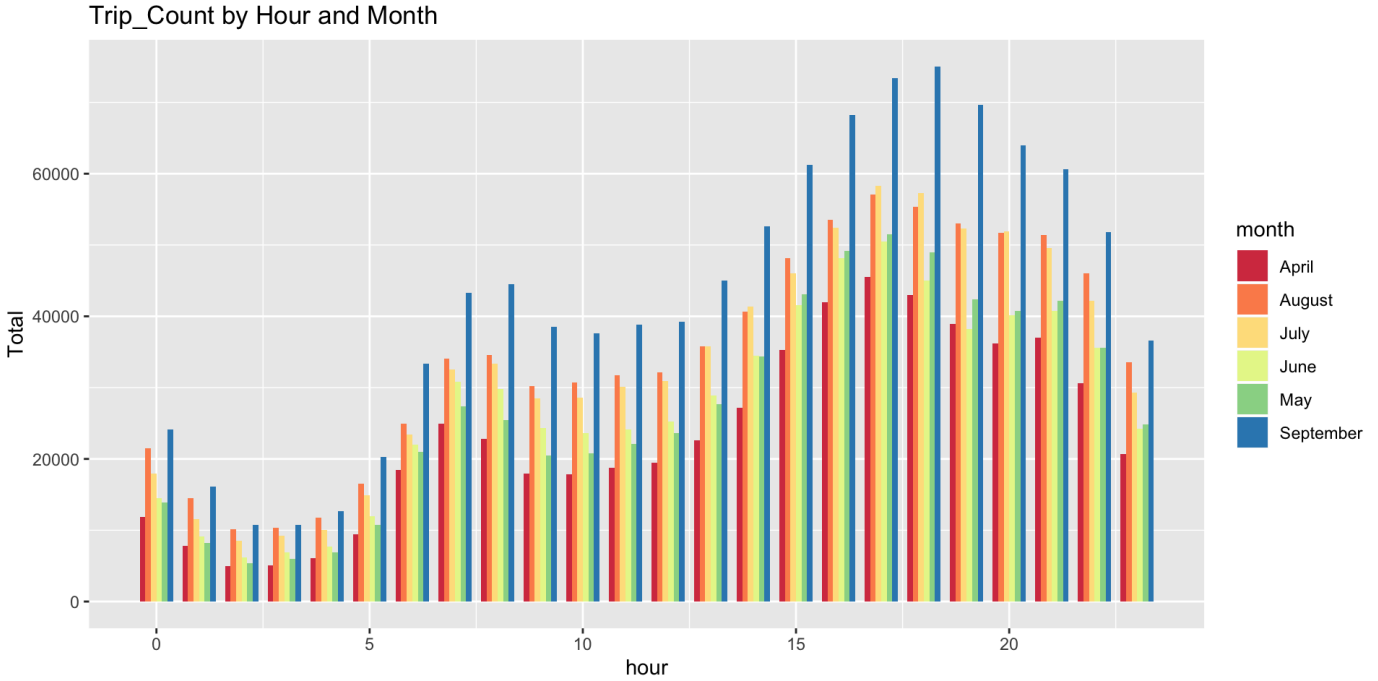
Therefore, fares are known to increase during peak times such as rush hour (evening hours). Moreover, Surge pricing can boost the cost of rides to multiple times the normal rate. Hence from customer perspective, evening hours are not favorable to book rides as the customer would end up paying more in comparison to other times in a day but for a driver, these hours are most important as they can make most of their money in these hours.

Month-Hour Level Analysis

Getting the trips by the months and Hours

Now, we will dig further in the data to and do hour level analysis for each moth to find out which hours in which months are the busiest. Same packages and methodolgy is used to summarize the data used in month-level analysis. However, I have implemented the case_when function to fit the names of months to the month numbers. For example 4 - April. This was done to get correct and accurate results in the visuals.

```
## # A tibble: 6 x 3
## # Groups:   hour [6]
##    hour month Total
##   <int> <dbl> <int>
## 1    18     9 75040
## 2    17     9 73373
## 3    19     9 69660
## 4    16     9 68224
## 5    20     9 63988
## 6    15     9 61219
```



Looking at the above data and plot, results again confirm that September is the most busiest month in the six month time frame and looking at the data from hours perspective, evening hours from 3:00pm-8:00pm in the month of september are the most busiest hours because at this time it is office and school dispersal times. Conversely, May and April are slowest months in terms of business. In addition, for obvious reasons, the trip count is lowest in the midnight for all the months but the trip count for midnight hours is extremly low for the month of April and May. The busiess of Uber supposedly picks at the end of quarter three as it is time of summer vacation and NYC gets really busy in summer time. Hence, again an important indictor for all drivers in business that should give maximum time to their work during summer time.

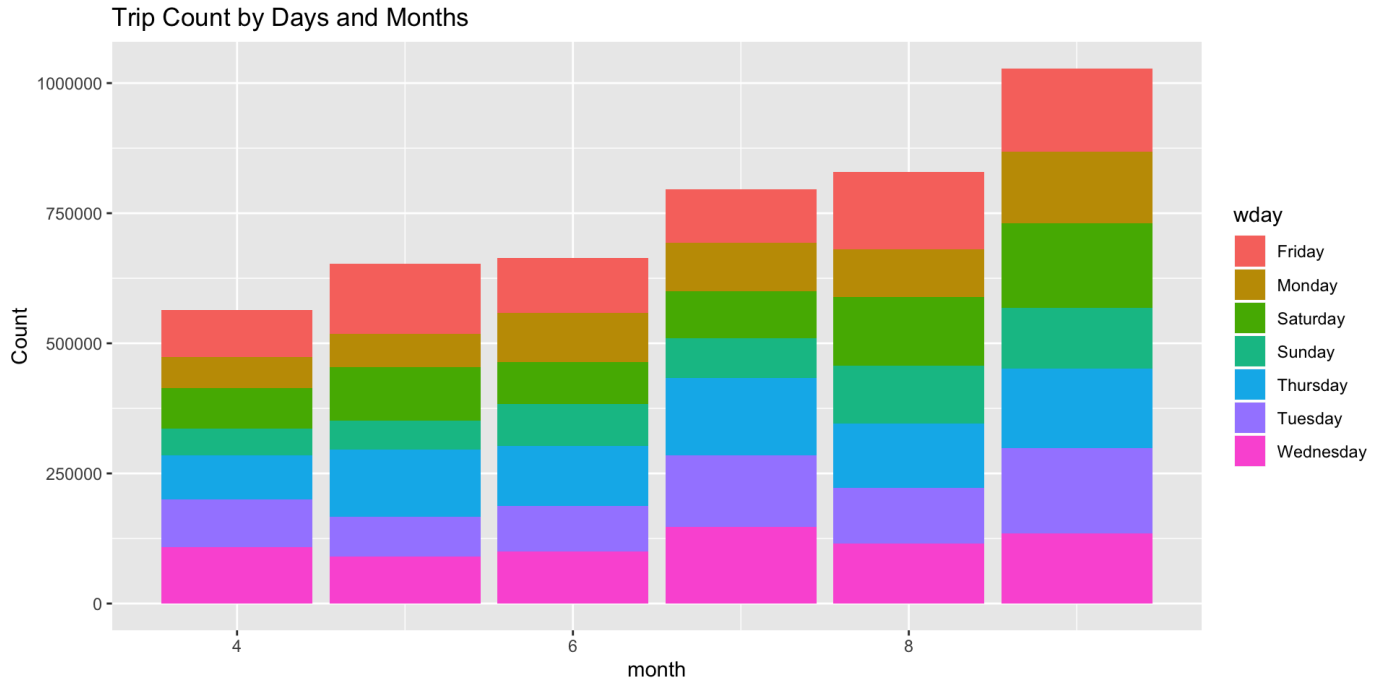
Weekday-Month Level Analysis

Getting the trips by the Weekdays and Months

In order to get the following data frame, I made an another column in df_uber which shows the names of days instead of numbers and the same dplyr package is used to summarize the results.

```
## # A tibble: 6 x 3
## # Groups:   month [3]
##   month wday      Count
##   <dbl> <chr>     <int>
## 1     9 Tuesday  163230
## 2     9 Saturday 162057
## 3     9 Friday   160380
## 4     9 Thursday 153276
## 5     8 Friday   148674
## 6     7 Thursday 148439
```

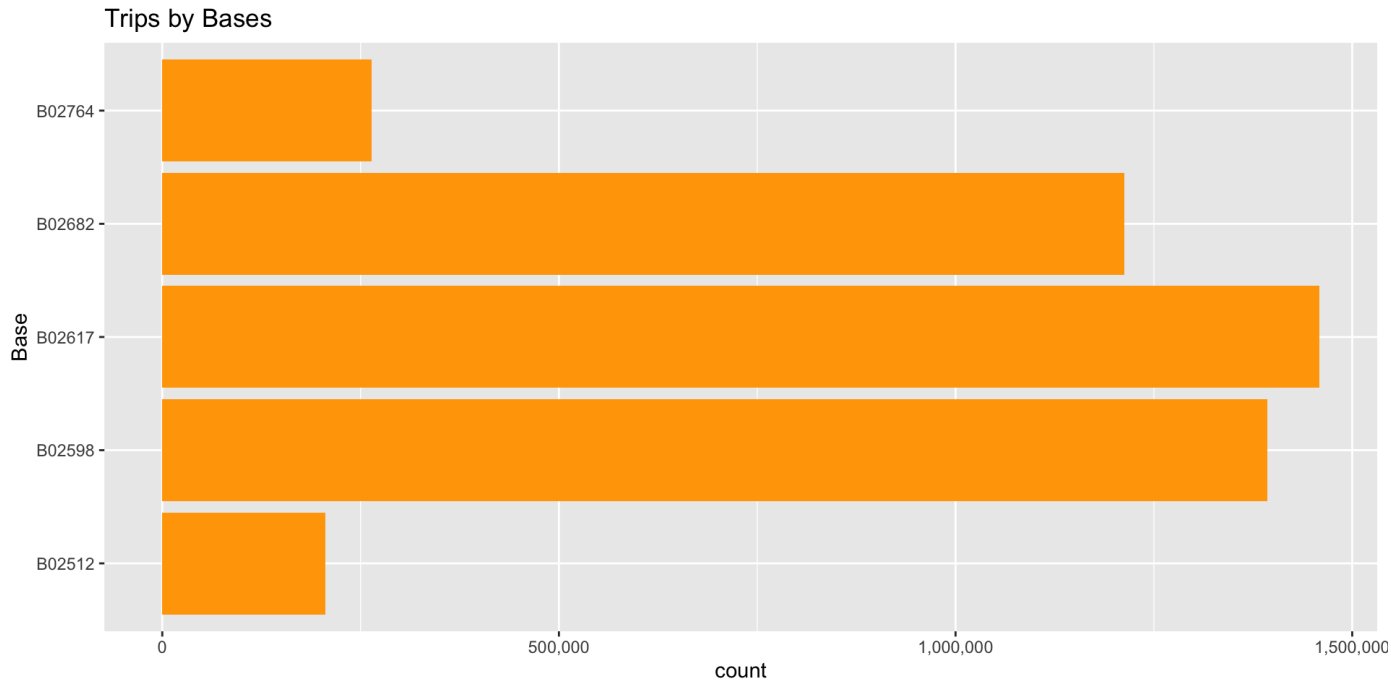
Looking at the above data frame, Tuesday in September is the busiest day with the count of 163230 followed by Saturday and Friday. Taking testimony from my friends who drives Uber, she said, “In general, Friday is the most busy week day for Uber driver” and my results confirm too. Contrarily, Sunday, Mondays have lowest trip counts across all the months. Sunday is an off day in NYC. Hence, it makes sense too and also supports the result drawn from the data. The below plot describes the above result in a more comprehensive way.



Base Level Analysis

Base refers to tracking codes that are given by TLC Base company to all the taxi companies and every taxi that run on road has a Base Code assigned to it. For a layman and in fact for divers, this analysis is not that important but for companies, it’s a vital information to know how many taxis are being operated under each base and whether the limit of base has been exhausted or not. In addition, according to one my friend’s who drives tax, “Bases are also used to classify a car to a base, for example, Suv’s will be assigned to particular base in which contains all Uber XL cars. This level of information is really important for TLC commission to keep track on the amount of taxis circulating in the city. In 2014, Uber had 7 bases:

B02598 : Hinter,B02617 : Weiter,B02682 : Schmecken,B02764 : Danach-NY,B02765 : Grun,B02835 : Dreist,B02836 : Drinnen



In the above plot, again same approach has been used to arrive at the summarized data frame. The results show that B02617 : Weiter base has the highest count of taxis with 1458853 taxis followed by B02598 Base with count of 1393113 and the Base B02512 has lowest taxi count 205673.

Competitor Level Analysis

Uber vs Lyft

Uber's biggest competitor is Lyft. According to second measure, together, the two ride-hailing giants capture more than 98% of market spending. Therefore in next segment of this report, I will show some insights from Lyft's data and compare it with Uber's performance. In addition, I first started with reading the CSV file as I did with Uber CSV's. The Lyft data is a 3 month data for one Base. The data contains 3 columns that are datetime stamp, longitude and latitude. Again, same methodology of cleaning and formatting was used on this dataset to arrive at the final data frame. I again made time columns in data like day, date, year, hours etc. to have concrete foundation to begin with analysis. Researching over google, Lyft came in business in the year 2012 and Uber came in the business in the year 2010 which gives advantage to Uber over Lyft.

```
##          time_of_trip start_lat start_lng year month day hour minute
## 1 2014-09-04 09:51:00  40.64705 -73.77988 2014     9   4    9     51
## 2 2014-08-27 21:13:00  40.74916 -73.98373 2014     8  27   21     13
## 3 2014-09-04 14:16:00  40.64065 -73.97594 2014     9   4   14     16
## 4 2014-09-04 16:08:00  40.75002 -73.99514 2014     9   4   16      8
## 5 2014-08-28 02:41:00  40.76715 -73.98636 2014     8  28    2     41
## 6 2014-09-13 03:57:00  40.70707 -74.01211 2014     9  13    3     57
```

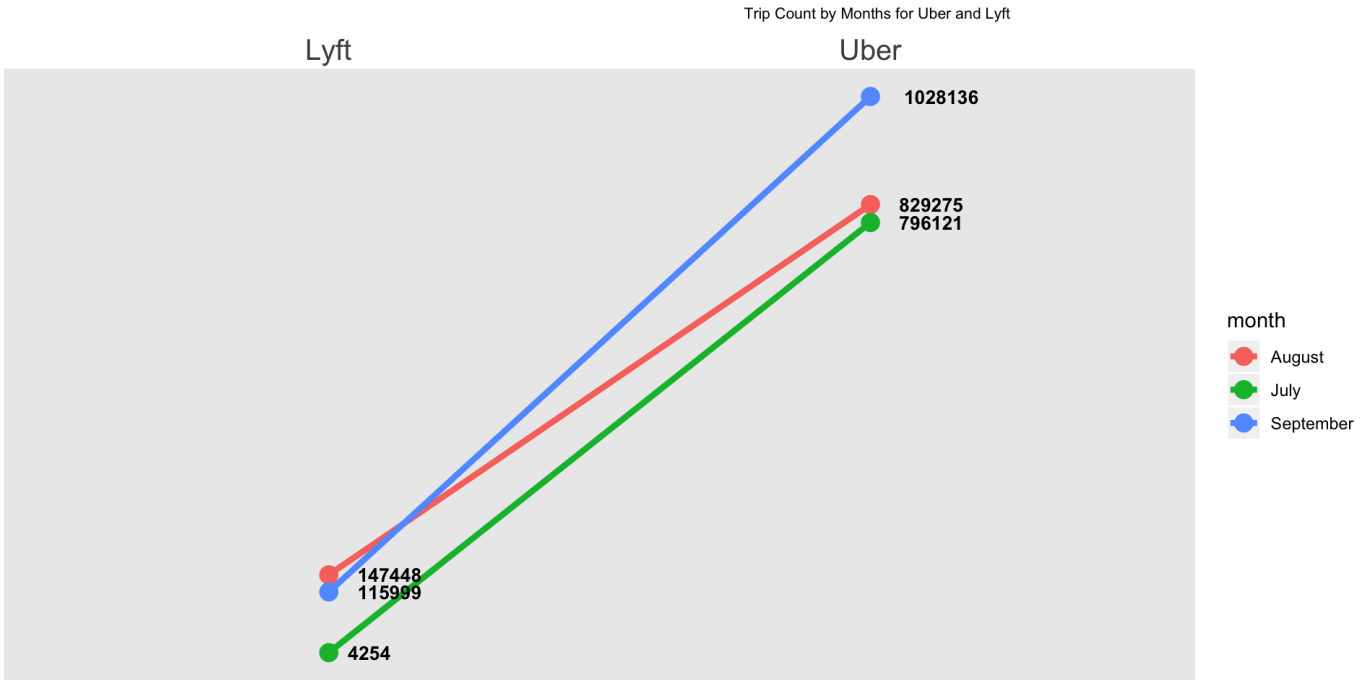
The above dataframe has same columns as Uber_df. Therefore, the consistency in both dataframes will ease the process of analysis.

Month Level Analysis: Uber Vs Lyft

I first start comparing the months in order to show which months are most busy for both the companies. This information is really helpful for drivers as a lot of drivers in NYC are enrolled in both the companies. Hence, they can divide their hours of operation accordingly among both the companies in different months to take the optimum benefit. I had used Dplyr package to summarize the count of Lyft and also used ggplot to plot the data frame. In addition, after obtaining Lyft month count, I filtered Uber dataset for the same three months and rbinded it into one data. Furthermore, I created a column in combined dataset to indicate which row belongs to Uber and which belonged to Lyft. Then using ggplot, I created slope plot which was really tedious to make as it had a lot of small pieces of code involved.

```
## # A tibble: 3 x 2
##   month  Count
##   <dbl>  <int>
## 1     8 147448
## 2     9 115999
## 3     7   4254
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

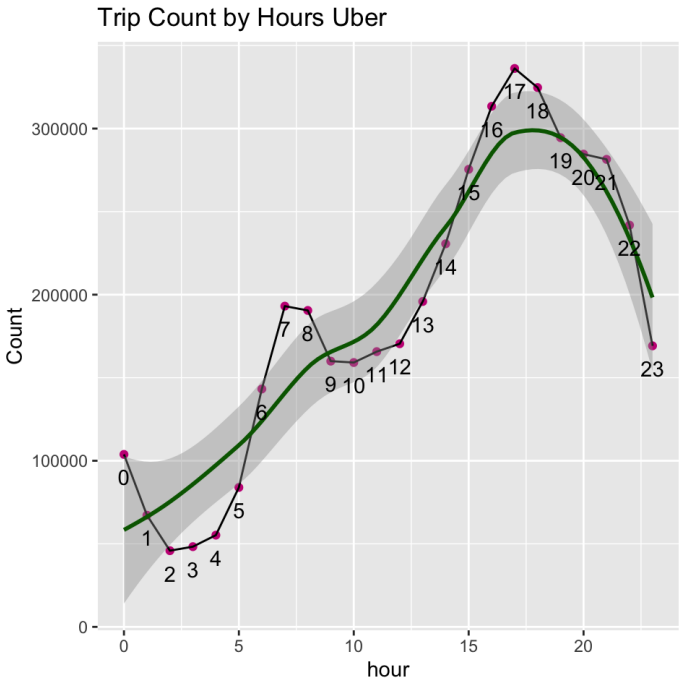


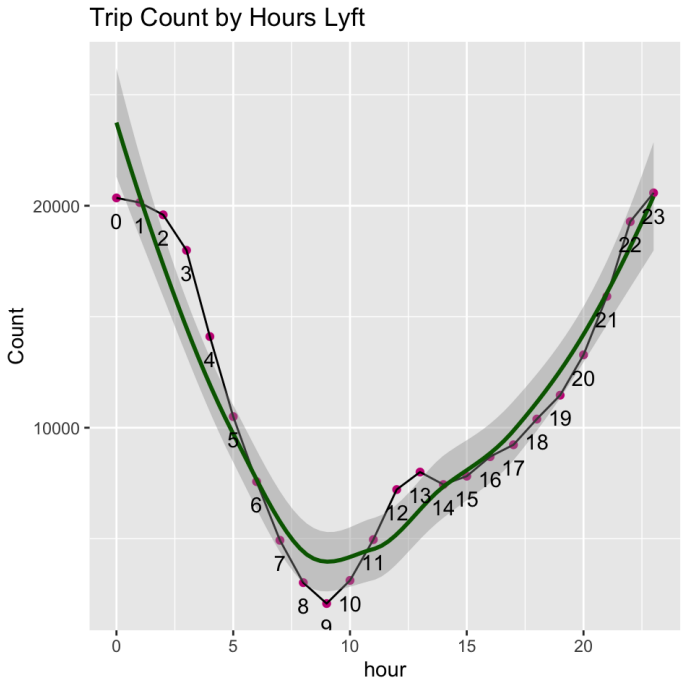
Looking at the above slope plot, we can clearly see that Lyft trip count is really small for all months in comparison to Uber. I personally believe that this result is not perfectly accurate and the main reason is that Uber started in year 2010 and Lyft in the year 2012. Since this data is from year 2014, hence Lyft was not that old in the market in comparison to Uber. Therefore, lower trip counts for Lyft makes sense. Given, todays market, both companies make up most of the market share of the industry. If I had this data of relatively current years, then the counts for Lyft have been very similar to Uber.

Hours Level Analysis: Uber Vs Lyft

In this part, I will summarize data for both companies on an hourly basis. This part will include a grid of plots to show side by side comparison.

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

Looking at the above grid, Uber has a linearly increasing curve, it also shows that early evening hours are really busy for Uber and on the other hand Lyft has U-shaped curve, conversely in comparison to Uber, early evening hours are not that busy. Lyft gets really busy from 3am-5am even though the trend is declining but for Uber these hours are really slow. This graph will be really helpful for the drivers who drive taxi for both companies as it would help them to divide their hours among both companies based the return in terms of trip which they could potentially get as results are really different for both the companies.

After looking at the competitor analysis, now I will move to second data set that is weather data and join it with our original Uber data frame.

Uber Weather Count Analyis

In this part of project, I had spent almost 8-9 hours just to clean and format my weather dataset. I had faced a lot of challenges which I will discuss later in the report to arrive at the cleaned data set that is shown below:

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

```
## # A tibble: 6 x 13
##   Date       Maximum Minimum Average Departure   HDD   CDD Precipitation NewSnow
##   <date>       <dbl>   <dbl>   <dbl>     <dbl> <dbl> <dbl>         <dbl>   <dbl>
## 1 2014-04-01      59      39    49         1.6    16     0           0           0
## 2 2014-04-02      53      41    47        -0.8    18     0           0           0
## 3 2014-04-03      64      42    53         4.8    12     0           0.07        0
## 4 2014-04-04      47      39    43        -5.5    22     0           0.19        0
## 5 2014-04-05      54      39    46.5      -2.4    18     0           0           0
## 6 2014-04-06      59      37    48        -1.3    17     0           0           0
## # … with 4 more variables: SnowDepth <dbl>, year <dbl>, month <dbl>, day <int>
```

Methodology:

I had six pdf files which I downloaded from climatological data website for all six months. After getting the files, I wrote a for loop so that the loop performs cleaning and formatting on all six files at one time. The package used to read and then convert these pdf files to a tabular form was pdftools. Inside the loop, I first began reading each file and then removed the lines which were not needed as pdftools gives the output in text form. After that, I removed all the empty spaces from weather data and also created a vector of column names which would be mutated after the data was wrangled. The next step was to make the data as data frame and it was done using dplyr package. At the end, inside the loop we had six data frames that were mutated to an empty list outside the loop. After the entire loop ran, the list outside the loop was now actually a list of six data frames.

The next step in the procedure was to rbind all the six data frames in the list into one main weather data frame. After rbinding, each column of the main data frame was formatted, ex: Date was formatted to data format, average temperature was formatted to numeric and this is how all columns were formatted to their respective format. This step was performed as the data was read as text by pdftools. Therefore, it was important to fix their formats. After this, all NA values were replaced with zeroes and further columns like day, minute etc. were also created using the date column in the weather data set to facilitate the further analysis process.

```
## # A tibble: 6 x 14
## # Groups:   day [6]
##     day month Count Date       Maximum Minimum Average Departure   HDD   CDD
##   <int> <chr> <int> <date>       <dbl>   <dbl>   <dbl>     <dbl> <dbl> <dbl>
## 1    13 Sept… 43205 2014-09-13      69      56    62.5      -4.8     2     0
## 2     5 Sept… 42319 2014-09-05      85      71    78         8.2     0    13
## 3    19 Sept… 41017 2014-09-19      63      51    57        -8.1     8     0
## 4     6 Sept… 40520 2014-09-06      88      67    77.5       8       0    13
## 5    18 Sept… 40274 2014-09-18      76      52    64        -1.4     1     0
## 6    12 Sept… 39540 2014-09-12      76      57    66.5      -1.1     0     2
## # … with 4 more variables: Precipitation <dbl>, NewSnow <dbl>, SnowDepth <dbl>,
## #   year <dbl>
```

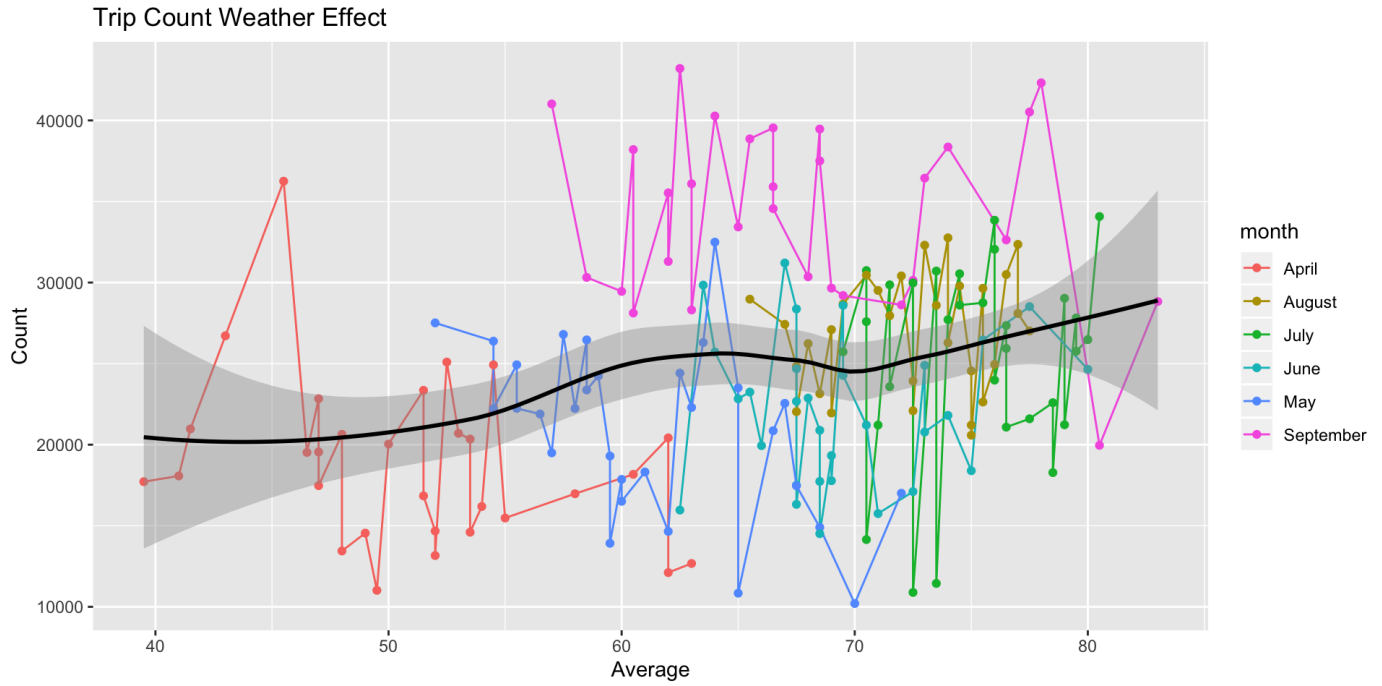
The above data frame shows a left join between Weather and Uber Data. The join was done based on date.

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

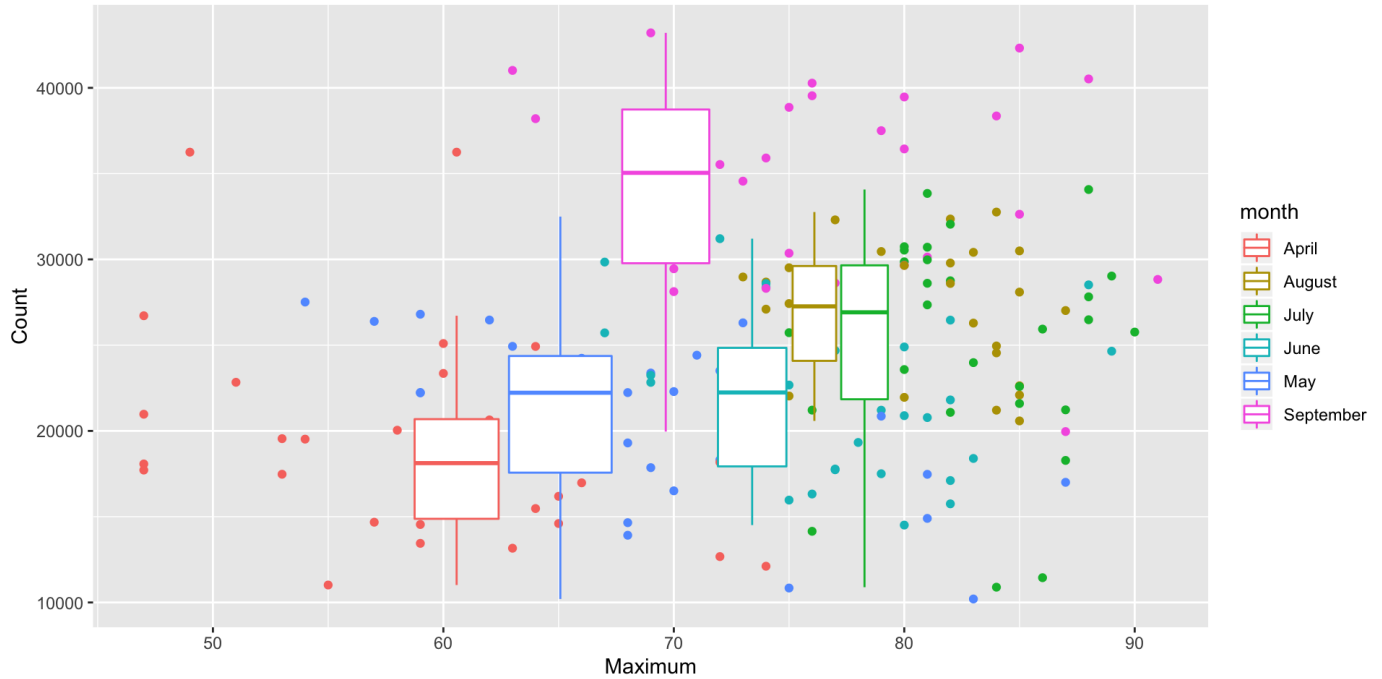
```
## Warning: Removed 3 rows containing missing values (geom_path).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```




```
## Warning: Removed 3 rows containing missing values (stat_boxplot).

## Warning: Removed 3 rows containing missing values (geom_point).
```



Looking at the line plot above, we can clearly see September is the month with highest counts in comparison to all other months. Using Geom_Smooth, we can see the trend is increasing but at very slow rate, meaning as the average temperature increases, the trip count also increases at very slow rate. Each month is represented through different colors in the plot and every month has 30 to 31 points on the plot which are nothing but the days of month. If we see the data from day level, it is highly variable like zig-zag. However, it can be concluded that as the average temperature increases, the trip count also increases but at a very slow rate which is also an important data result for drivers as they can plan their day accordingly to get the maximum business.

Boxplot in basically divides the data set into three quartiles. it shows the minimum, maximum, median, first quartile and third quartile in the data set. Looking at the box plot again, it shows the data is highly variable but one important insight from the data and plot is that when the maximum temperature is between 68-73F, the average trip count is maximum in comparison to other temperatures. Moreover,

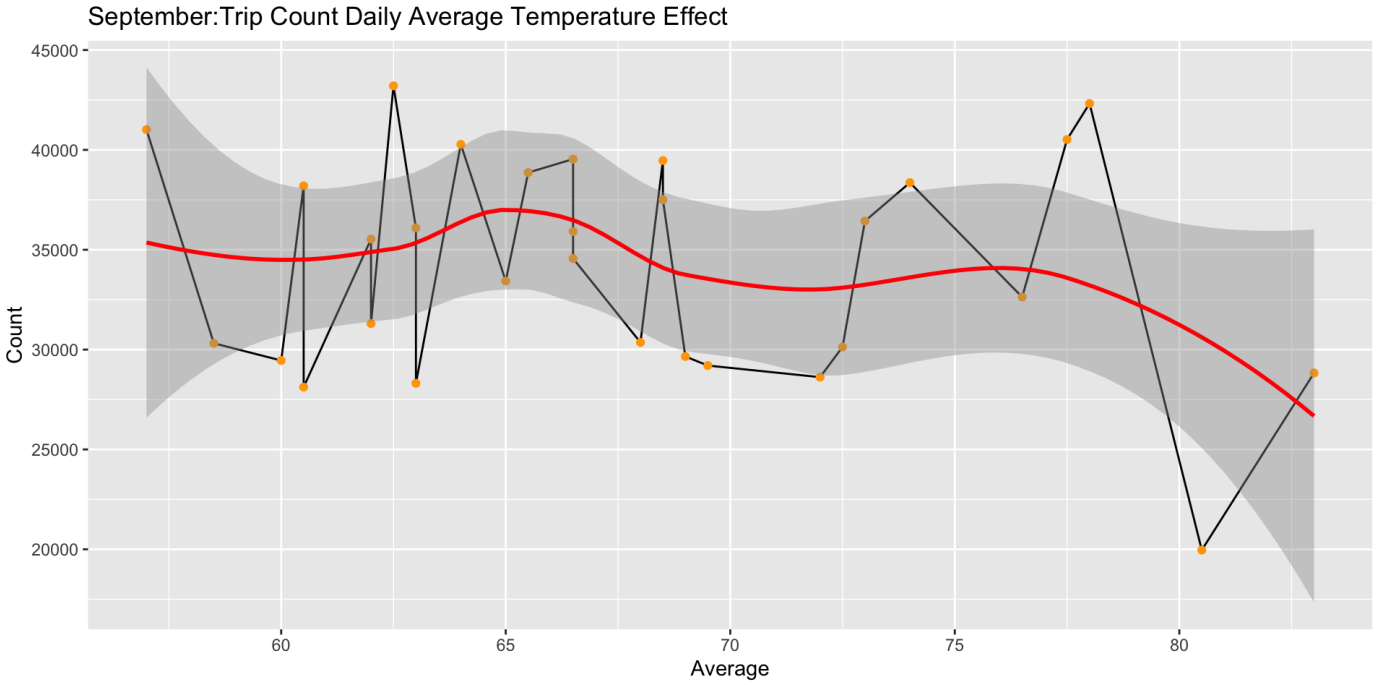
to support this, it can be said these temperatures usually show up during summer time and summer is the most busy time of the city in the year. Hence the trips counts are also really high. It can also be concluded that September has highest trip count. Therefore, in next part we will subset data and look only September to get more insights as doing analysis on weather holistically is really hard as the data is extremely fluctuating.

Subsetting the September from the data frame.

```
## # A tibble: 6 x 14
## # Groups:   day [6]
##     day month Count Date       Maximum Minimum Average Departure   HDD   CDD
##   <int> <chr> <int> <date>       <dbl>   <dbl>   <dbl>     <dbl> <dbl> <dbl>
## 1    13 Sept… 43205 2014-09-13      69      56    62.5      -4.8     2     0
## 2     5 Sept… 42319 2014-09-05      85      71    78         8.2     0    13
## 3    19 Sept… 41017 2014-09-19      63      51    57        -8.1     8     0
## 4     6 Sept… 40520 2014-09-06      88      67    77.5       8       0    13
## 5    18 Sept… 40274 2014-09-18      76      52    64        -1.4     1     0
## 6    12 Sept… 39540 2014-09-12      76      57    66.5      -1.1     0     2
## # … with 4 more variables: Precipitation <dbl>, NewSnow <dbl>, SnowDepth <dbl>,
## #   year <dbl>
```

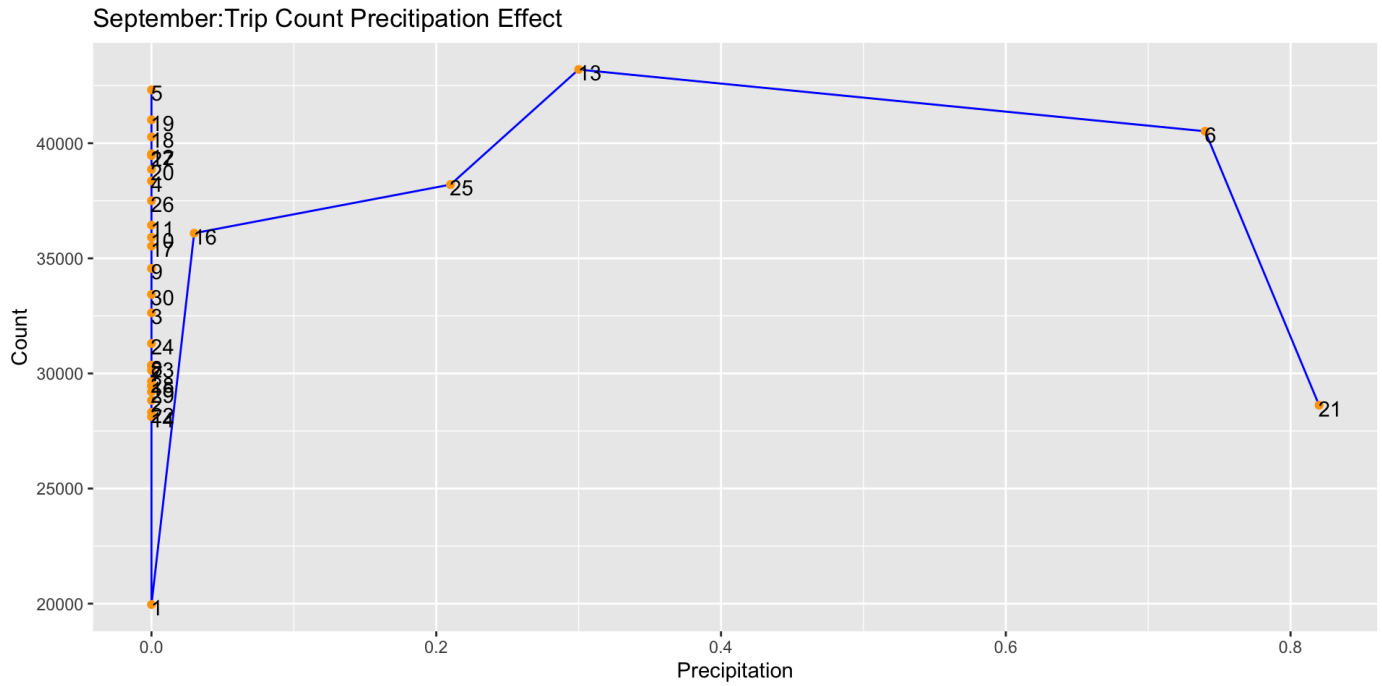
Making Plots

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



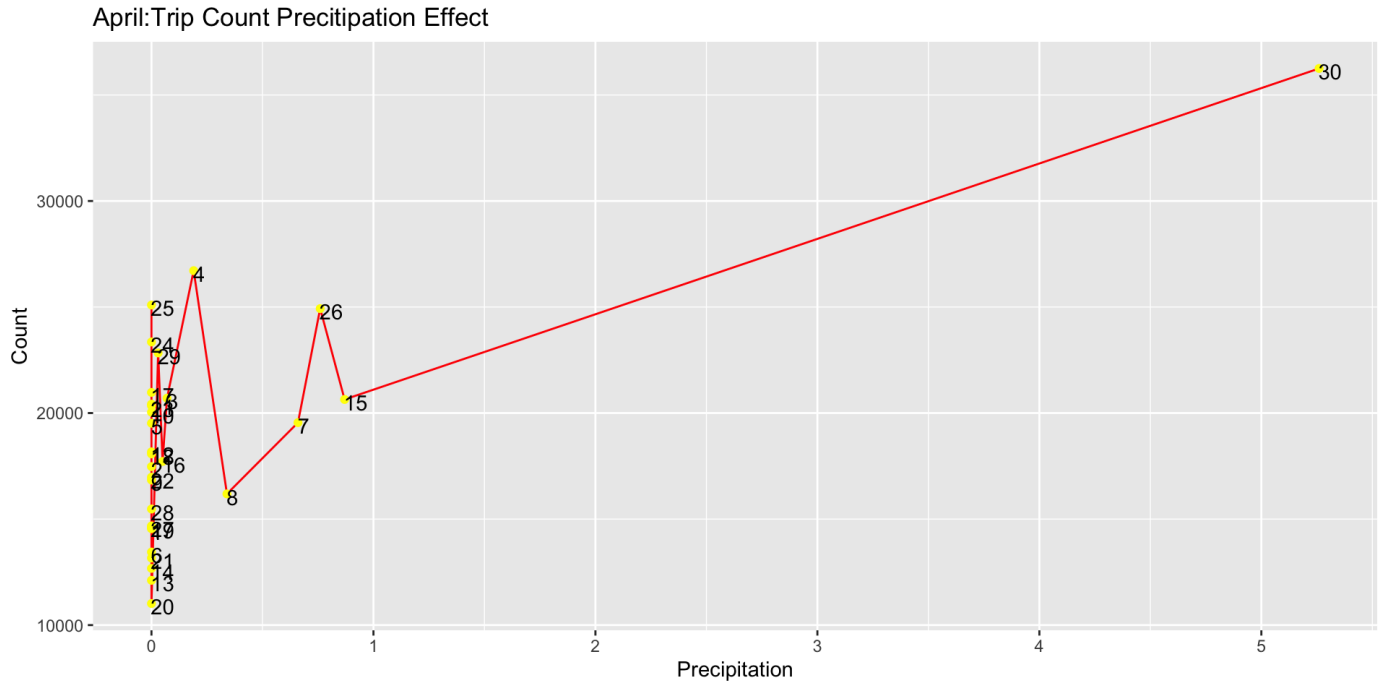
Looking at the plot above, no concrete interpretation can be done as the trip count is pretty much constant for range of temperatures. However, if divide the plot in 2 parts, it can be said that lower temperatures have little higher count than High temperatures. Moreover, I believe that September is mostly the end of summer time and people in city NYC usually enjoy warmer temperatures, therefore there are less Uber bookings.

Now we will see how rain affects the business of taxis.



Looking at the plot above, When there is no precipitation, the trip count is highly variable for the respective days and there could be may reasons behind it. However if look on the right side of the plot, we can say as the precipitation level increases, the trip count also increase but you might be wondering that why there is steep decline in count from 13 to 6 to 21. The main reason behind the steep decline is that both 6 and 21 are Saturday and Sunday respectively in September in 2014 and from the earlier analysis we know that both days are slow in NYC as the city is mostly closed on both days. In order to consider the result, we will again verify it by looking the month of April as in the weather data shows that April had most precipitation.

Looking Precitipation effect on the trip count in the month of April



Looking at the precipitation line chart for the month of April, we can easily conclude now that as the precipitation level increases in NYC, the trip count also increases. Higher the level of precipitation, higher the trip count for Uber. Moreover, we can see from the chart that there is sharp increase in trip count for 30th day as this day in the month had highest level of precipitation at 5.0. Hence rainy days are significantly related to the Uber trips. According to the article “Has Uber Made It Easier to Get a Ride in the Rain?” by Abel Brodeur and Kerry Nield, “An increase in demand from rain will cause the Uber fare rate to increase. Hence, it is favorable situation for the divers but customers end up paying significantly more for taxis during rains.

Challenges faced in project:

- Converting pdf text to data was not an easy task in R for me. There is library called tabulizer that easily does it but that library was not working on my computer. Therfore, I had to use another library called pdftools which was a difficult was of converting pdf in to table as it reads all the pdf file as lines of text. Hence a lot of formatting and wrangling was performed in order to arrive at the cleaned data.
- Since, I am new to R and I am not very comfortable in writing loops in R but I had to write one for this project. As mentioned in point one, the wrangling of pdf data was really hard and doing the same process over six files would have been very time consuming.

Therefore, I decided to write all the wrangling and formatting process inside the loop and the main issue which I encountered was to store result of my loop each time in a list but after 5 hours of search, trial and error, I found a way and made list of dataframes.

- Finding historical weather data was also not easy as many websites would ask a price for it.
- Doing analysis over weather data after combining it with Uber data and coming with meaningful results was also a challenging situation as weather data was highly variable. For example, one day in the month of September would be 60F and the other day would be 85F, hence made it difficult for me to write something conclusive and concrete on that result.
- Making a slope plot was also time consuming as it had multiple components to it in terms of formatting and dividing of the data. In general, making a cleaned plot in R takes about 15-20 mins with some preprocessing and summarization but making of this plot almost took 1.5 hrs for me.
- Formatting in RMD file was also a challenge faced by me as this was the first time in this class when I was formally introduced to RMD files

Conclusion

In conclusion, this project enhanced my skills in R as it helped me to tie everything together. Since the start of semester, we had learnt many libraries in R and did a lot of Homeworks on them individually. Doing a project like this helped me to implement them holistically and get an overall view about the practical implementation. I now feel very confident in working in R with RMD file, libraries like ggplot, tidyverse, dplyr etc. In addition, the project significantly helped me to enhance my data visualization skills and also helped me to improve my skills of data storytelling which is important tool in data science. I personally feel that this project has prepared me for my future to deal practical data problems.

```
## [1] "Thankyou"
```

Github Link

"https://github.com/amalik0205/PROJECT"

Works Cited

1. "Uber"Uber Pickups in New York City." Kaggle, www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city.
2. R Libraries
3. Lecture Notes 4.Prabhakaran, Selva. "Eval(ez_write_tag([[728,90],'r_statistics_co-Box-3','ezslot_1',109,'0','0'])) ;Top 50 ggplot2 Visualizations - The Master List (With Full R Code)." Top 50 ggplot2 Visualizations - The Master List (With Full R Code), r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html.