Project in R – Uber Data Analysis Project

LANGAT ERICK

Project in R – Uber Data Analysis Project

we will analyze the ***Uber Pickups in New York City dataset***. This is more of a data visualization project that will guide you towards using the ggplot2 library for understanding the data and for developing an intuition for understanding the customers who avail the trips.

Talking about our Uber data analysis project, data storytelling is an important component of ***Machine Learning*** through which companies are able to understand the background of various operations. With the help of visualization, companies can avail the benefit of understanding the complex data and gain insights that would help them to craft decisions. You will learn how to implement the ggplot2 on the Uber Pickups dataset and at the end, master the art of data visualization in R. Uber Dataset

IMPORT ESSENTIAL LIBRARIES

```
# library(bookdown)
library(tidyr)
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(ggthemes))
suppressPackageStartupMessages(library(scales))
suppressPackageStartupMessages(library(DT))
suppressPackageStartupMessages(require(recommenderlab))
require(lubridate)
```

Reading the Data into their designated variables

Now, we will read several csv files that contain the data from April 2014 to September 2014. We will store these in corresponding data frames like apr_data, may_data, etc. After we have read the files, we will combine all of this data into a single dataframe called 'data_2014'.

```
apr_data <-
read.csv("C:/Users/langa/OneDrive/Desktop/Dataset/UBERDATASET/uber-raw-apr
data.csv")
may_data <-
read.csv("C:/Users/langa/OneDrive/Desktop/Dataset/UBERDATASET/uber-raw-data-
may14.csv")
june_data <-
read.csv("C:/Users/langa/OneDrive/Desktop/Dataset/UBERDATASET/uber-raw-data-
jun14.csv")
july_data <-
read.csv("C:/Users/langa/OneDrive/Desktop/Dataset/UBERDATASET/uber-raw-data-
```

```
jul14.csv")
aug_data <-
read.csv("C:/Users/langa/OneDrive/Desktop/Dataset/UBERDATASET/uber-raw-data-
aug14.csv")
sep_data <-
read.csv("C:/Users/langa/OneDrive/Desktop/Dataset/UBERDATASET/uber-raw-data-
sep14.csv")

#bind all the data
data_2014 <- rbind(apr_data, may_data, june_data, july_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)

## Warning: 1959 failed to parse.

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)))

## Warning in .parse_hms(..., order = "HMS", quiet = quiet): Some strings
failed
## to parse, or all strings are NAs

data_2014$minute <- factor(minute(hms(data_2014$Time)))

## Warning in .parse_hms(..., order = "HMS", quiet = quiet): Some strings
failed
## to parse, or all strings are NAs

data_2014$second <- factor(second(hms(data_2014$Time)))

## Warning in .parse_hms(..., order = "HMS", quiet = quiet): Some strings
failed
## to parse, or all strings are NAs

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

## PhantomJS not found. You can install it with webshot::install_phantomjs().
If it is installed, please make sure the phantomjs executable can be found
via the PATH variable.
```

Show 10 entries

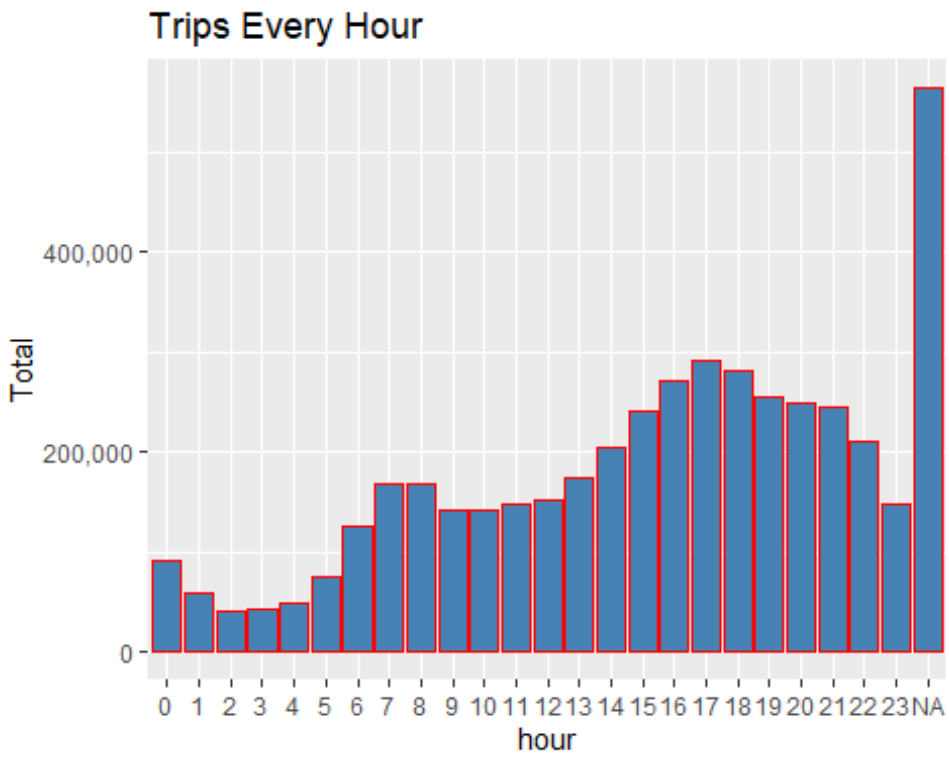
Search:

	hour	Total
1	0	91926
2	1	59458
3	2	40930
4	3	43247
5	4	49135
6	5	74463
7	6	124715
8	7	168170
9	8	167661
10	9	142028

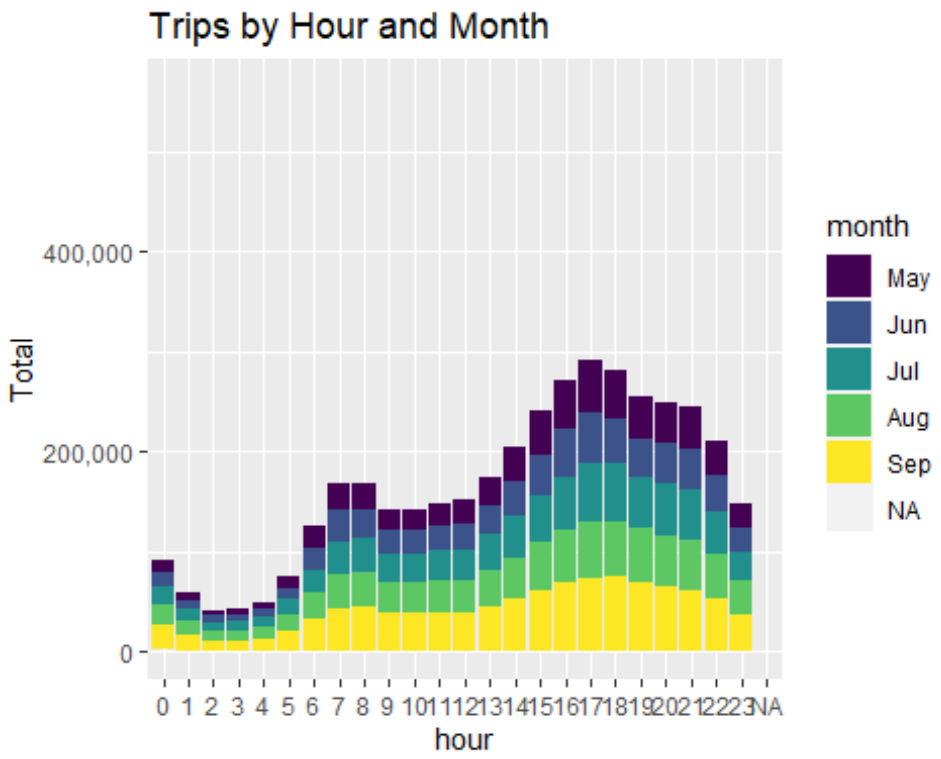
Showing 1 to 10 of 25 entries

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)
```




```
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)
```



Plotting data by trips during every day of the month

In this section of R project, we will learn how to plot our data based on every day of the month. We observe from the resulting visualization that 30th of the month had the highest trips in the year which is mostly contributed by the month of April.

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

Show 10 entries

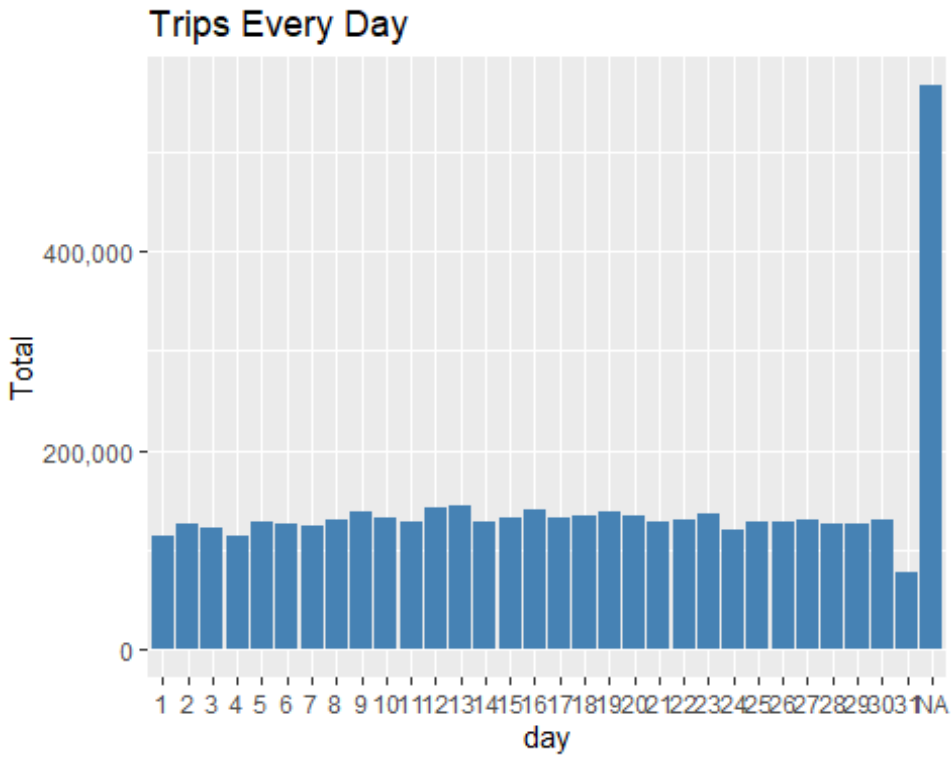
Search:

	day	Total
1	1	112817
2	2	125677
3	3	122217
4	4	114162
5	5	127464
6	6	126385
7	7	123875
8	8	129744
9	9	138240
10	10	132399

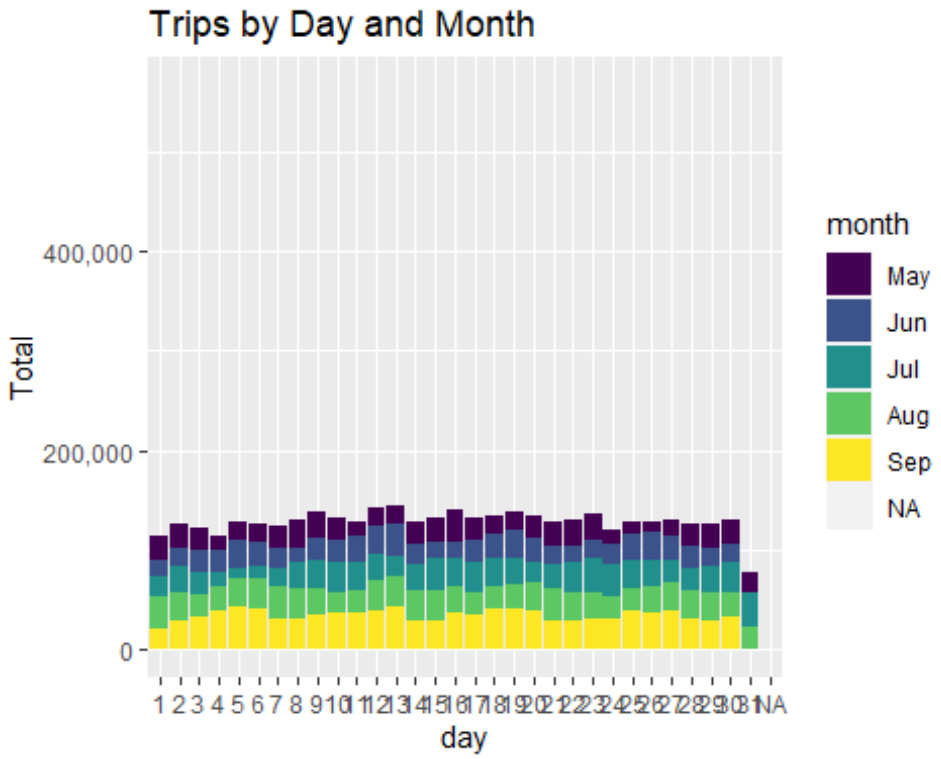
Showing 1 to 10 of 32 entries

Previous 1 2 3 4 Next

```
ggplot(day_group, aes(day, Total)) +
        geom_bar( stat = "identity", fill = "steelblue") +
           ggtitle("Trips Every Day") +
            theme(legend.position = "none") +
            scale_y_continuous(labels = comma)
```




```
day_month_group <- data_2014 %>%
         group_by(month, day) %>%
             dplyr::summarize(Total = n())

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

ggplot(day_month_group, aes(day, Total, fill = month)) +
        geom_bar( stat = "identity") +
           ggtitle("Trips by Day and Month") +
            scale_y_continuous(labels = comma)
```




```
            # scale_fill_manual(values = colors)
```

Number of Trips taking place during months in a year

In this section, we will visualize the number of trips that are taking place each month of the year. In the output visualization, we observe that most trips were made during the month of September. Furthermore, we also obtain visual reports of the number of trips that were made on every day of the week.

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

Show 10 entries

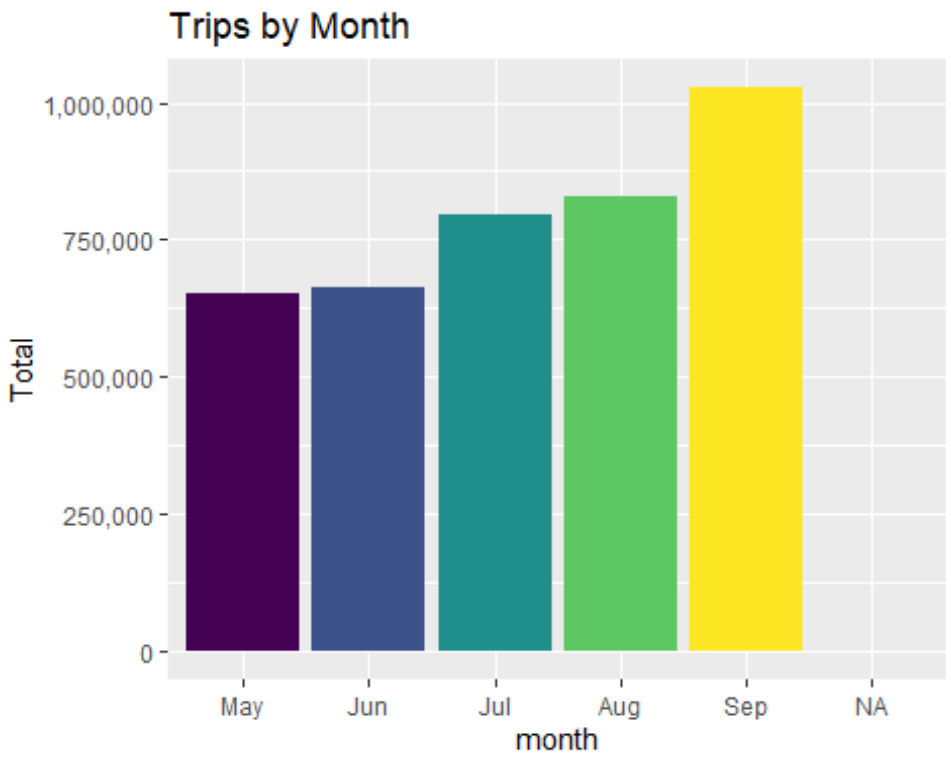
Search:

	month	Total
1	May	652124
2	Jun	663545
3	Jul	795732
4	Aug	828805
5	Sep	1027646
6		566475

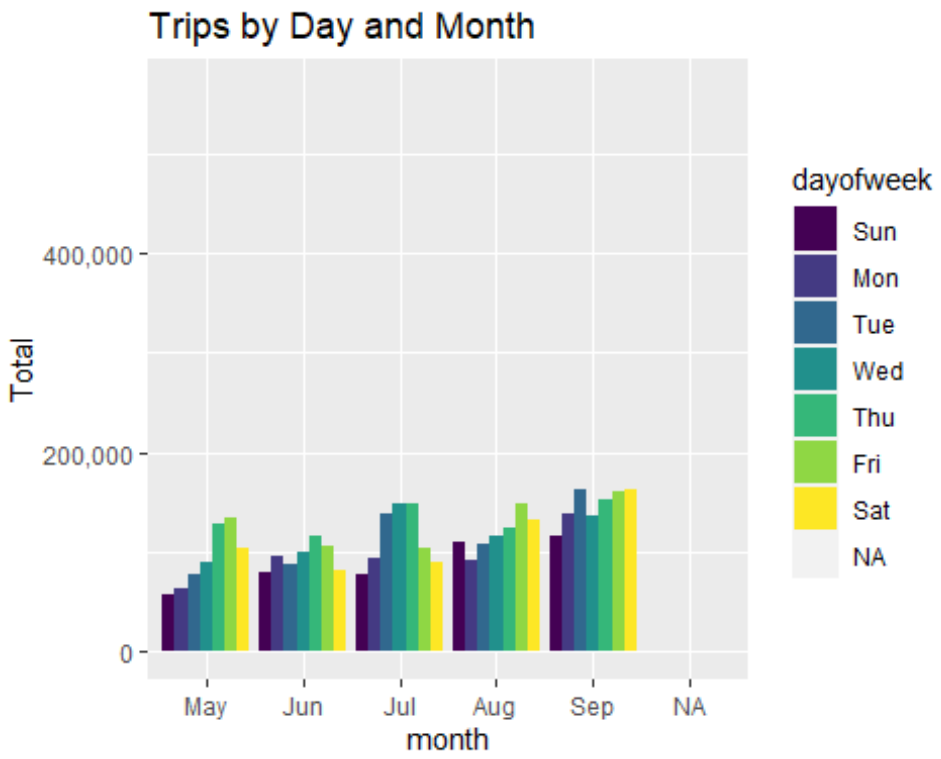
Showing 1 to 6 of 6 entries

Previous 1 Next

```
ggplot(month_group, aes(month, Total,fill = month)) +
   geom_bar(stat = 'identity') +
  scale_y_continuous(labels=comma)+
   ggtitle("Trips by Month") +
   theme(legend.position = "none")
```




```
month_weekday <- data_2014 %>%
         group_by(month, dayofweek) %>%
             dplyr::summarize(Total = n())

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

ggplot(month_weekday, aes(month, Total, fill = dayofweek)) +
       geom_bar( stat = "identity", position = "dodge") +
          ggtitle("Trips by Day and Month") +
           scale_y_continuous(labels = comma) #+
```

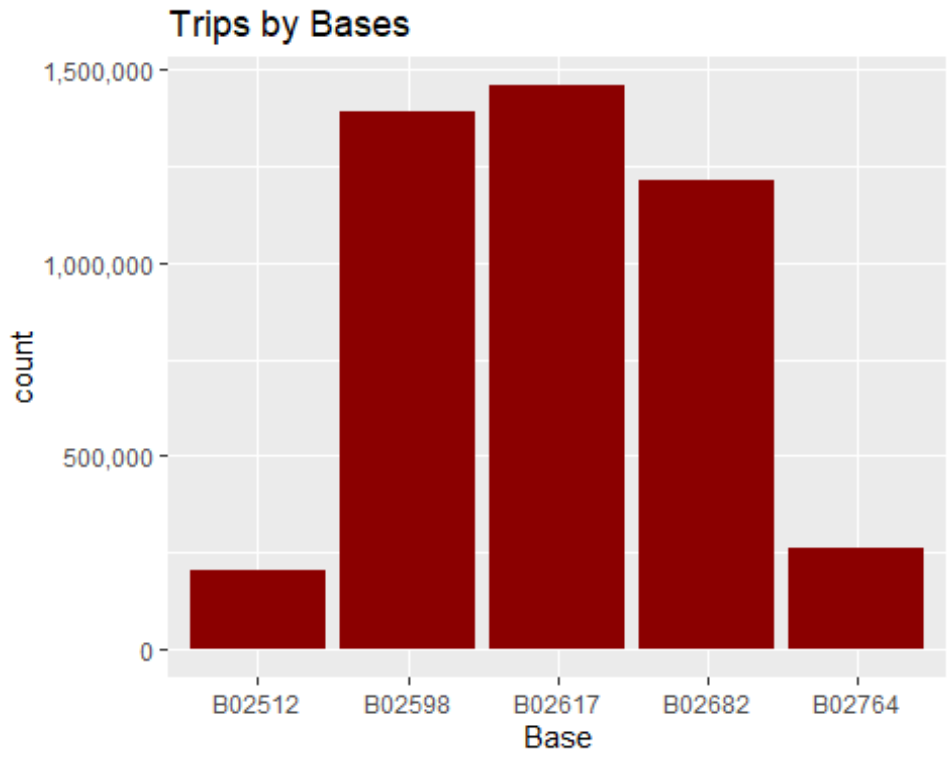



```
            #scale_fill_manual(values = colors)
```

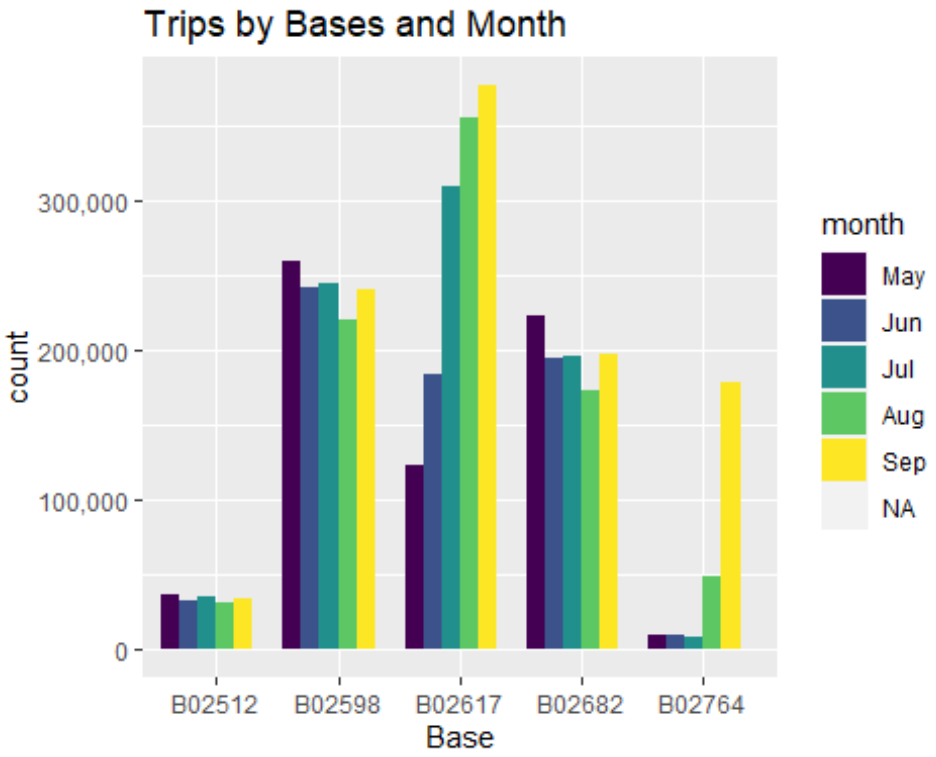
Finding out the number of Trips by bases

In the following visualization, we plot the number of trips that have been taken by the passengers from each of the bases. There are five bases in all out of which, we observe that B02617 had the highest number of trips. Furthermore, this base had the highest number of trips in the month B02617. Thursday observed highest trips in the three bases – B02598, B02617, B02682.

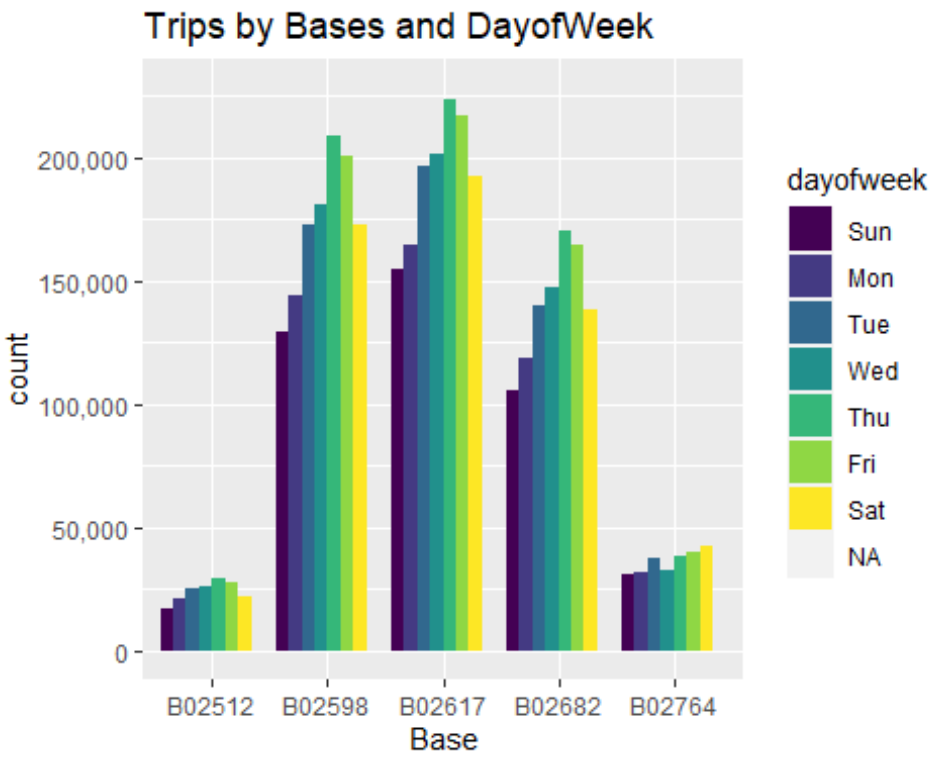
```
ggplot(data_2014, aes(Base)) +
 geom_bar(fill = "darkred") +
 scale_y_continuous(labels = comma) +
 ggtitle("Trips by Bases")
```




```
ggplot(data_2014, aes(Base, fill = month)) +
 geom_bar(position = "dodge") +
 scale_y_continuous(labels = comma) +
 ggtitle("Trips by Bases and Month") #+
```




```
 # scale_fill_manual(values = colors)

ggplot(data_2014, aes(Base, fill = dayofweek)) +
 geom_bar(position = "dodge") +
 scale_y_continuous(labels = comma) +
 ggtitle("Trips by Bases and DayofWeek")
```



Creating a Heatmap visualization of day, hour and month

In this section, we will learn how to plot heatmaps using ggplot(). We will plot five heatmap plots –

- First, we will plot Heatmap by Hour and Day.
- Second, we will plot Heatmap by Month and Day.
- Third, a Heatmap by Month and Day of the Week.
- Fourth, a Heatmap that delineates Month and Bases.

Finally, we will plot the heatmap, by bases and day of the week.

```
day_and_hour <- data_2014 %>%
         group_by(day, hour) %>%
            dplyr::summarize(Total = n())

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

datatable(day_and_hour)
```

Show 10 entries

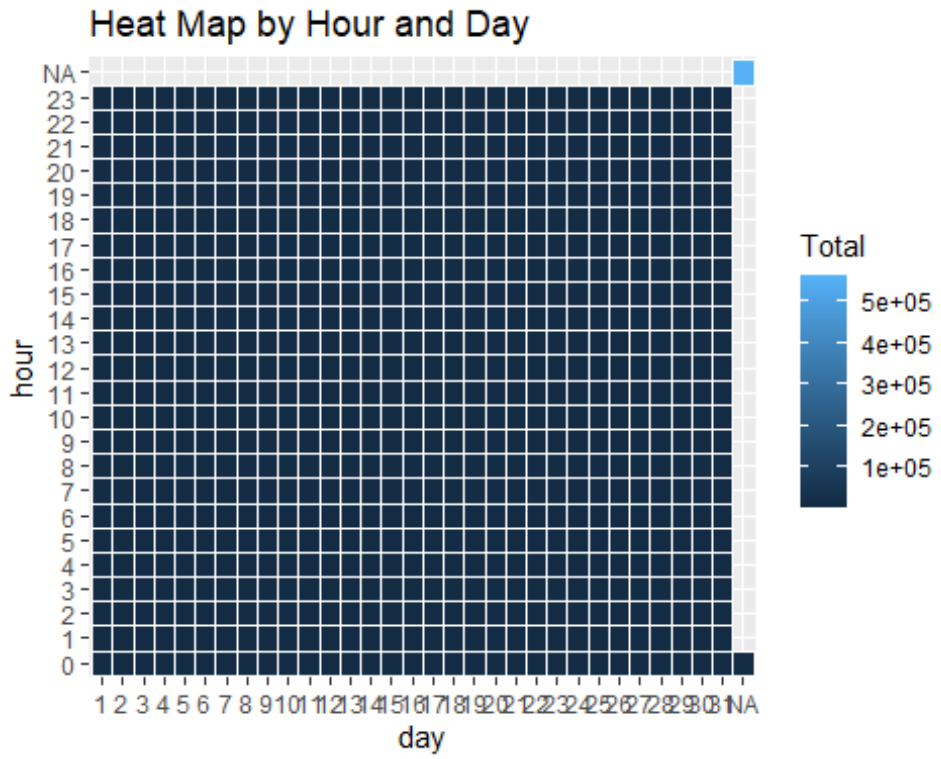
Search:

	day	hour	Total
1	1	0	3042
2	1	1	1916
3	1	2	1231
4	1	3	1238
5	1	4	1292
6	1	5	1838
7	1	6	2995
8	1	7	4469
9	1	8	4596
10	1	9	4156

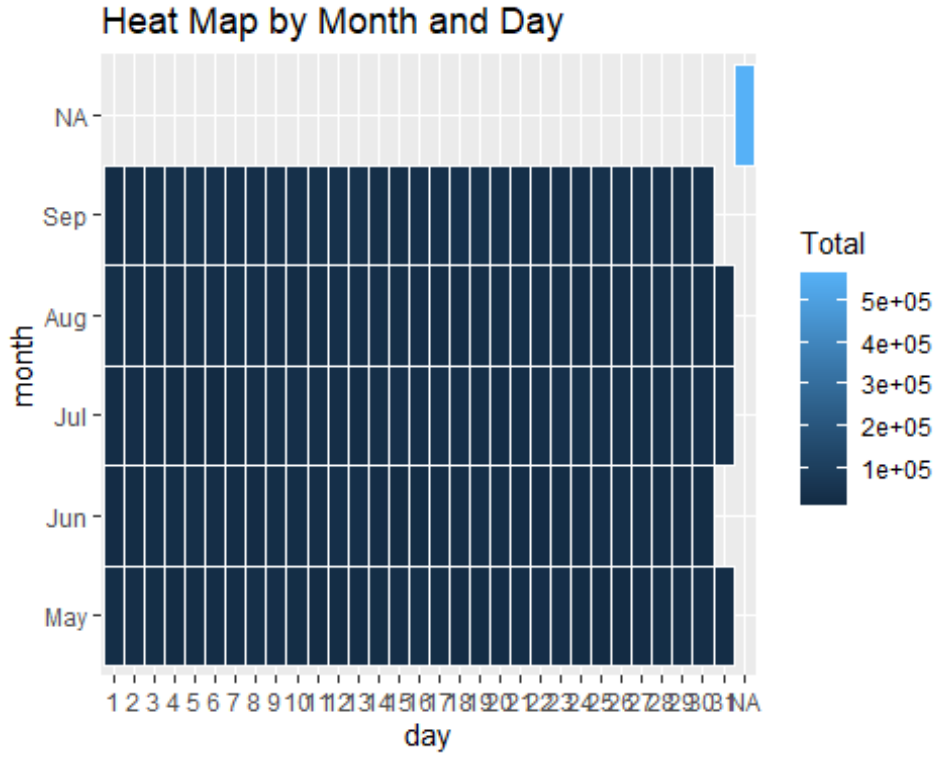
Showing 1 to 10 of 746 entries

Previous 1 2 3 4 5 … 75 Next

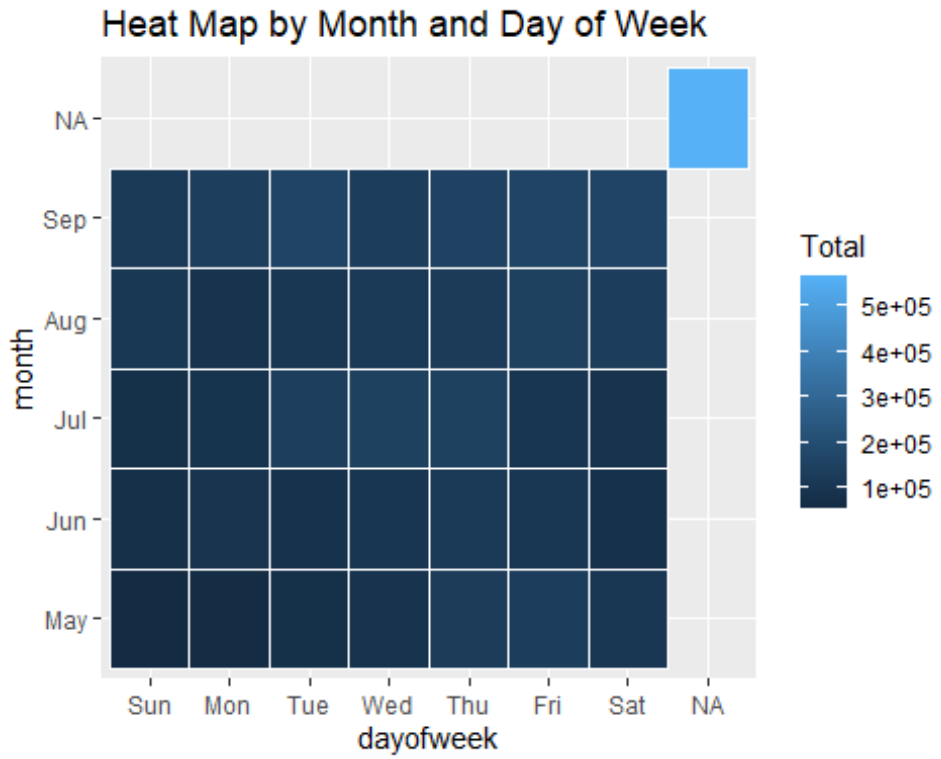
```
ggplot(day_and_hour, aes(day, hour, fill = Total)) +
            geom_tile(color = "white") +
              ggtitle("Heat Map by Hour and Day")
```



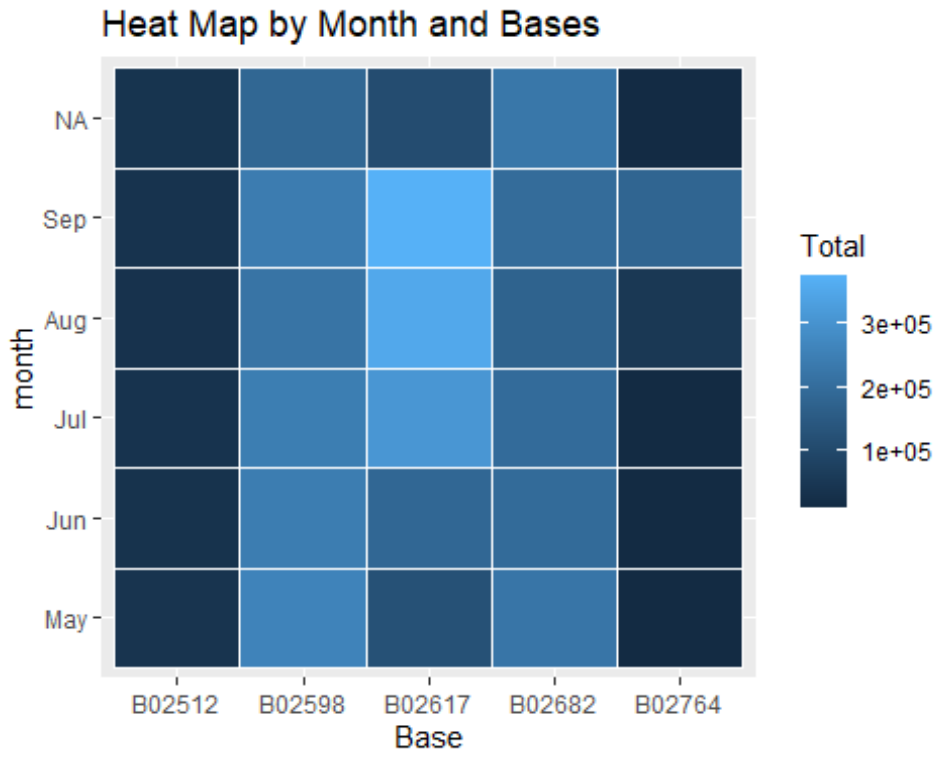

```
ggplot(day_month_group, aes(day, month, fill = Total)) +
            geom_tile(color = "white") +
              ggtitle("Heat Map by Month and Day")
```



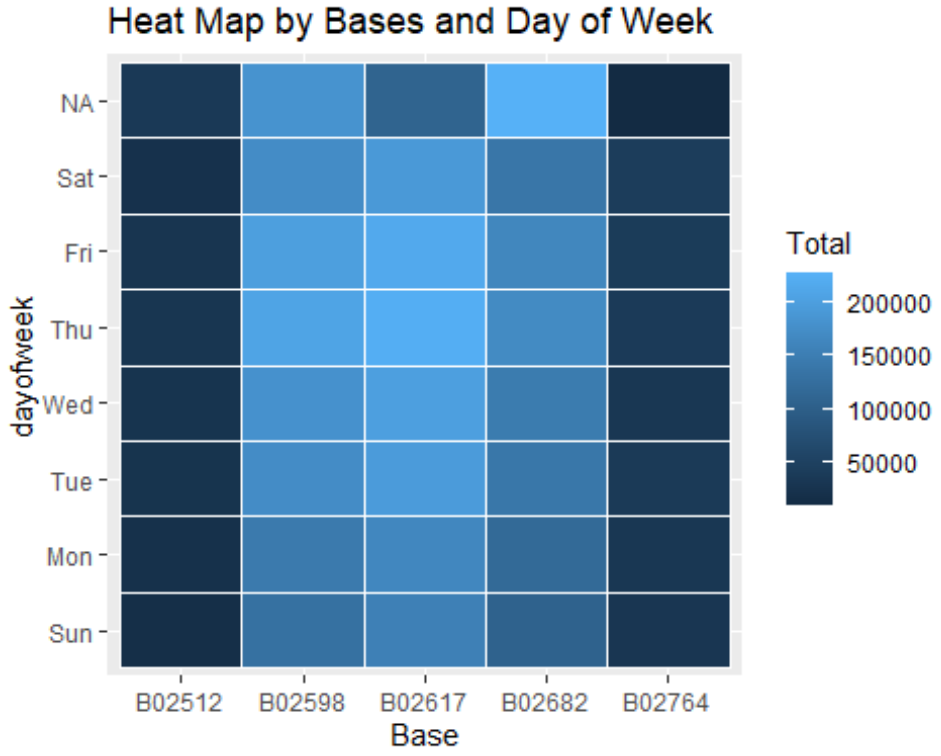
```
ggplot(month_weekday, aes(dayofweek, month, fill = Total)) +
            geom_tile(color = "white") +
              ggtitle("Heat Map by Month and Day of Week")
```




```
month_base <-  data_2014 %>%
                    group_by(Base, month) %>%
                        dplyr::summarize(Total = n())

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

day0fweek_bases <-  data_2014 %>%
                    group_by(Base, dayofweek) %>%
                        dplyr::summarize(Total = n())

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

ggplot(month_base, aes(Base, month, fill = Total)) +
            geom_tile(color = "white") +
              ggtitle("Heat Map by Month and Bases")
```



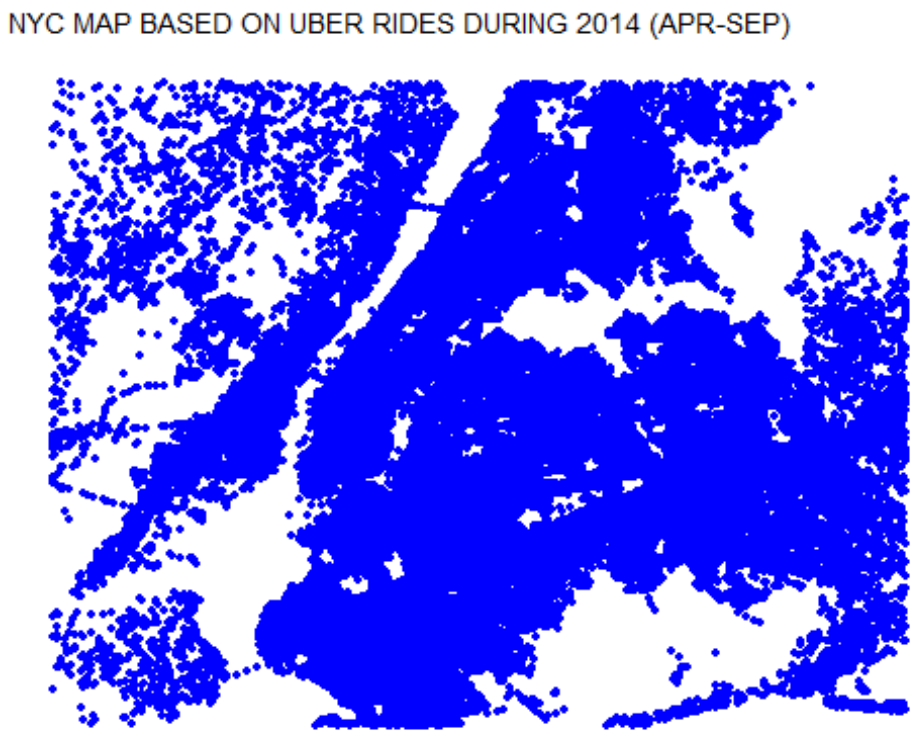

```
ggplot(day0fweek_bases, aes(Base, dayofweek, fill = Total)) +
            geom_tile(color = "white") +
              ggtitle("Heat Map by Bases and Day of Week")
```



Creating a map visualization of rides in New York

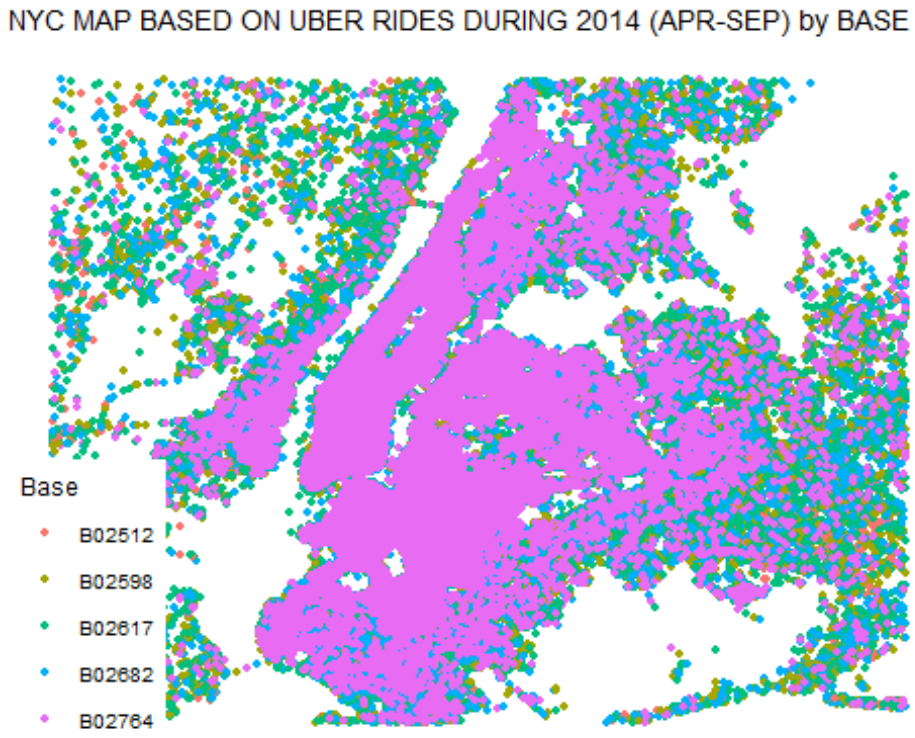
In the final section, we will visualize the rides in New York city by creating a geo-plot that will help us to visualize the rides during 2014 (Apr – Sep) and by the bases in the same period.

```
min_lat <- 40.5774
max_lat <- 40.9176
min_long <- -74.15
max_long <- -73.7004

ggplot(data_2014, aes(x=Lon, y=Lat)) +
  geom_point(size=1, color = "blue") +
     scale_x_continuous(limits=c(min_long, max_long)) +
      scale_y_continuous(limits=c(min_lat, max_lat)) +
        theme_map() +
           ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP)")

## Warning: Removed 71701 rows containing missing values (`geom_point()`).
```




```
ggplot(data_2014, aes(x=Lon, y=Lat, color = Base)) +
  geom_point(size=1) +
     scale_x_continuous(limits=c(min_long, max_long)) +
      scale_y_continuous(limits=c(min_lat, max_lat)) +
       theme_map() +
          ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP) by
BASE")
```

```
## Warning: Removed 71701 rows containing missing values (`geom_point()`).
```



Summary

At the end of the **Uber data analysis R project**, we observed how to create data visualizations. We made use of packages like ggplot2 that allowed us to plot various types of visualizations that pertained to several time-frames of the year. With this, we could conclude how time affected customer trips. Finally, we made a geom plot of New York that provided us with the details of how various users made trips from different bases.