Uber Dataset Visualization and Analysis

Mini Project Report submitted in partial fulfilment.

of the requirement for the degree of

B. E. (Information Technology)

Submitted By

Sharvari Mhatre (18101A0023)
Shruti Mahishi (18101A0025)
Shreya Bhagwat (18101A0028)

Under the Guidance of

Prof. Shruti Agarwal

Department of Information Technology

Vidyalankar Institute of Technology

Wadala(E), Mumbai 400 037

University of Mumbai

2021-22

CERTIFICATE OF APPROVAL

For
Mini Project Report
On
R Programming Lab

This is to Certify that

Sharvari Mhatre (18101A0023)
Shruti Mahishi (18101A0025)
Shreya Bhagwat (18101A0028)

Have successfully carried out Mini Project entitled

"**Uber Dataset Visualization and Analysis**"

In partial fulfilment of degree course in

Information Technology

As laid down by University of Mumbai during the academic year 2021-22

Under the Guidance of
"Prof. Shruti Agarwal"

Signature of Guide

Head of Department

Examiner 1

Examiner 2

Principal
Dr. S. A. Patekar

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us with the possibility to complete this report. We express our profound gratitude we give to **Prof. Shruti Agarwal** Ma'am, our respectable project guide, for her gigantic support and guidance. Without her counseling, our project would not have seen the light of the day.

We extend our sincere thanks to **Prof. Vipul Dalal**, Head of the Department of Information Technology for offering valuable advice at every stage of this undertaking. We would like to thank all the staff members who willingly helped us. We are grateful to VIDYALANKAR INSTITUTE OF TECHNOLOGY for giving us this opportunity.

The days we have spent in the institute will always be remembered and also be reckoned as guiding in our career.

i. Sharvari Mhatre (18101A0023)
ii. Shruti Mahishi (18101A0025)
iii. Shreya Bhagwat (18101A0028)

Abstract

This project revolves data visualization using the R Language and also covers the data analytics part of the same. Here the main aim is to extract useful information from a dataset obtained from online sources with the primary use of data analytics tools and packages. As we all know data visualization is the best way to gain insights from raw data. Data analytics is the most understandable and well-known form of calculations to be performed on heaps of data. This raw data can be of hundreds of thousands of rows and columns and also can be troublesome to understand. Data visualization can bring life to this plain simple data. We as humans have a very visual memory and we can easily infer from visualized dashboards containing charts, graphs, etc. The dataset used in this project is obtained from Kaggle and contains various data rows and columns about Uber. The dataset contains valuable data that when properly passed through data visualization and big data analytics scripts can be converted into nice visuals containing charts and graphs spitting out valuable information or statistics for say. This data about Uber will reveal many useful parameters that can be later on used to analyze the business situations. This data visualization can be of utmost importance to business owners. Some part of the analyzed as well as visualized data can also be useful for customers. The customers can understand that the business is determined towards providing quality service for them. Big data analytics is now becoming the most basic need as companies are producing enormous amounts of data. This data can become the most vital in measuring the current stats.

Table of Contents

Introduction

This project involves the main concepts of big data analytics and data visualization that is crucial for the business and the customer point of view as well. This project is based on the R language and thus becomes the proper industry standard when it comes to the use of technology in it. The dataset consists of 453,000 entries and can give valuable insights when properly analyzed. The dataset contains various columns like latitude, longitude, date/time, etc. This data is collected over six months and hence we understand that the data that is being worked upon is huge and also provides the correct base for complex big data calculations. Here the main aim is to obtain visual information about the locations where the business is or can bloom. Based upon the analysis we can find out the locations where the customers prefer Uber rides over other means of transport. This project can also help find out the approximate customer density at various locations. The date/time column can also reveal more information about the time when the company can expand its reach and cater to a much larger audience. The company can, later on, make decisions about their plans and also conduct an in-depth analysis of their past performance. This way the company will profit and hence sales increase can be achieved. Complex mathematical calculations are involved in the big data analytics part but thanks to the R programming language that makes it look easy. R programming language is the industry standard when it comes to Big Data Analytics. The extensive use of packages in this code makes it very crucial for us to understand the whole work of it. There was a lot of learning involved in the whole project.

Problem Definition

Understanding the what and why of Big Data Analytics and Data Visualization is the new need when it comes to anywhere a lot of data is generated. This new way of representing data has captured a larger audience than any other form of representation. This way of representing data can help even non-technical people to decode and understand complex looking data. This is one of the reasons this new method is gaining traction.

As it is already known the company Uber generates huge data from all of its different operations. The rides that Uber offers generates data based on various parameters like pickup location, drop-off location, total travel time, time of travel, etc. To pass this enormous data through proper Big Data Analytics and Data Visualization algorithms and gain the output in proper self-describing format is the main problem definition here. The problem definition when solved will help both the parties involved, i.e. the customers as well as the business owners.

Components

4.1 Hardware Components

- A PC or Laptop with a minimum of 4 GB Ram and 500 GB Hard Disk.

4.2 Software Components

- R Programming Language
- R Studio

System Implementation

This system revolves around the use of the R programming language to visualize data found from the internet source Kaggle. The use of various packages in R makes it useful for the programmer to explore various ways to create data structures namely line graphs, charts and pie diagrams. The various packages used in this code are ‘readr’, ‘ggplot’, ‘dplyr’ and ‘lubridate’.

The goal of 'readr' is to provide a fast and friendly way to read rectangular data (like 'csv', 'tsv', and 'fwf'). It is designed to flexibly parse many types of data found in the wild, while still cleanly failing when data unexpectedly changes. The next package is a system for declaratively creating graphics, based on The Grammar of Graphics. Provide the data, tell ggplot how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details. dplyr is a package that provides a set of tools for efficiently manipulating datasets in R.

This code makes extensive use of the various functionalities available in the R programming language. This whole code works systematically. The whole code works upon first finding the files, reading data from them, etc. Once the data is read from various sources the integration of it all in is done. The next step contains importing the necessary date and time formats for the program to run and also setting the various variable and parameters necessary. The step following it is performing some complex mathematical calculations on the collected data and trying to output its result. The steps following involve using the imported packets along with the variable and parameters defined for calculation and plotting of the various graphs. The corresponding outputs are thus in the form of various visualizations. The whole aim of this project is to help a normal non-technical person gain valuable output about the complete statistics of Uber as a service company. Various sections of the code deal with the various graphs outputted respectively. Each new graph displays quickly readable and understandable information.

The various graphs that the code outputs, includes the count of trips per month, average number of trips per day, total count of trips per day, total count of trips per hour, etc. The various charts thus output valuable information which can help find out whether the company Uber is performing good or bad in various respects. The charts outputted are also interdependent, so by mix and match, we can infer more valuable information. Let’s say we compare the total count of trips per day with the total count of trips per hour on that specific day we can find out the numbers that need to be worked upon. Say the number of trips is high when it is peak traffic time and the demand is very less during the other parts of the day, the company can hence chalk out a plan to meet the required demand.

Code

```
library(readr)

library(ggplot2)

library(dplyr)

library(lubridate)


uber_apr <- read_csv("uber-raw-data-apr14.csv")

uber_may <- read_csv("uber-raw-data-may14.csv")

uber_jun <- read_csv("uber-raw-data-jun14.csv")

uber_jul <- read_csv("uber-raw-data-jul14.csv")

uber_aug <- read_csv("uber-raw-data-aug14.csv")

uber_sep <- read_csv("uber-raw-data-sep14.csv")


uber14 <- bind_rows(uber_apr, uber_may, uber_jun, uber_jul, uber_aug, uber_sep)


uber14$`Date/Time` <- mdy_hms(uber14$`Date/Time`)

uber14 <- uber14 %>% mutate(year = year(`Date/Time`),

                            month = month(`Date/Time`),

                            monthname = month(`Date/Time`, label = TRUE, abbr = FALSE),

                            day = day(`Date/Time`),

                            weekday = wday(`Date/Time`, label = TRUE, abbr = FALSE),

                            hour = hour(`Date/Time`),

                            minute = minute(`Date/Time`))
```

```
head(uber14)

str(uber14)

summary(uber14)

mean(uber14$hour)

library(anytime)

quantile(uber14$day,0.95)

sd(uber14$day)


hist(uber14$day)


ggplot(uber14, aes(monthname, fill = weekday)) +

  geom_bar() + ggtitle("Count of trips per month")


avg_Mday <- uber14 %>% group_by(day) %>% summarise(avg_perday = n() / 6)


ggplot(avg_Mday, aes(day, avg_perday)) + geom_line(size = 1.5, col = "deepskyblue4") +

  geom_point(size = 3, col = "darkred") +

  ggtitle("AvG number of trips per day") +

  scale_x_continuous(breaks = seq(1, 31, 1))


ggplot(uber14, aes(day)) +

  geom_bar(fill = "skyblue") + ggtitle("total Count of trips per day") +

  scale_x_continuous(breaks = seq(1, 31, 1))
```

```
ggplot(uber14, aes(day, col = monthname)) +
  geom_freqpoly(size = 1.5, alpha = 0.5) + ggtitle("Count of trips per day each month") +
  scale_x_continuous(breaks = seq(1, 31, 1), expand = c(0, 0))


ggplot(uber14, aes(weekday)) +
  geom_bar(fill = "darkviolet") +
   ggtitle("Total count of trips per weekday") +
   scale_y_continuous(breaks = seq(0, 800000, 100000), labels = scales::comma)


avg_Wday <- uber14 %>% group_by(weekday) %>% summarise(avg_perWday = n() / 24)


ggplot(avg_Wday, aes(weekday, avg_perWday)) +
  geom_line(aes(group = 1), size = 1.5, col = "orangered2") +
  geom_point(size = 3, col = "deepskyblue4") +
  ggtitle("Avg count of trips per week day")


ggplot(uber14, aes(hour, fill = weekday)) +
  geom_bar() + ggtitle("total Count of trips per hour") +
  scale_x_continuous(breaks = seq(0, 23, 1)) +
  scale_y_continuous(labels = scales::comma)


ggplot(uber14, aes(Base, fill = Base)) + geom_bar() +
  ggtitle("Total count of trips per base") +
  scale_y_continuous(breaks = seq(0, 1500000, 250000))
```

```
ggplot(uber14, aes(Base, fill = Base)) + geom_bar() +
  ggtitle("Total count of trips per base") +
  scale_y_continuous(breaks = seq(0, 1500000, 250000)) +
  facet_grid(. ~ monthname)


day_and_hour <- uber14 %>%
  group_by(day, hour) %>%
  dplyr::summarize(Total = n())


ggplot(day_and_hour, aes(day, hour, fill = Total)) +
  geom_tile(color = 'white') +
  ggtitle('Heat Map by Hour and Day')
```

Result and Discussion

```
> head(uber14)
# A tibble: 6 x 11
  `Date/Time`           Lat   Lon Base    year month monthname   day weekday  hour minute
  <dttm>              <dbl> <dbl> <chr>  <dbl> <dbl> <ord>     <int> <ord>   <int>  <int>
1 2014-04-01 00:11:00  40.8 -74.0 B02512  2014     4 April         1 Tuesday     0     11
2 2014-04-01 00:17:00  40.7 -74.0 B02512  2014     4 April         1 Tuesday     0     17
3 2014-04-01 00:21:00  40.7 -74.0 B02512  2014     4 April         1 Tuesday     0     21
4 2014-04-01 00:28:00  40.8 -74.0 B02512  2014     4 April         1 Tuesday     0     28
5 2014-04-01 00:33:00  40.8 -74.0 B02512  2014     4 April         1 Tuesday     0     33
6 2014-04-01 00:33:00  40.7 -74.0 B02512  2014     4 April         1 Tuesday     0     33
>
```

Fig. 7.1 Head(uber14) calculation

```
> str(uber14)
spec_tbl_df[,11] [4,534,327 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Date/Time: POSIXct[1:4534327], format: "2014-04-01 00:11:00" "2014-04-01 00:17:00" "2014-04-01 00:21:00" "2014-04-01 00:28:00" ...
 $ Lat      : num [1:4534327] 40.8 40.7 40.8 40.8 ...
 $ Lon      : num [1:4534327] -74 -74 -74 -74 -74 ...
 $ Base     : chr [1:4534327] "B02512" "B02512" "B02512" "B02512" ...
 $ year     : num [1:4534327] 2014 2014 2014 2014 2014 ...
 $ month    : num [1:4534327] 4 4 4 4 4 4 4 4 4 4 ...
 $ monthname: Ord.factor w/ 12 levels "January"<"February"<..: 4 4 4 4 4 4 4 4 4 4 ...
 $ day      : int [1:4534327] 1 1 1 1 1 1 1 1 1 1 ...
 $ weekday  : Ord.factor w/ 7 levels "Sunday"<"Monday"<..: 3 3 3 3 3 3 3 3 3 3 ...
 $ hour     : int [1:4534327] 0 0 0 0 0 0 0 0 0 1 ...
 $ minute   : int [1:4534327] 11 17 21 28 33 33 39 45 55 1 ...
 - attr(*, "spec")=
  .. cols(
  ..   `Date/Time` = col_character(),
  ..   Lat = col_double(),
  ..   Lon = col_double(),
  ..   Base = col_character()
  .. )
>
```

Fig. 7.2 Str(uber14) calculation

```
> summary(uber14)
   Date/Time                       Lat             Lon             Base               year          month          monthname          day       
 Min.   :2014-04-01 00:00:00   Min.   :39.66   Min.   :-74.93   Length:4534327     Min.   :2014   Min.   :4.000   September:1028136   Min.   : 1.00  
 1st Qu.:2014-05-28 15:18:00   1st Qu.:40.72   1st Qu.:-74.00   Class :character   1st Qu.:2014   1st Qu.:5.000   August   : 829275   1st Qu.: 9.00  
 Median :2014-07-17 14:45:00   Median :40.74   Median :-73.98   Mode  :character   Median :2014   Median :7.000   July     : 796121   Median :16.00  
 Mean   :2014-07-11 18:50:50   Mean   :40.74   Mean   :-73.97                      Mean   :2014   Mean   :6.829   June     : 663844   Mean   :15.94  
 3rd Qu.:2014-08-27 21:55:00   3rd Qu.:40.76   3rd Qu.:-73.97                      3rd Qu.:2014   3rd Qu.:8.000   May      : 652435   3rd Qu.:23.00  
 Max.   :2014-09-30 22:59:00   Max.   :42.12   Max.   :-72.07                      Max.   :2014   Max.   :9.000   April    : 564516   Max.   :31.00  
                                                                                                                  (Other)  :      0                  
      weekday           hour           minute      
 Sunday   :490180   Min.   : 0.00   Min.   : 0.0  
 Monday   :541472   1st Qu.:10.00   1st Qu.:14.0  
 Tuesday  :663789   Median :15.00   Median :29.0  
 Wednesday:696488   Mean   :14.22   Mean   :29.4  
 Thursday :755145   3rd Qu.:19.00   3rd Qu.:44.0  
 Friday   :741139   Max.   :23.00   Max.   :59.0  
 Saturday :646114                                 
>
```

Fig. 7.3 Summary(uber14) calculation

```
> mean(uber14$hour)
[1] 14.21831
>
```

Fig. 7.4 Mean(uber14$hour) calculation

```
> quantile(uber14$day,0.95)
95% 
 30 
>
```

Fig. 7.5 Quant(uber14$day,0.95) calculation

```
> sd(uber14$day)
[1] 8.744902
> |
```

Fig. 7.6 Sd(uber14$day) calculation

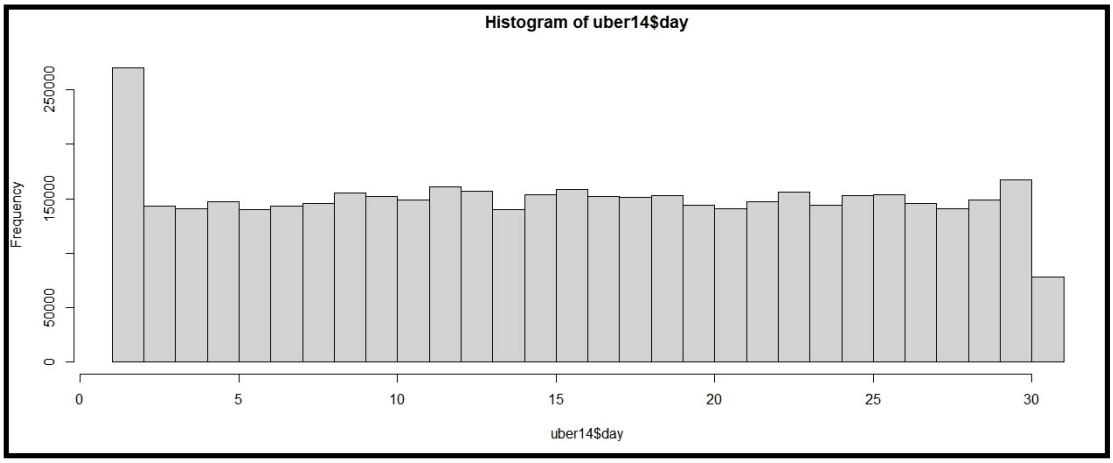

Fig 7.7 Histogram of uber14$day

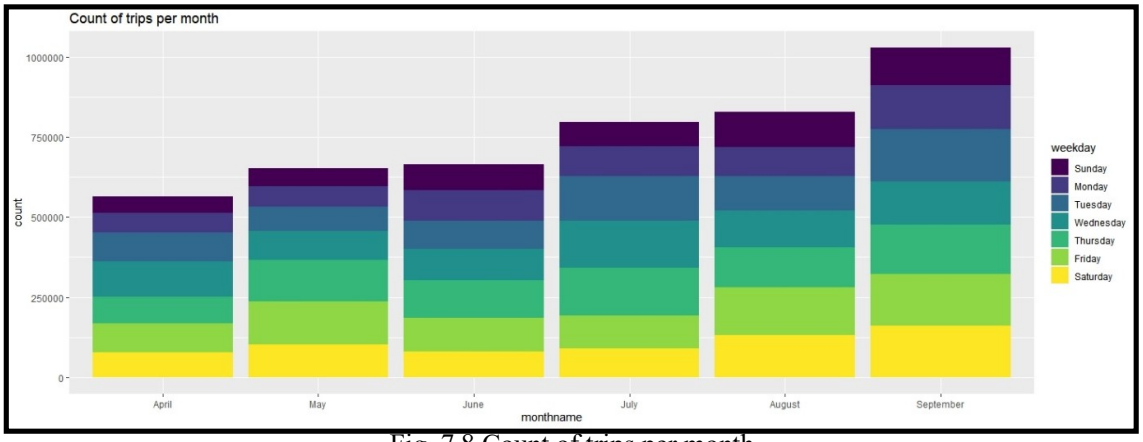

Fig. 7.8 Count of trips per month

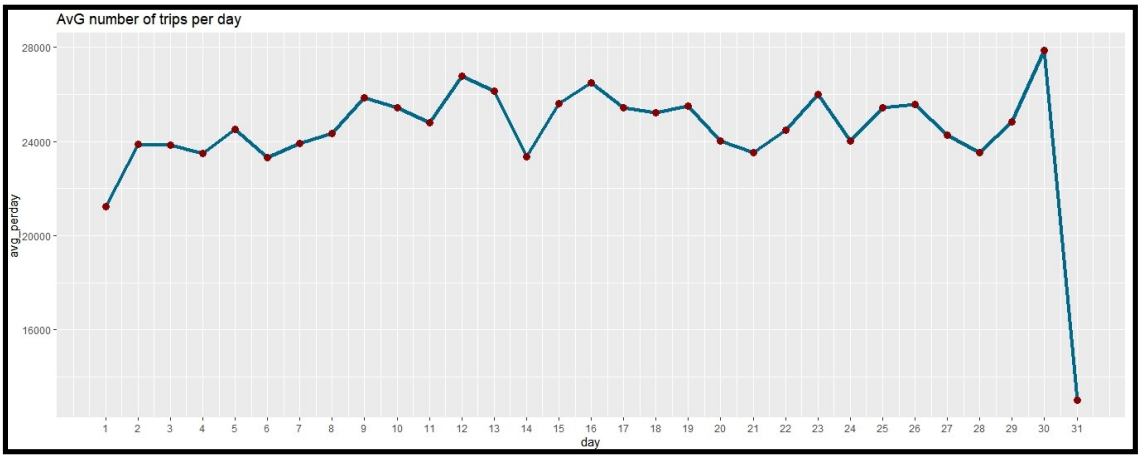

Fig 7.9 Average number of trips per day

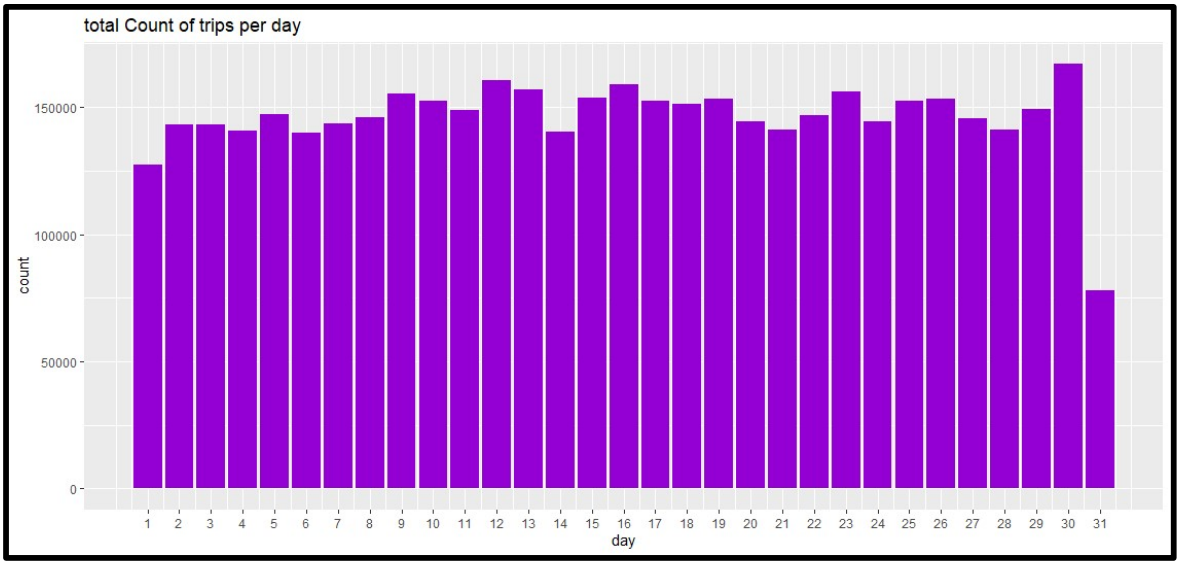

Fig 7.10 Total count of trips per day

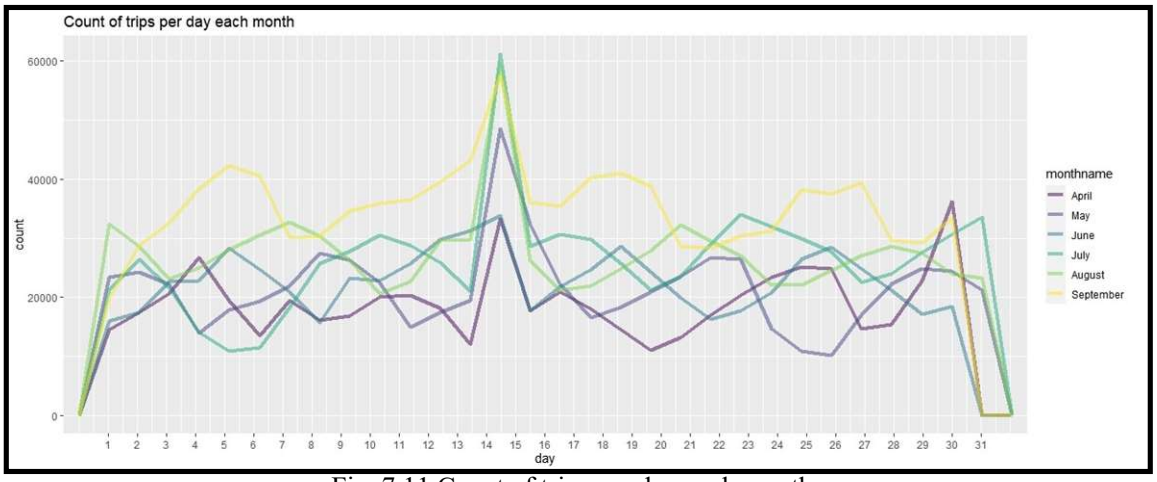

Fig. 7.11 Count of trips per day each month

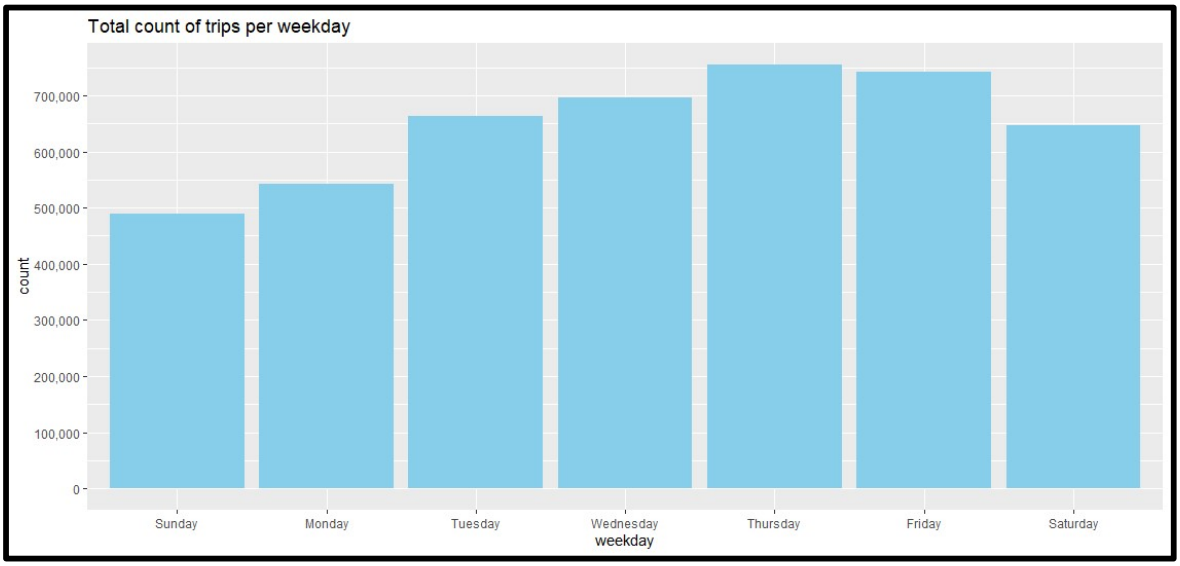

Fig. 7.12 Total count of trips per weekday

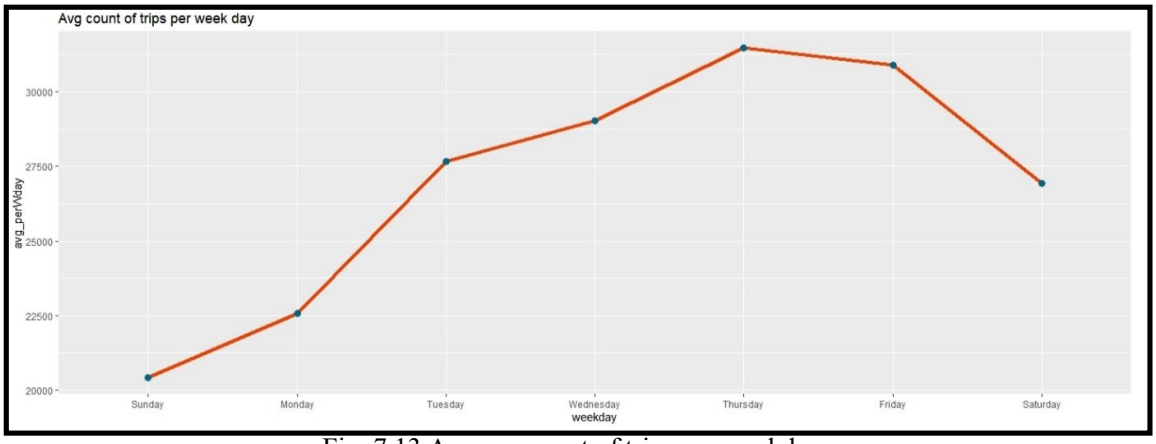

Fig. 7.13 Average count of trips per weekday

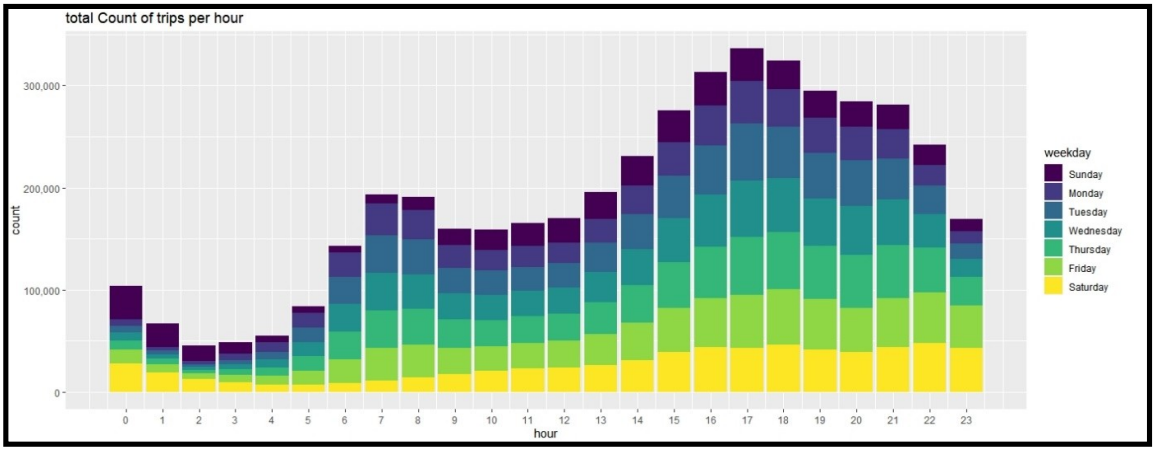

Fig. 7.14 Total count of trips per hour

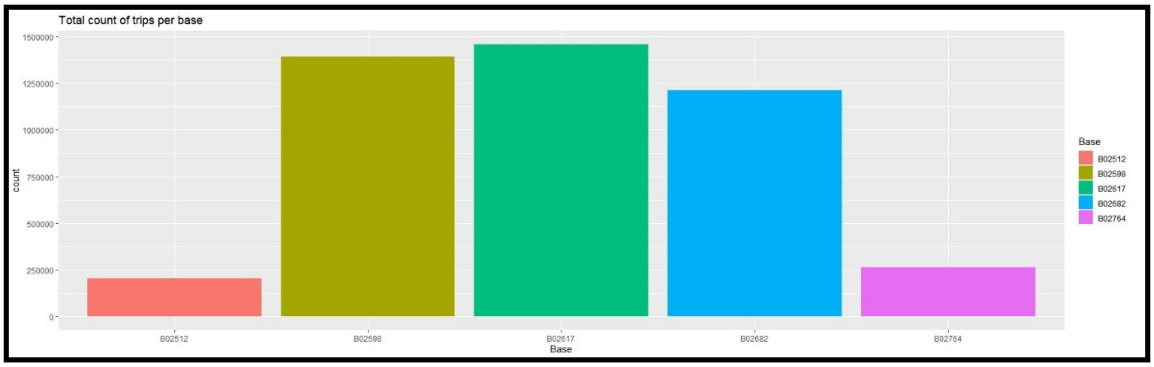

Fig. 7.15 Total count of trips per base

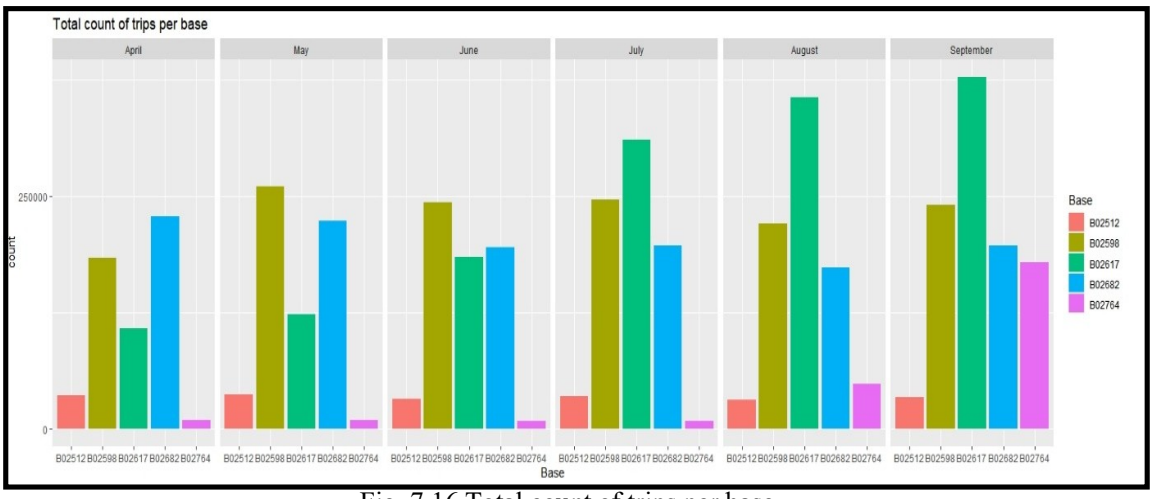

Fig. 7.16 Total count of trips per base

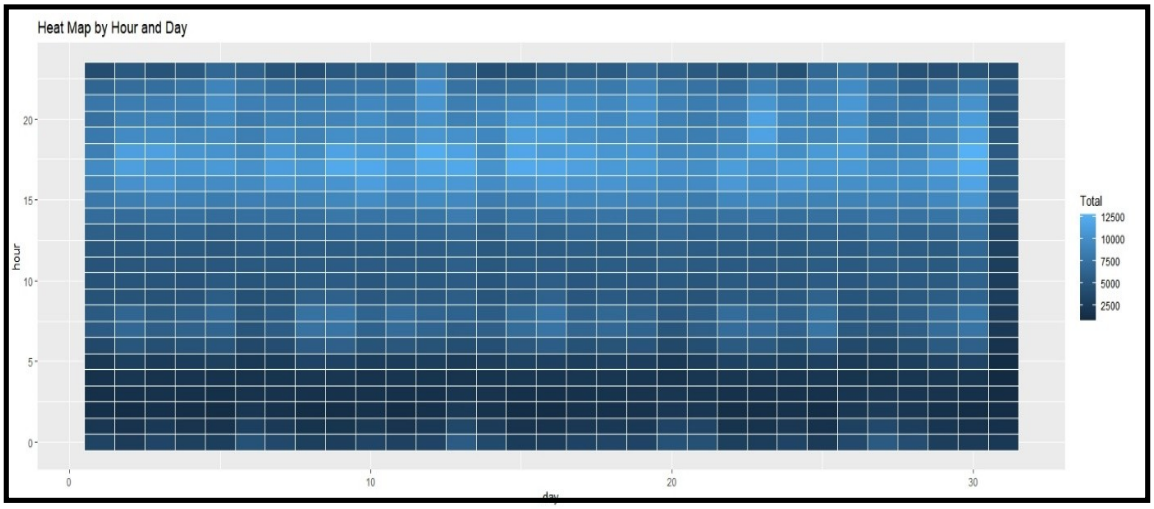

Fig. 7.17 Heat map by the hour and day

Conclusion

Big Data Analytics and Data Visualization bring life to a boring and very plain dataset are what was concluded from this project. The data represented is of very high value and the way it is represented is very visually appealing and thus can be understood even by a non-technical person. Even though there were various complex mathematical calculations for the data analytics part involved, R programming language eased the load is what can be learned. This project helped us understand the way visualization in R works and the various libraries and packages used for creating the visualized data. The data in the dataset which is visualized can anytime change and yet the same code can be anytime used for visualizing and analyzing it, thus guaranteeing consistency and stability. R programming language works wonders for data science.

References

[1] https://iedu.us/project-in-r-uber-data-analysis-project/

[2] https://www.tutorialspoint.com/r/r_data_frames.htm

[3] https://www.udemy.com/course/r-programming/

[4] https://github.com/bking2415/Uber-data-analysis

[5] https://rpubs.com/serena049/uberpickup1

[6] https://hdsr.mitpress.mit.edu/pub/zok97i7p/release/3