TITLE OF THE PROJECT

UBER DATA ANALYSIS IN NEW YORK CITY

A Project Report

to be submitted by:

DASARI SHIVA-20BCE7075

DASARI SRIKANTH-20BCE7076

VIT-AP UNIVERSITY

AMARAVATHI

NOVEMBER-2022

DATA ANALYTICS PROJECT

UBER DATA ANALYSIS

Abstract:

Data Analytics has assisted many organizations with enhancing and develop their exhibition for the many years. Information examination and perception has helped us with a few advantages, not many of them being recognizing arising patterns, concentrating on connections and examples in information, examination top to bottom. This project is all about understanding one such data set of uber from New York City and is very component to understand the use of data analytics and visualization. It is produced with the assistance of 'R' programming language utilizing libraries and packages, for example, ggplot2, lubridate, dplyr and tidyr. Through projects like this, we can gain knowledge of various complex operations performed in data visualization. It will empower us to perceive the examples in information of this gigantic association and gives basic bits of knowledge of undiscovered data. Additionally guide us in grasping the activities of various R libraries.

Key words— Uber, Data analytics, Data visualization, R programming, ggplot, lubridate, dplyr, tidyr, DT, scales.

Packages:

1.ggplot2

2.ggthemes

3.lubridate

4.dplyr

5.tidyr

6.DT

7.scales

Problem Statement:

Uber is an application-based transportation organization and taxi organization. In its many rides in a specific city, large numbers of its users face the problem of cancellation by the driver or non-availability of cars.

These issues influence the matter of Uber and it misses out on its income.

INTRODUCTION:

In the modern world, Uber has emerged as the leading organisation for setting up new transportation options. People use analytics in their businesses to help it grow, and this market is one that is rapidly expanding. This project will improve our understanding of how to use the ggplot2 library to comprehend the data and to acquire an intuition for comprehending the consumers who benefit from the travels.

The key to solving this problem is to comprehend customer segmentation. Customer segmentation can be compared to a child's activity of sorting balls and cubes based on their colour or shape. Customer segmentation is, to put it simply, the process of separating customers from the market according to numerous criteria and categorising them according to different traits. Although the Uber data is not as granular as the taxi data—unusually, Uber only offers time and location for pickups, not drop-offs—I nevertheless wanted to offer a dataset that combined all of the taxi and Uber information that was currently available. Uber analyses historical data for, say, the past three or four weeks and finds areas of the city where demand is particularly strong.

To grow business with this competitive environment data analysis is necessary. Data analysis reports, and other kinds of analysis and report documents must be developed by businesses so that they can have references for peculiar activities and undertakings especially when making decisions for the future operations of the company. The Excel files with the weather data and Uber pick-up data should be joined together for the analysis. A data analysis can be developed accordingly if you can arrange all the information based on the activity that you will undergo.

The only transportation service to analyse and communicate actual sustainability statistics is Uber. Our goal in this R project is to analyse the dataset for Uber pickups in New York City. This project focuses primarily on data visualisation and will teach us how to use the ggplot2 library to understand the data and develop an accurate understanding of the travellers.

Literature review:

Ggplot2 is currently more than 10 years of age and is utilized by 100's of 1000's of individuals to make a huge number of plots.

It is an R package dedicated to data visualization. It can significantly improve the quality and feel of your illustrations, and will make you substantially more efficient in making them. Ggplot2 allows building almost any type of chart. It is a framework for revelatory making designs, in light of the syntax of illustrations. You give the information, tell ggplot2 how to plan factors to style, what graphical primitives to use, and it takes care of the details.

Our main objectives of this project are:

- Visualize the growth of Uber in NYC;
- Analyse the demand based on trends in the time series;
- and calculate the market size for Uber in NYC.

- Additional information on how the service is used;

- An effort to estimate the rise in demand.

Description of Dataset:

The dataset contains information about Uber pickups in New York City from April 2014 to September 2014. It has over 500k pickups (rows) and the following 4 columns:

Date/Time - The date and time of pickup

Lat - Latitude of pickup

Long - Longitude of pickup Base

Methodology:

Initial Analysis: Understand the dataset given. Look through its structure, identify the datatypes of various columns and get a basic idea of the dataset to proceed further.

Data Cleaning: Look out for visible data quality issues and rectify them. Check for blanks, duplicate data and convert certain columns to required datatypes.

Exploratory Data Analysis: Making use of R, carry out various EDA operations like Univariate and Segmented Univariate analysis and come up with intuitive insights of the Supply-Demand problem.

Visualization: Make use of R libraries & package to plot various graphs with proper aesthetics and geometry, clearly displaying important insights.

Proposed system:

We proposed that we will build a data visualization project with ggplot2 using R and its libraries. Analyse various parameters like

(a) Trips by the hours in a day’

(b) Trips during months in a year.

At the end create visualizations for different timeframes of the year. Explain how time affects customer trips.

• Find the days on which each basement has a greater number of active vehicles.

• Can tap growing markets in suburban areas where taxi services are not available.

• The estimated time of arrival can be shortened with an increase in Uber drivers, which would gradually raise customer satisfaction and increase both the company's income and drivers' profits. The most popular travel destinations, as determined by the number of booked trips, will be identified based on the data.

1.Importing the Essential Packages

In the first step of our R project, we will import the essential packages that we will use in this uber data analysis project. Some of the ***important libraries of R*** that we will use are –

- **ggplot2**

This is the backbone of this project. ggplot2 is the most popular data visualization library that is most widely used for creating aesthetic visualization plots.

- **ggthemes**

This is more of an add-on to our main ggplot2 library. With this, we can create better create extra themes and scales with the mainstream ggplot2 package.

- **dplyr**

This package is the lingua franca of data manipulation in R.

- **tidyr**

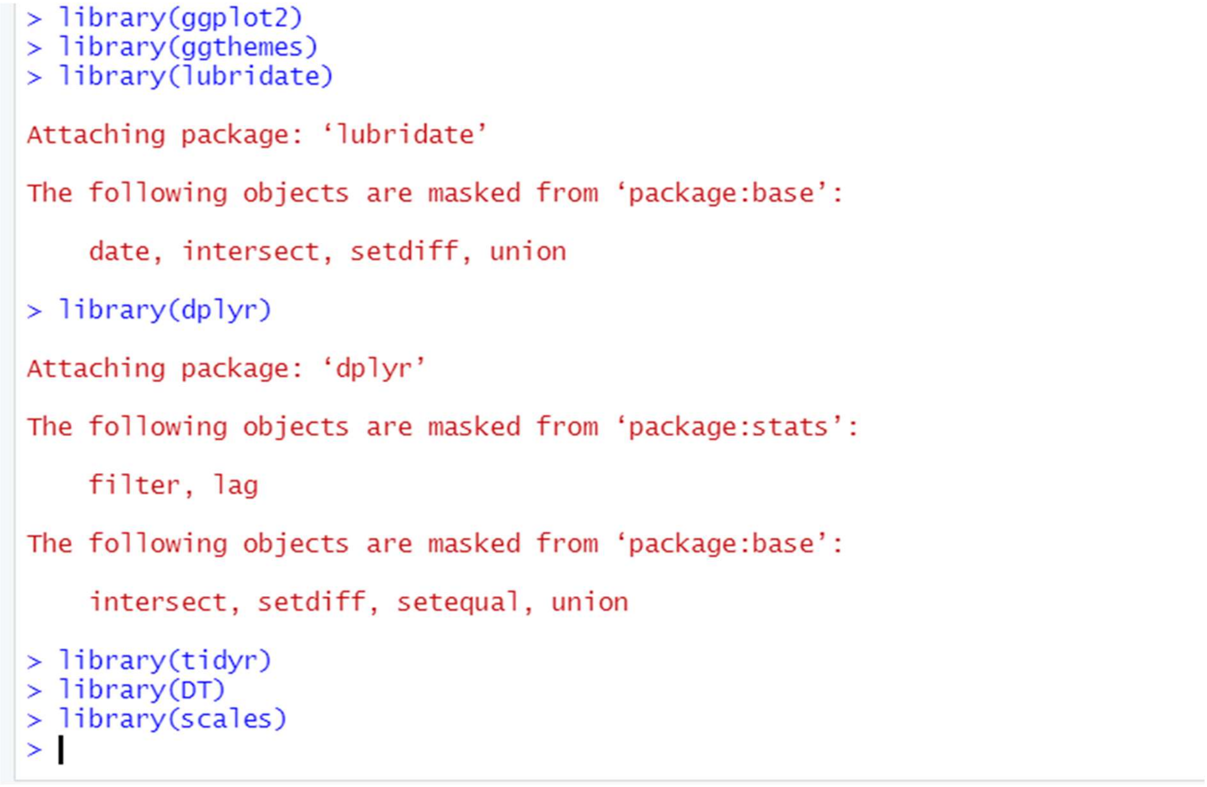
This package will help you to tidy your data. The basic principle of tidyr is to tidy the columns where each variable is present in a column, each observation is represented by a row and each value depicts a cell.

- **DT**

With the help of this package, we will be able to interface with the ***javascript*** Library called – Data tables.

- **Scales**

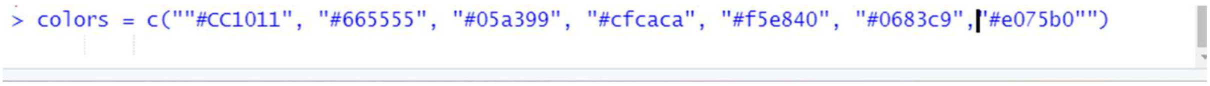
With the help of graphical scales, we can automatically map the data to the correct scales with well-placed axes and legends. (They take your data and turn it into something that you can see, like size, colour, position or shape.)



```
> library(ggplot2)
> library(ggthemes)
> library(lubridate)

Attaching package: ‘lubridate’

The following objects are masked from ‘package:base’:

    date, intersect, setdiff, union

> library(dplyr)

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)
> library(DT)
> library(scales)
> 
```

2. Creating vector of colours to be implemented in our plots

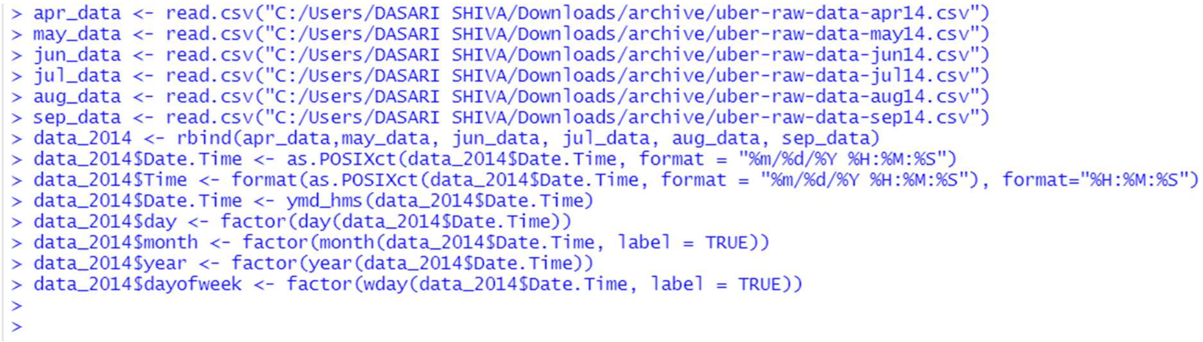
we will create a vector of our colours that are in our plotting functions. You can also select your own colours that you want.



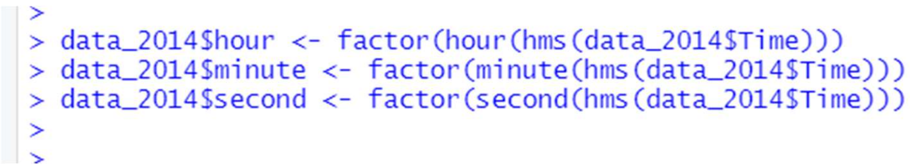
```
> colors = c(""#CC1011", "#665555", "#05a399", "#cfcaca", "#f5e840", "#0683c9","#e075b0"")
```

3. Reading the Data into their designated variables

we will read the 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 data frame called ‘data_2014’.



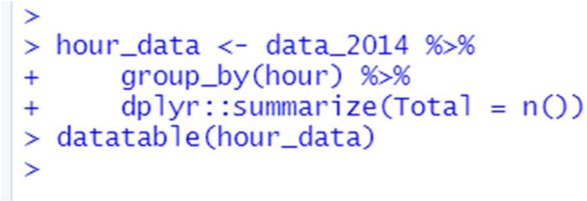
```
> apr_data <- read.csv("C:/Users/DASARI SHIVA/Downloads/archive/uber-raw-data-apr14.csv")
> may_data <- read.csv("C:/Users/DASARI SHIVA/Downloads/archive/uber-raw-data-may14.csv")
> jun_data <- read.csv("C:/Users/DASARI SHIVA/Downloads/archive/uber-raw-data-jun14.csv")
> jul_data <- read.csv("C:/Users/DASARI SHIVA/Downloads/archive/uber-raw-data-jul14.csv")
> aug_data <- read.csv("C:/Users/DASARI SHIVA/Downloads/archive/uber-raw-data-aug14.csv")
> sep_data <- read.csv("C:/Users/DASARI SHIVA/Downloads/archive/uber-raw-data-sep14.csv")
> 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))
>
>
```



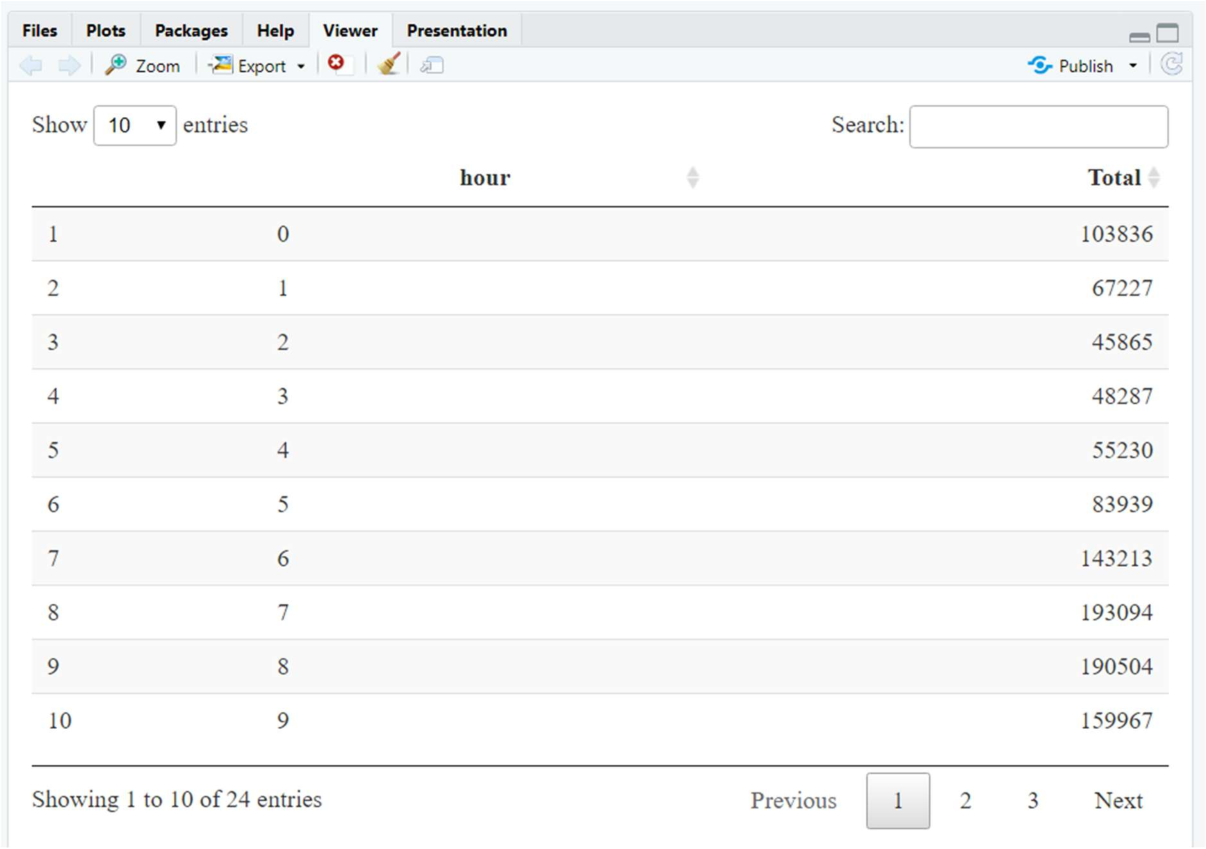
```
>
> data_2014$hour <- factor(hour(hms(data_2014$Time)))
> data_2014$minute <- factor(minute(hms(data_2014$Time)))
> data_2014$second <- factor(second(hms(data_2014$Time)))
>
>
```

Plotting the trips by the hours in a day

Here we will be using the ggplot function to plot the number of trips that the passengers had made in a day. We will also use dplyr to aggregate our data. In the resulting visualizations, we can understand how the number of passengers fares throughout the day. We observe that the number of trips are higher in the evening around **5:00 and 6:00 PM.**



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



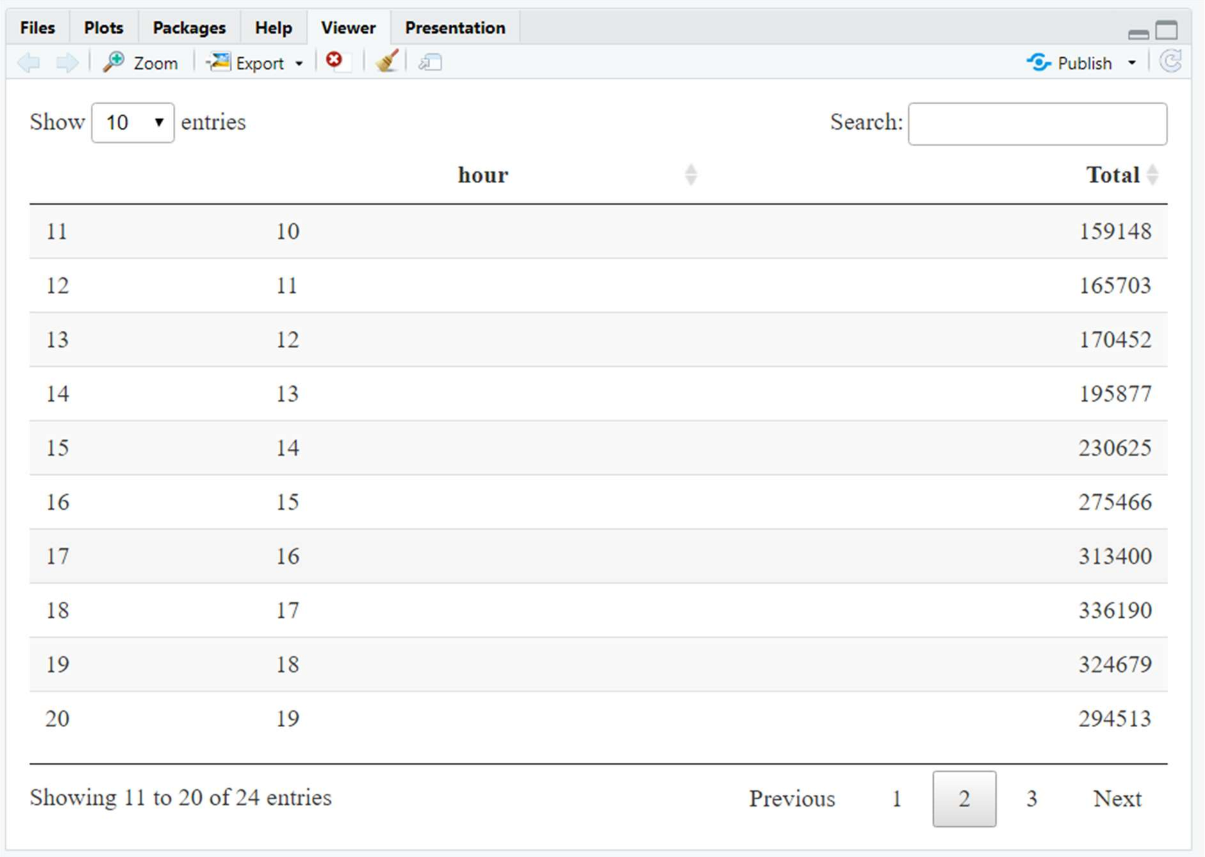
Show 10 entries

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

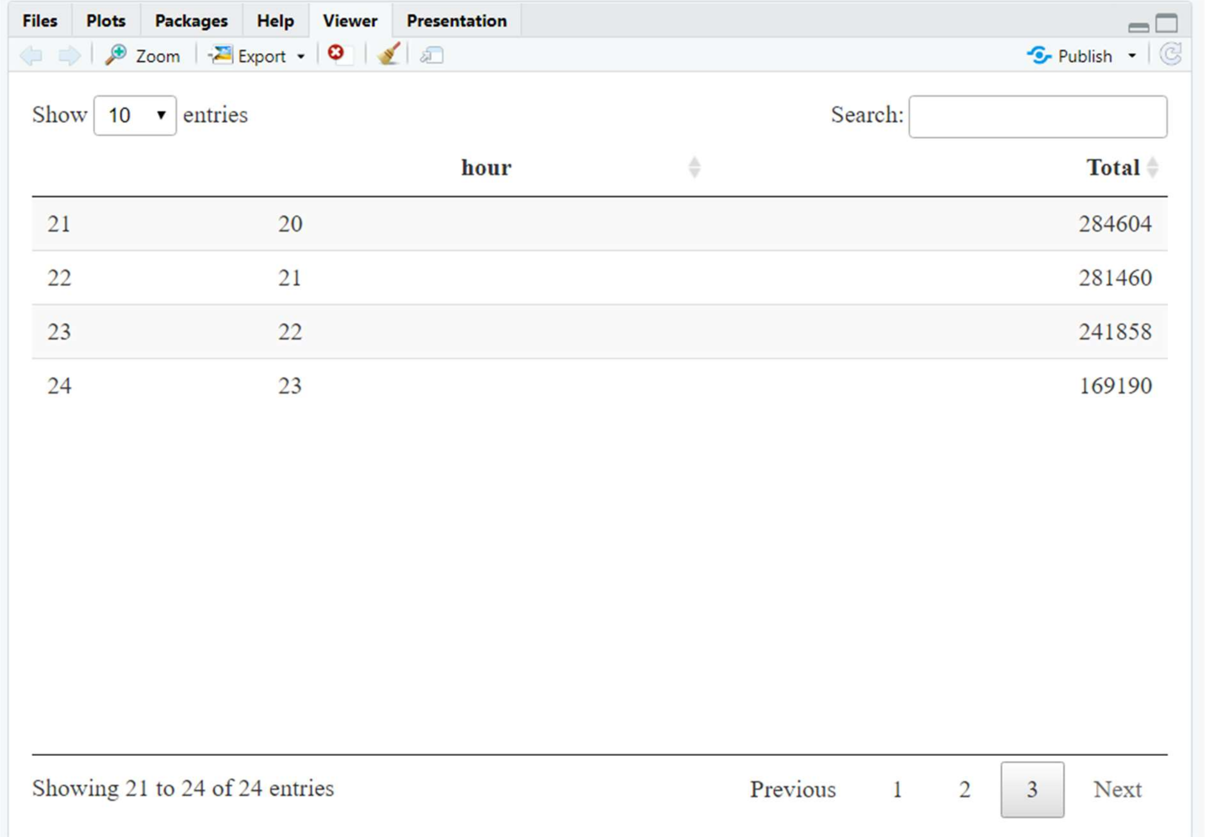
Showing 1 to 10 of 24 entries

Previous 1 2 3 Next



	hour	Total
11	10	159148
12	11	165703
13	12	170452
14	13	195877
15	14	230625
16	15	275466
17	16	313400
18	17	336190
19	18	324679
20	19	294513

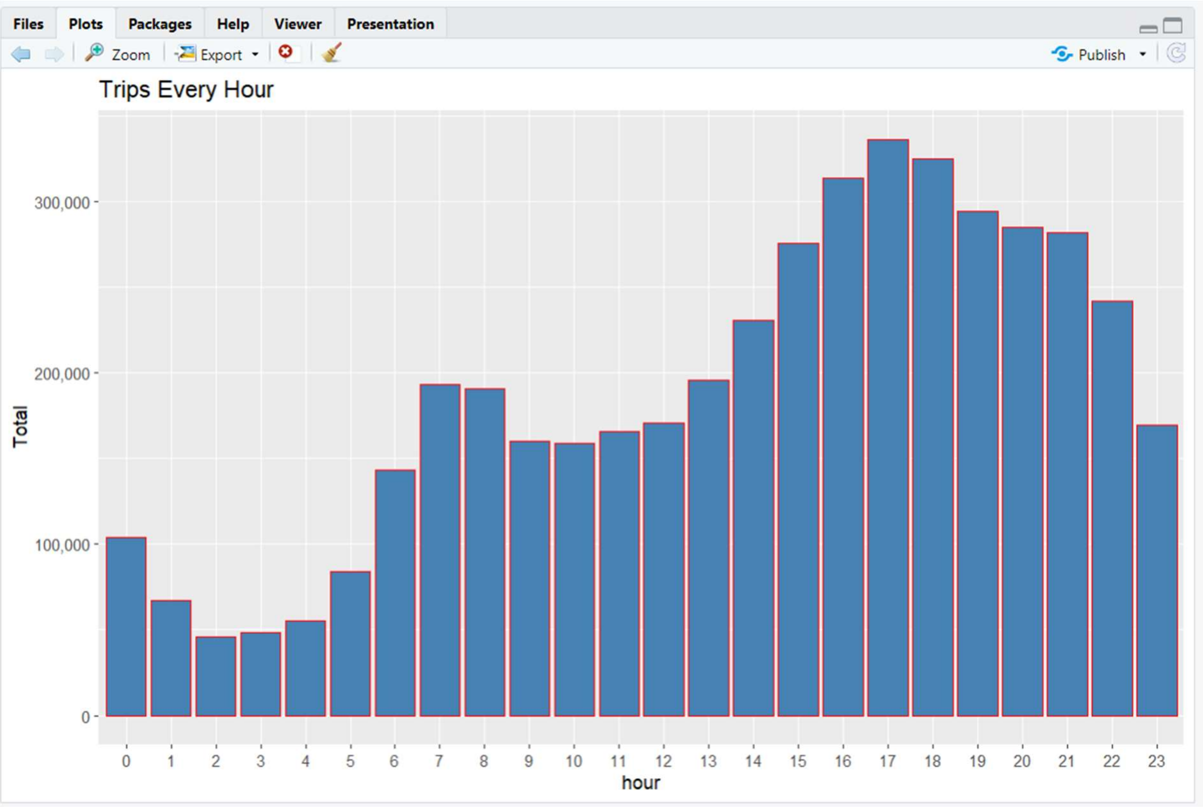
Showing 11 to 20 of 24 entries



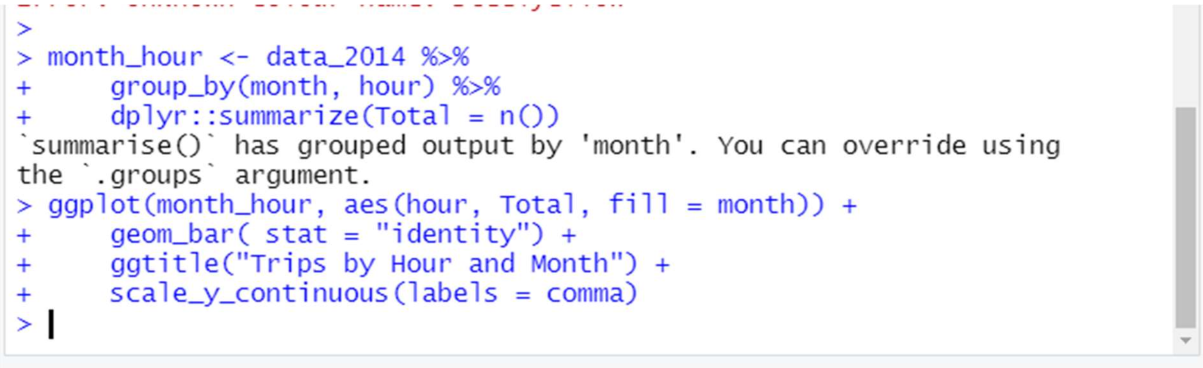
	hour	Total
21	20	284604
22	21	281460
23	22	241858
24	23	169190

Showing 21 to 24 of 24 entries

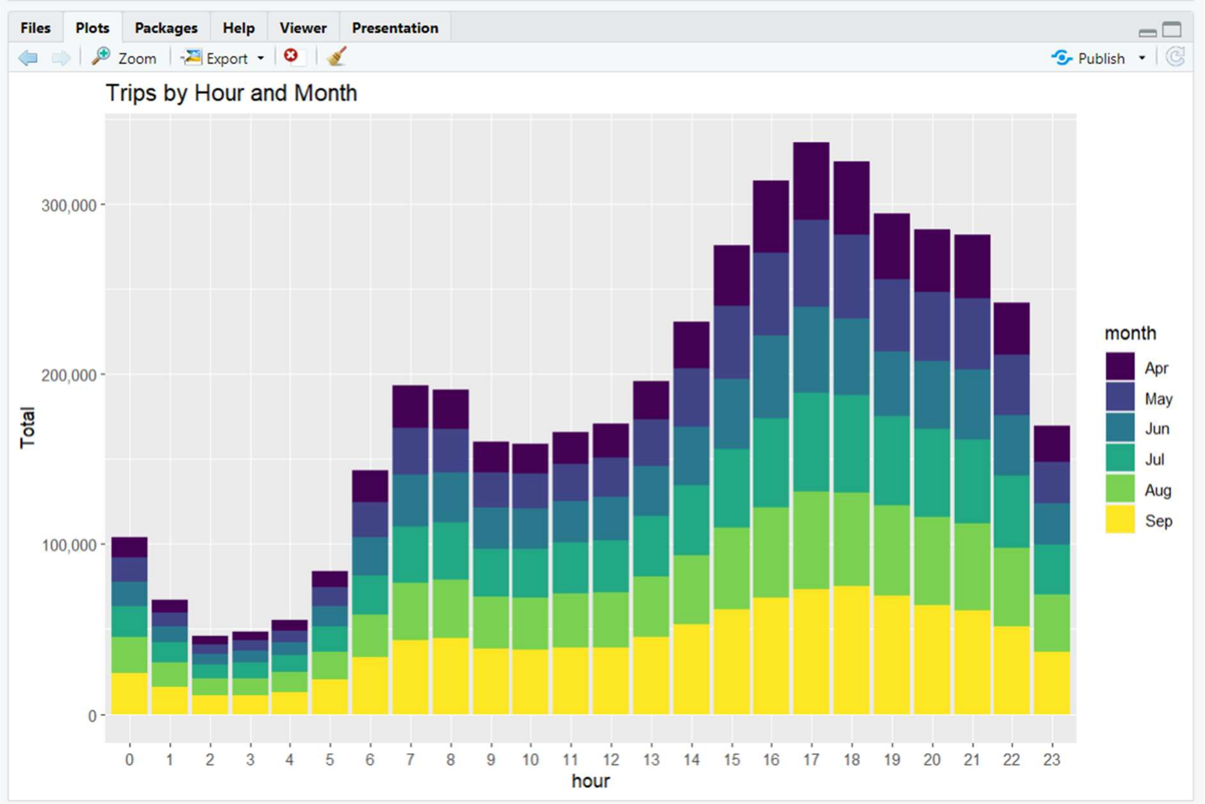
Trips by Every hour:



Trips by Every hour and month:

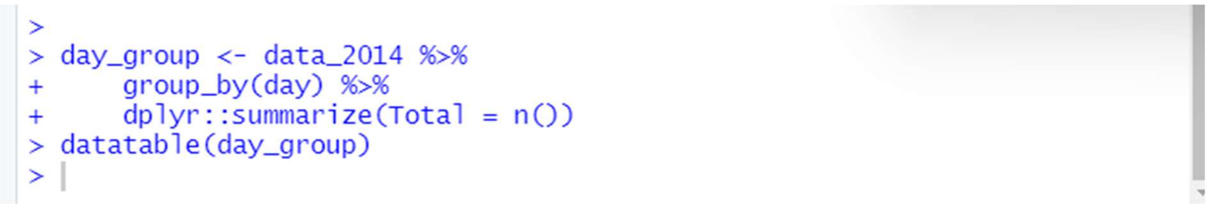


```
>
> month_hour <- data_2014 %>%
+     group_by(month, hour) %>%
+     dplyr::summarize(Total = n())
`summarise()` has grouped output by 'month'. You can override using
the `.groups` argument.
> ggplot(month_hour, aes(hour, Total, fill = month)) +
+     geom_bar( stat = "identity") +
+     ggtitle("Trips by Hour and Month") +
+     scale_y_continuous(labels = comma)
> 
```

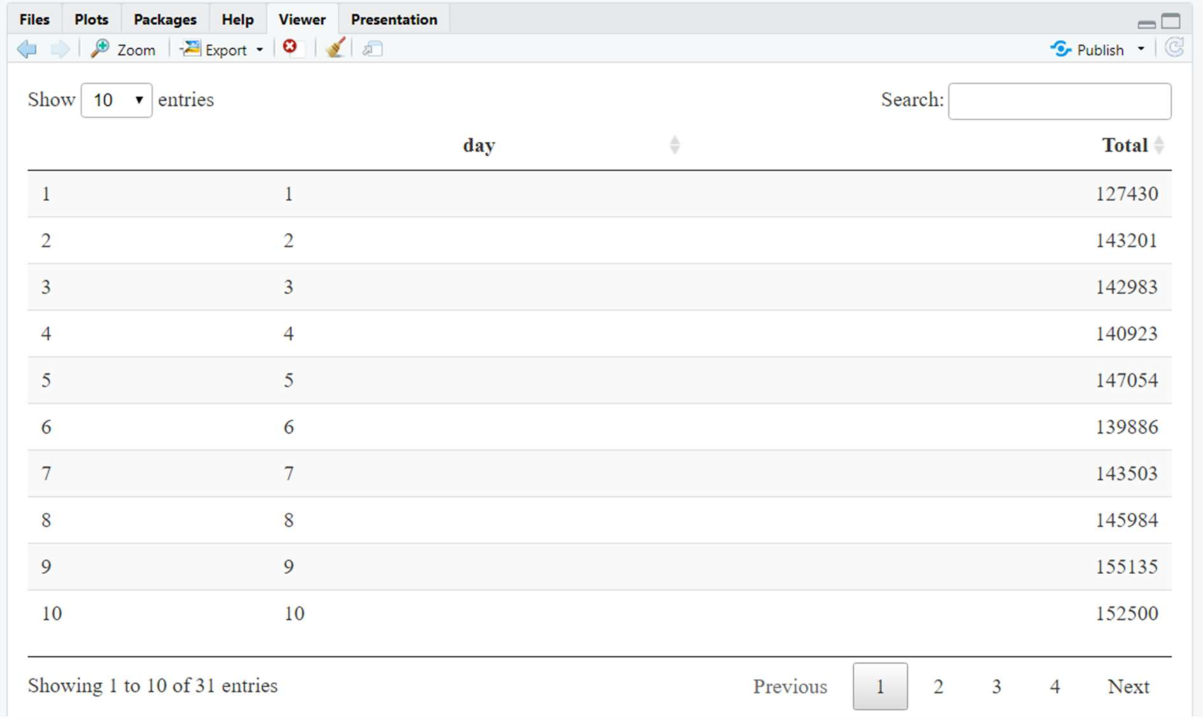


Plotting data by trips during every day of the month

Here, we will be plotting 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.



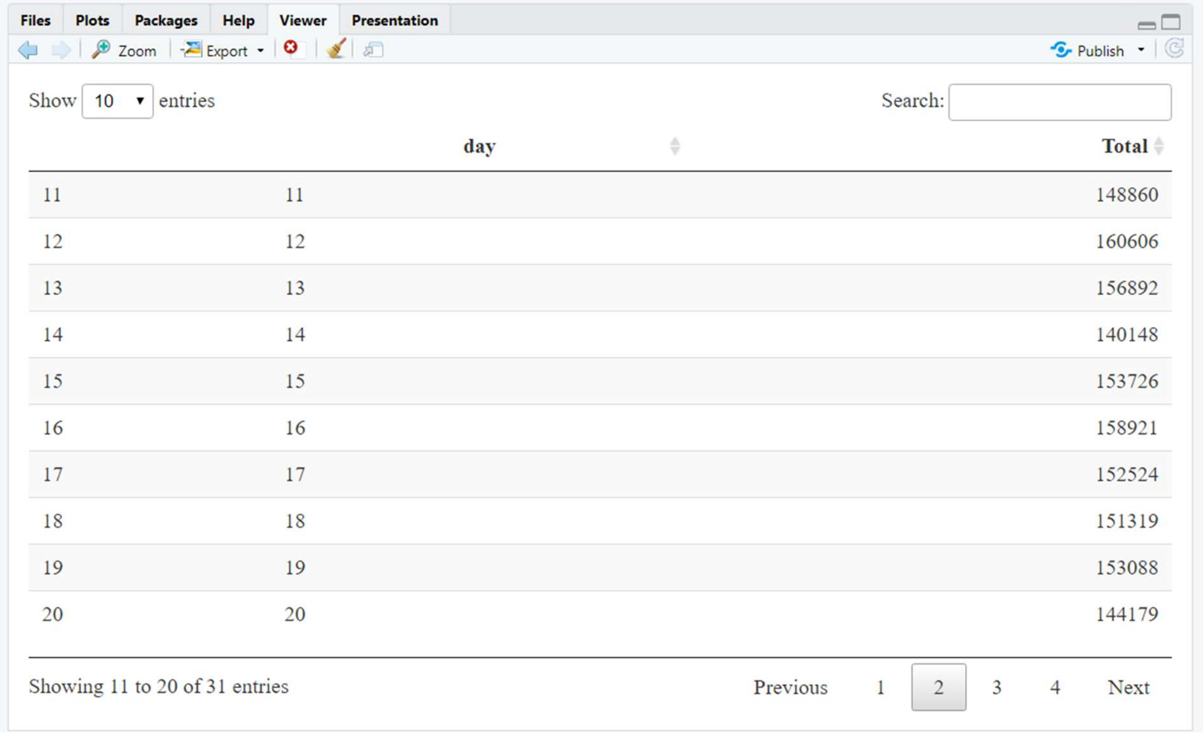
```
>
> day_group <- data_2014 %>%
+     group_by(day) %>%
+     dplyr::summarize(Total = n())
> datatable(day_group)
> |
```



Show 10 entries Search:

	day	Total
1	1	127430
2	2	143201
3	3	142983
4	4	140923
5	5	147054
6	6	139886
7	7	143503
8	8	145984
9	9	155135
10	10	152500

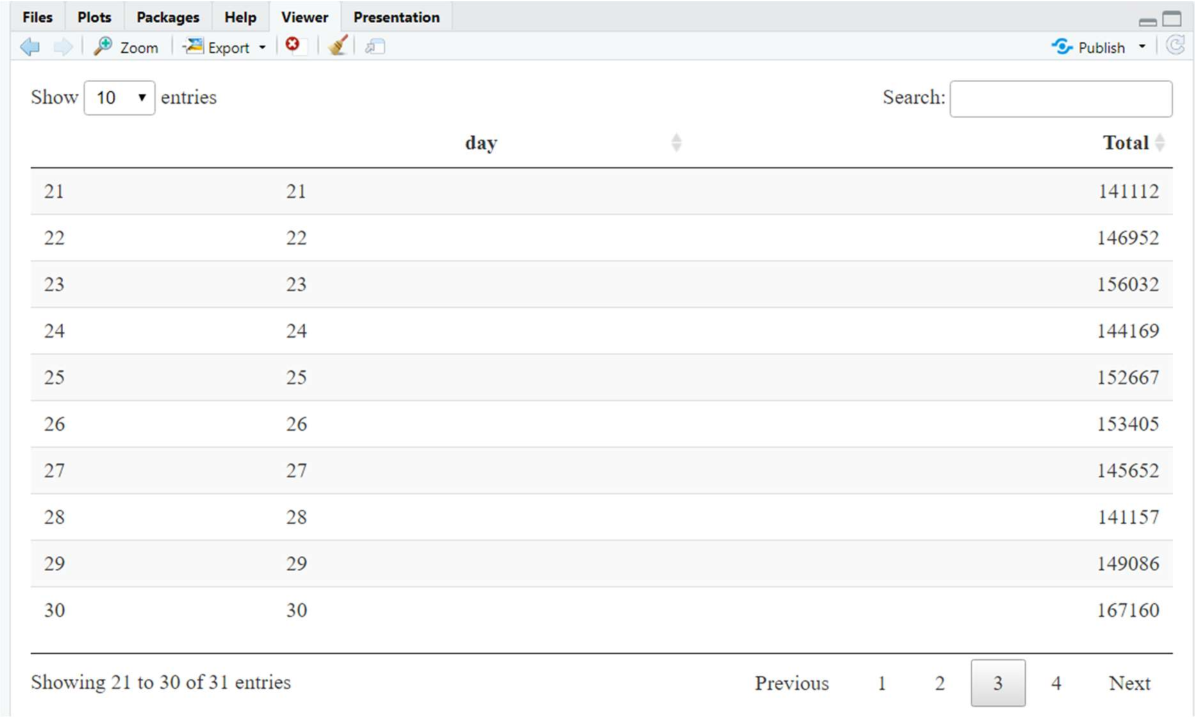
Showing 1 to 10 of 31 entries



Show 10 entries Search:

	day	Total
11	11	148860
12	12	160606
13	13	156892
14	14	140148
15	15	153726
16	16	158921
17	17	152524
18	18	151319
19	19	153088
20	20	144179

Showing 11 to 20 of 31 entries



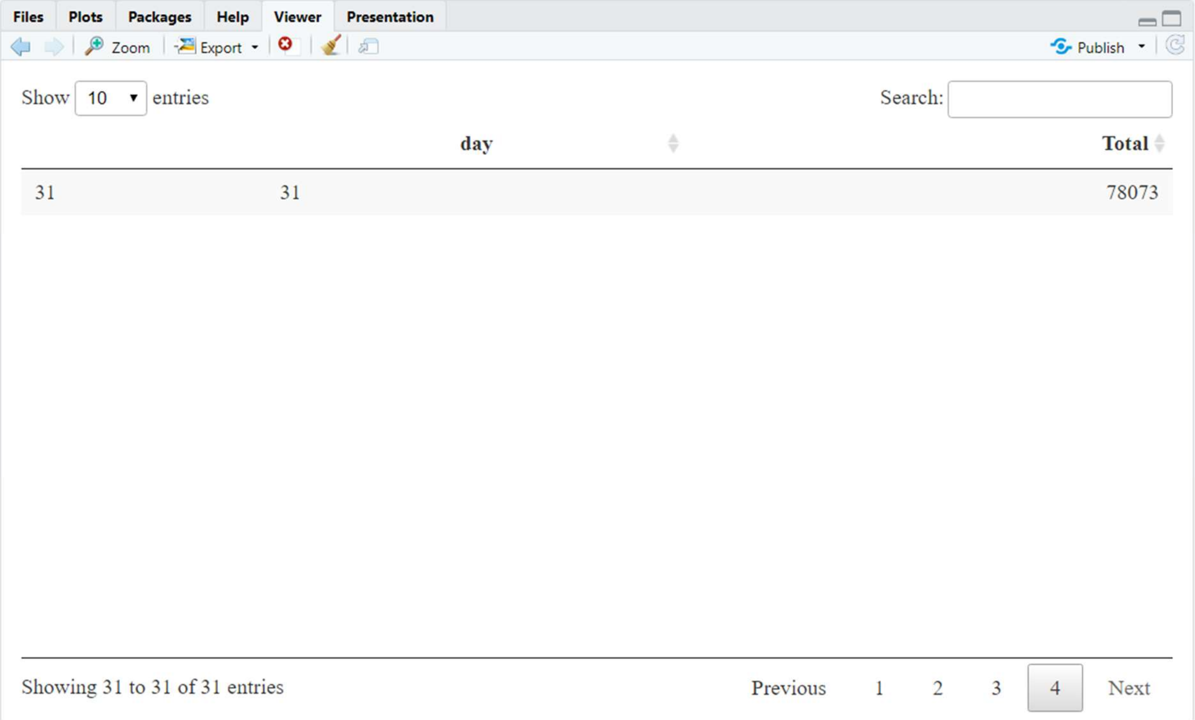
Show 10 entries

Search:

	day	Total
21	21	141112
22	22	146952
23	23	156032
24	24	144169
25	25	152667
26	26	153405
27	27	145652
28	28	141157
29	29	149086
30	30	167160

Showing 21 to 30 of 31 entries

Previous 1 2 3 4 Next



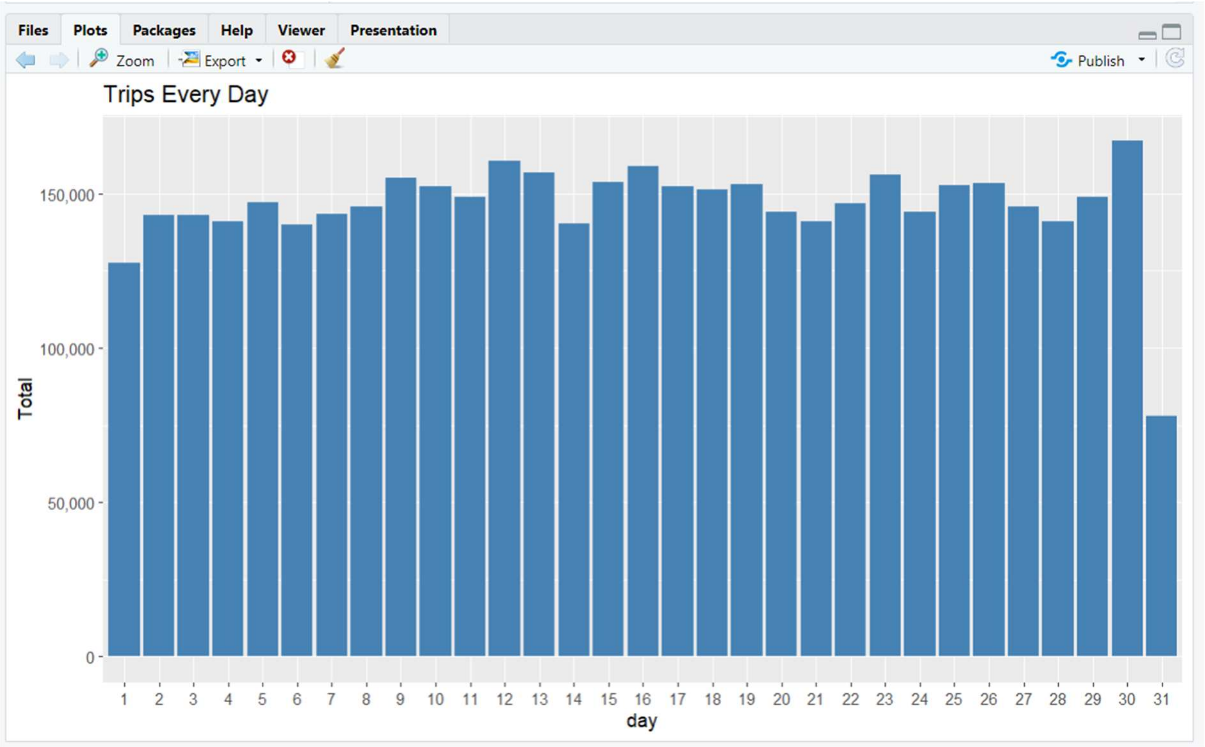
Show 10 entries

Search:

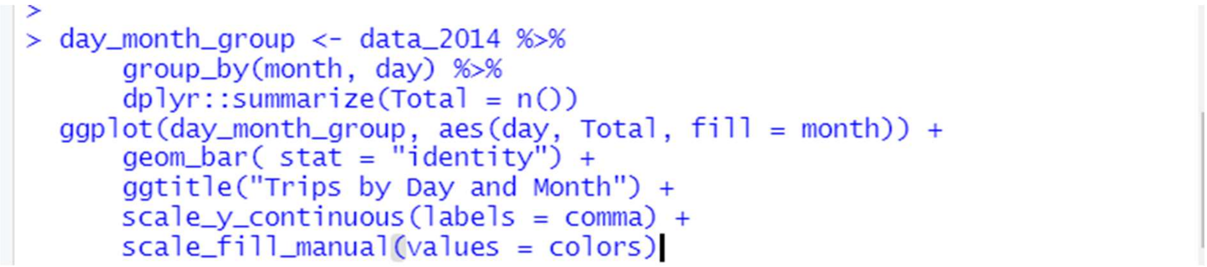
	day	Total
31	31	78073

Showing 31 to 31 of 31 entries

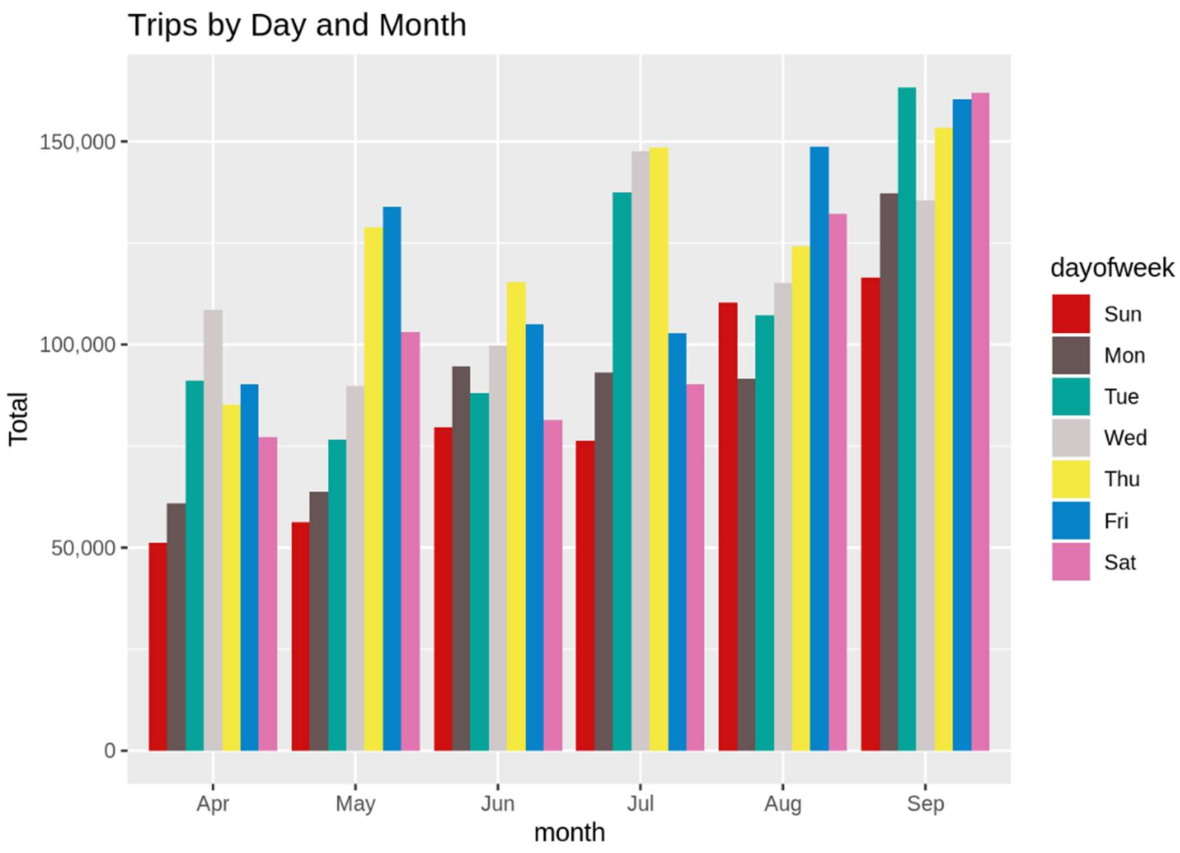
Previous 1 2 3 4 Next



Trips by Day and Month:

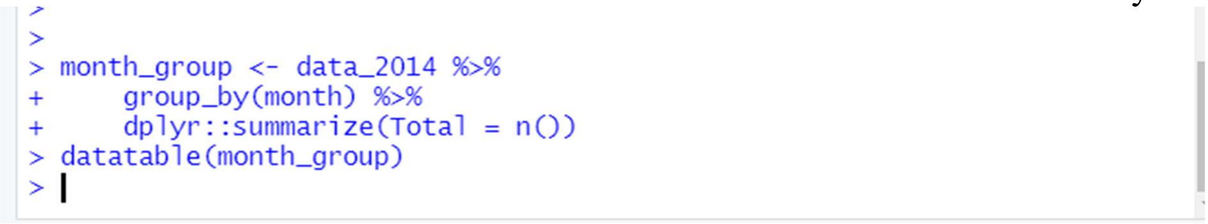


```
>
> day_month_group <- data_2014 %>%
      group_by(month, day) %>%
      dplyr::summarize(Total = n())
  ggplot(day_month_group, aes(day, Total, fill = month)) +
      geom_bar( stat = "identity") +
      ggtitle("Trips by Day and Month") +
      scale_y_continuous(labels = comma) +
      scale_fill_manual(values = colors)
```



Number of Trips taking place during months in a year

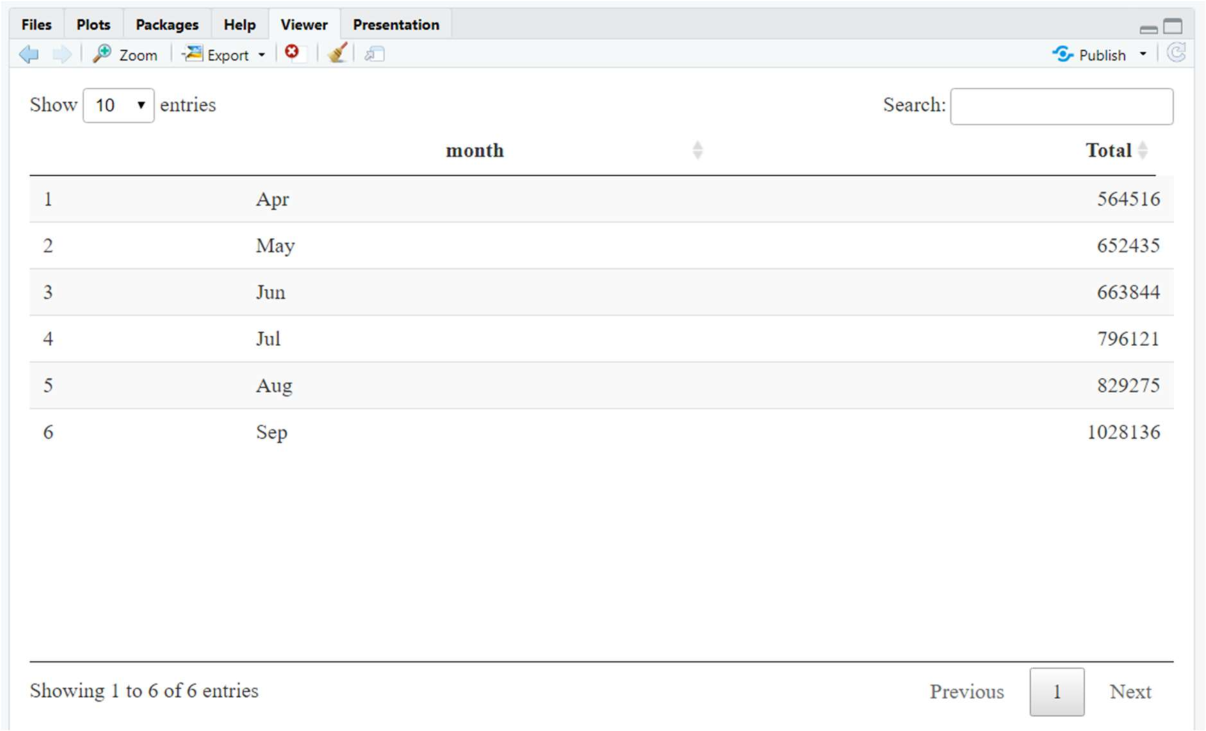
Here, we will visualize the number of trips that are taking place each month of the year.



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

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.

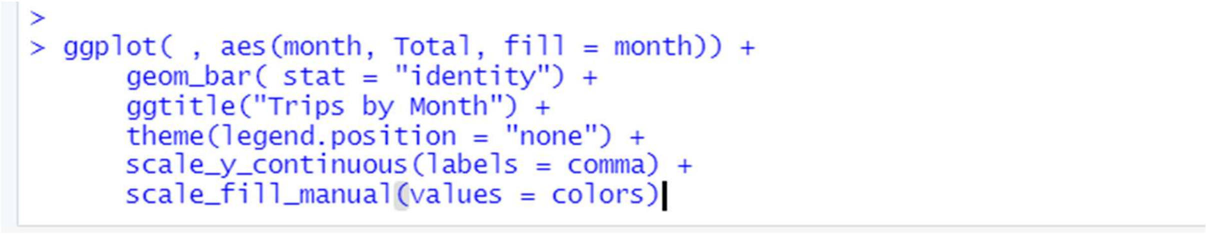


Show 10 entries

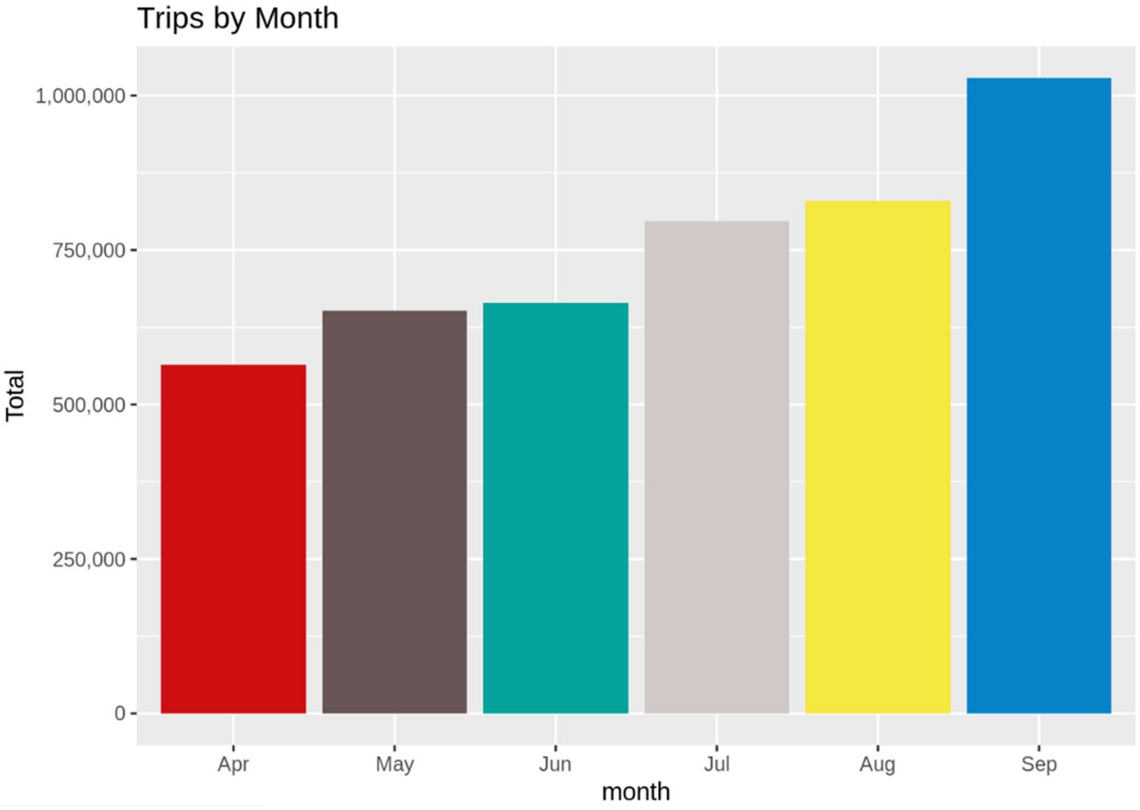
Search:

	month	Total
1	Apr	564516
2	May	652435
3	Jun	663844
4	Jul	796121
5	Aug	829275
6	Sep	1028136

Showing 1 to 6 of 6 entries

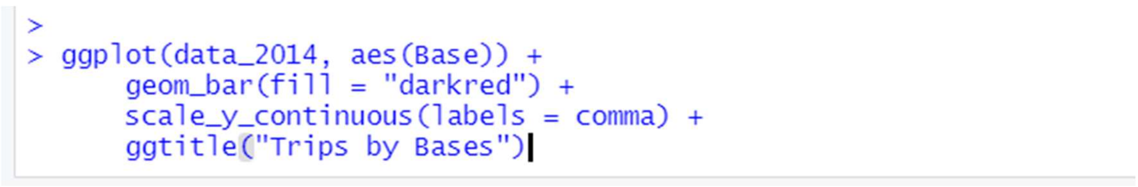


```
>
> ggplot( , aes(month, Total, fill = month)) +
      geom_bar( stat = "identity") +
      ggtitle("Trips by Month") +
      theme(legend.position = "none") +
      scale_y_continuous(labels = comma) +
      scale_fill_manual(values = colors)
```

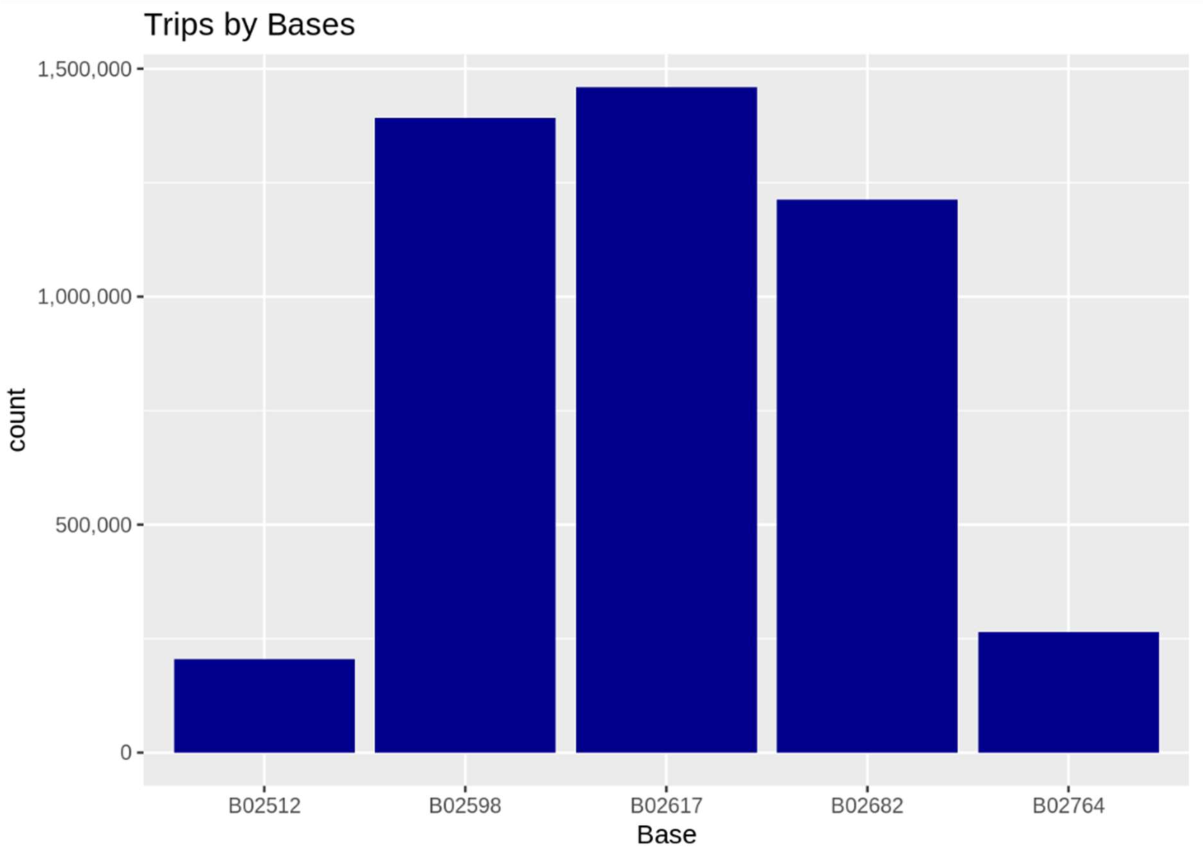


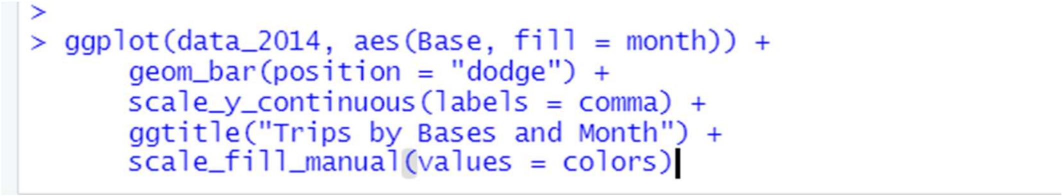
Finding out the number of Trips by bases

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.

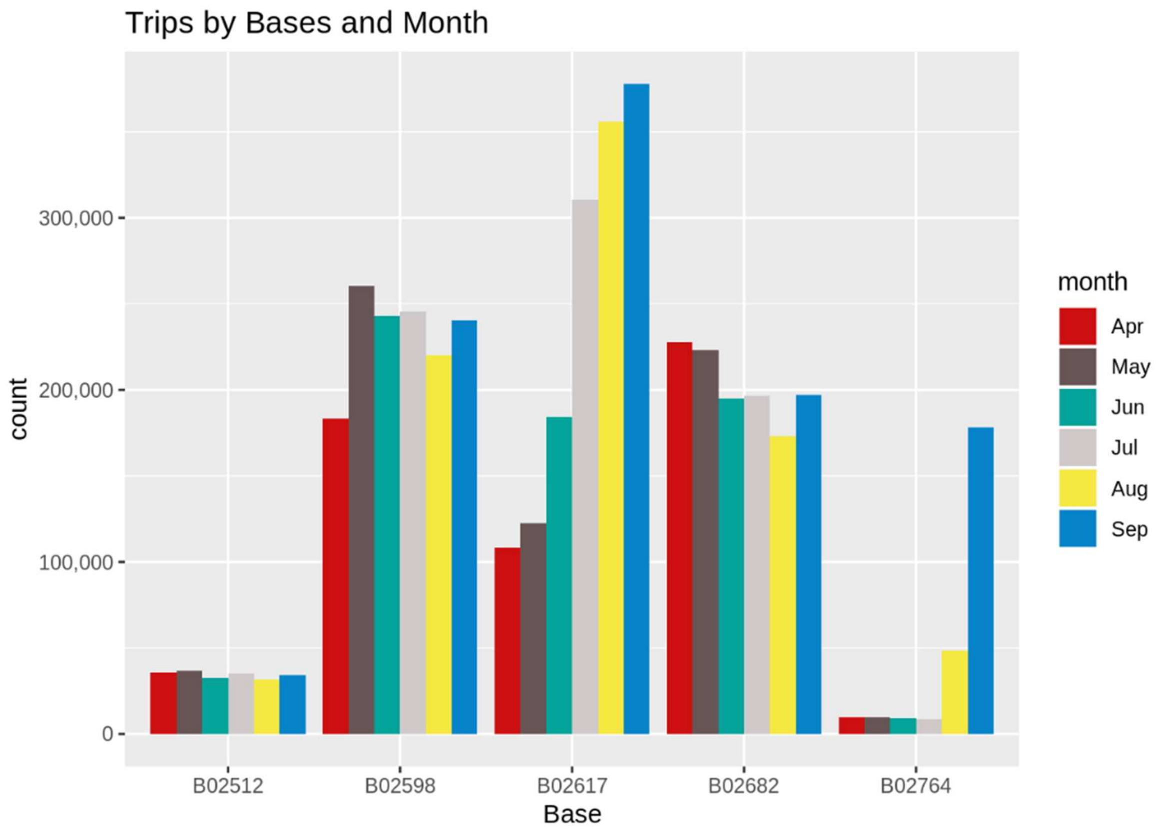


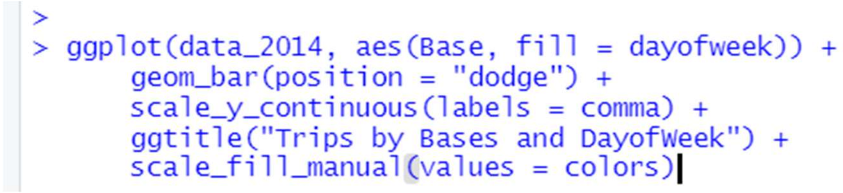
```
>
> ggplot(data_2014, aes(Base)) +
      geom_bar(fill = "darkred") +
      scale_y_continuous(labels = comma) +
      ggtitle("Trips by Bases")
```



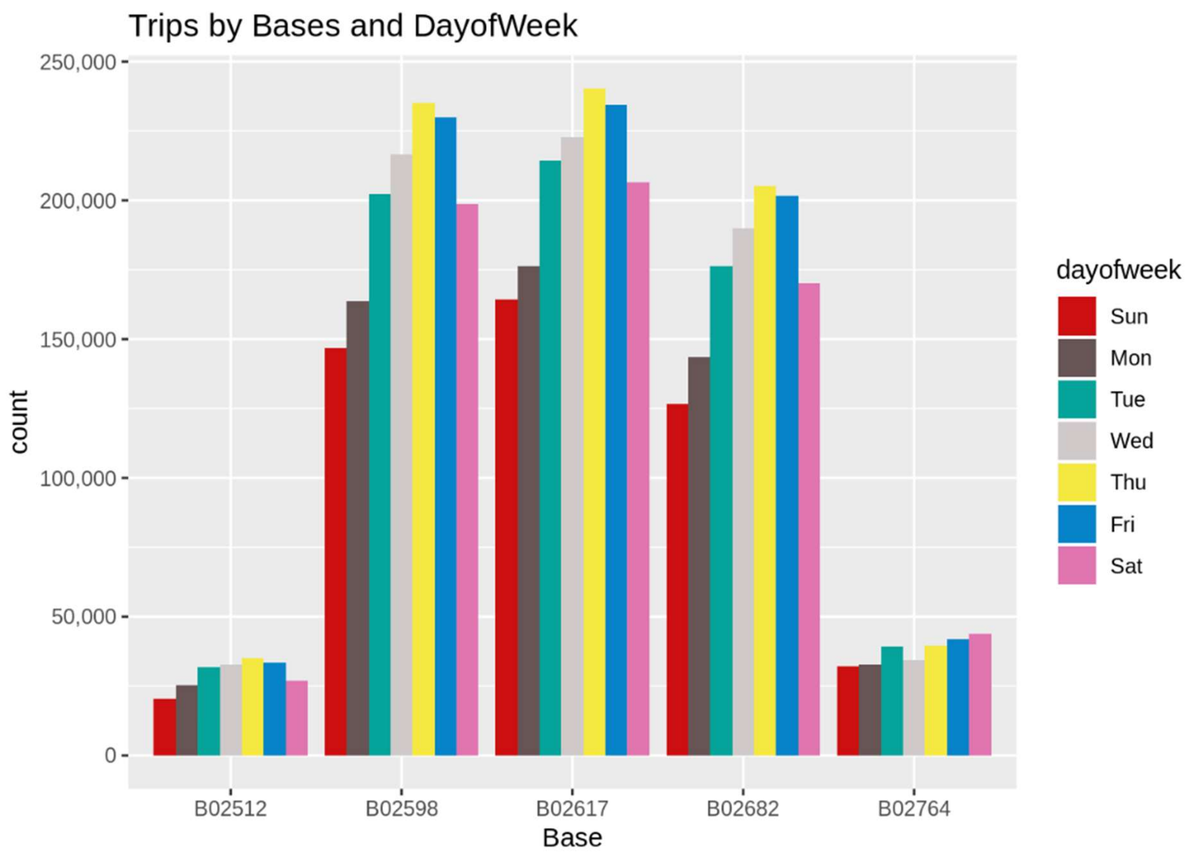


```
>
> ggplot(data_2014, aes(Base, fill = month)) +
      geom_bar(position = "dodge") +
      scale_y_continuous(labels = comma) +
      ggtitle("Trips by Bases and Month") +
      scale_fill_manual(values = colors)
```

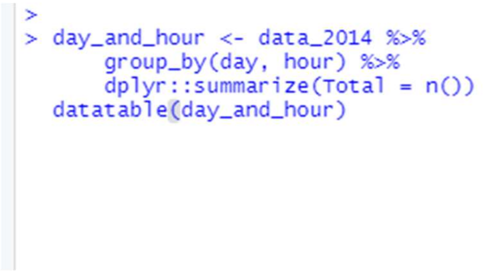




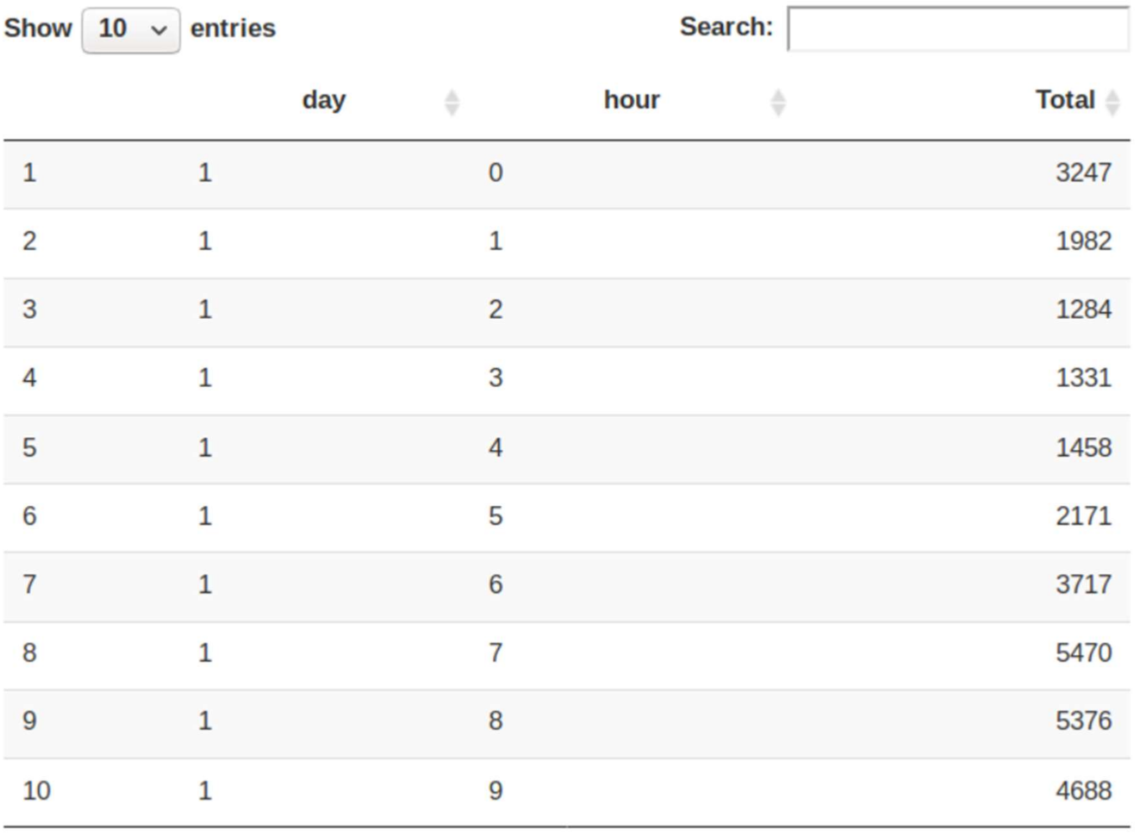
```
>
> ggplot(data_2014, aes(Base, fill = dayofweek)) +
      geom_bar(position = "dodge") +
      scale_y_continuous(labels = comma) +
      ggtitle("Trips by Bases and DayofWeek") +
      scale_fill_manual(values = colors)
```



Creating a Heatmap visualization of day, hour and month

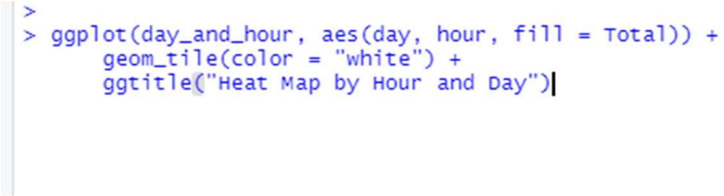


```
>
> day_and_hour <- data_2014 %>%
      group_by(day, hour) %>%
      dplyr::summarize(Total = n())
  datatable(day_and_hour)
```

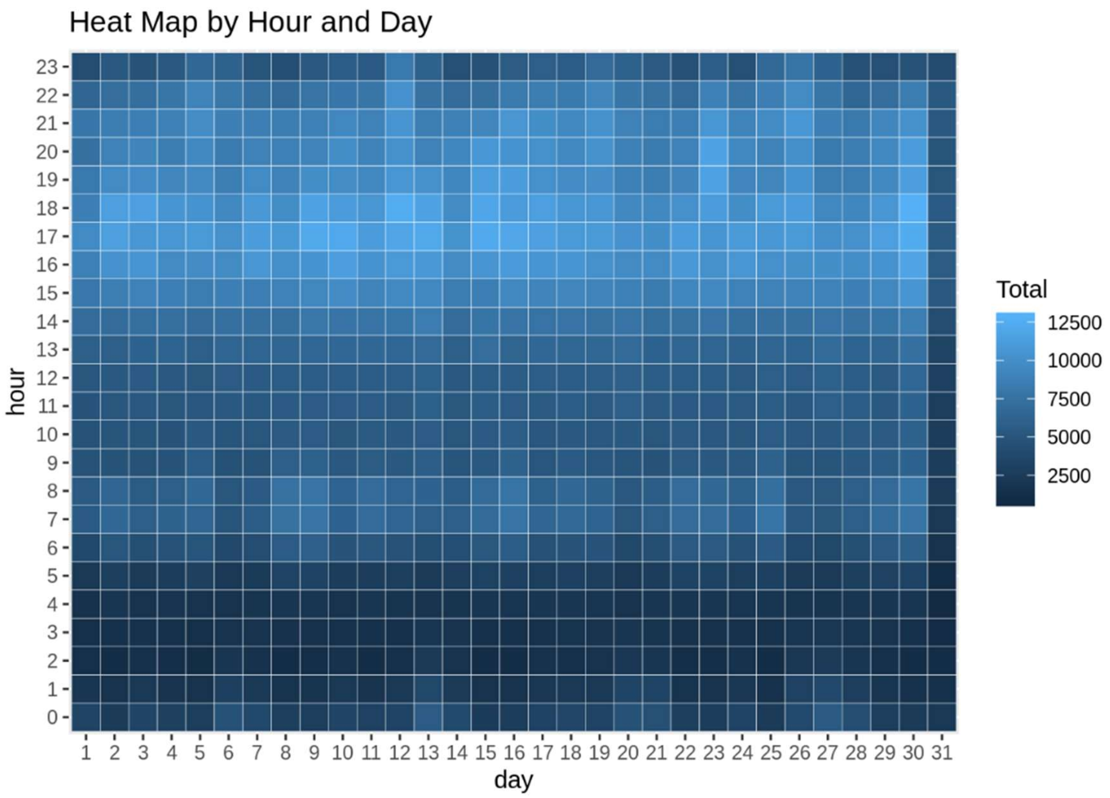


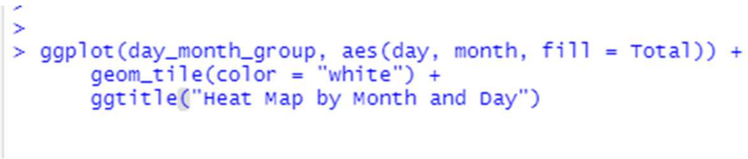
Show 10 entries Search:

	day	hour	Total
1	1	0	3247
2	1	1	1982
3	1	2	1284
4	1	3	1331
5	1	4	1458
6	1	5	2171
7	1	6	3717
8	1	7	5470
9	1	8	5376
10	1	9	4688

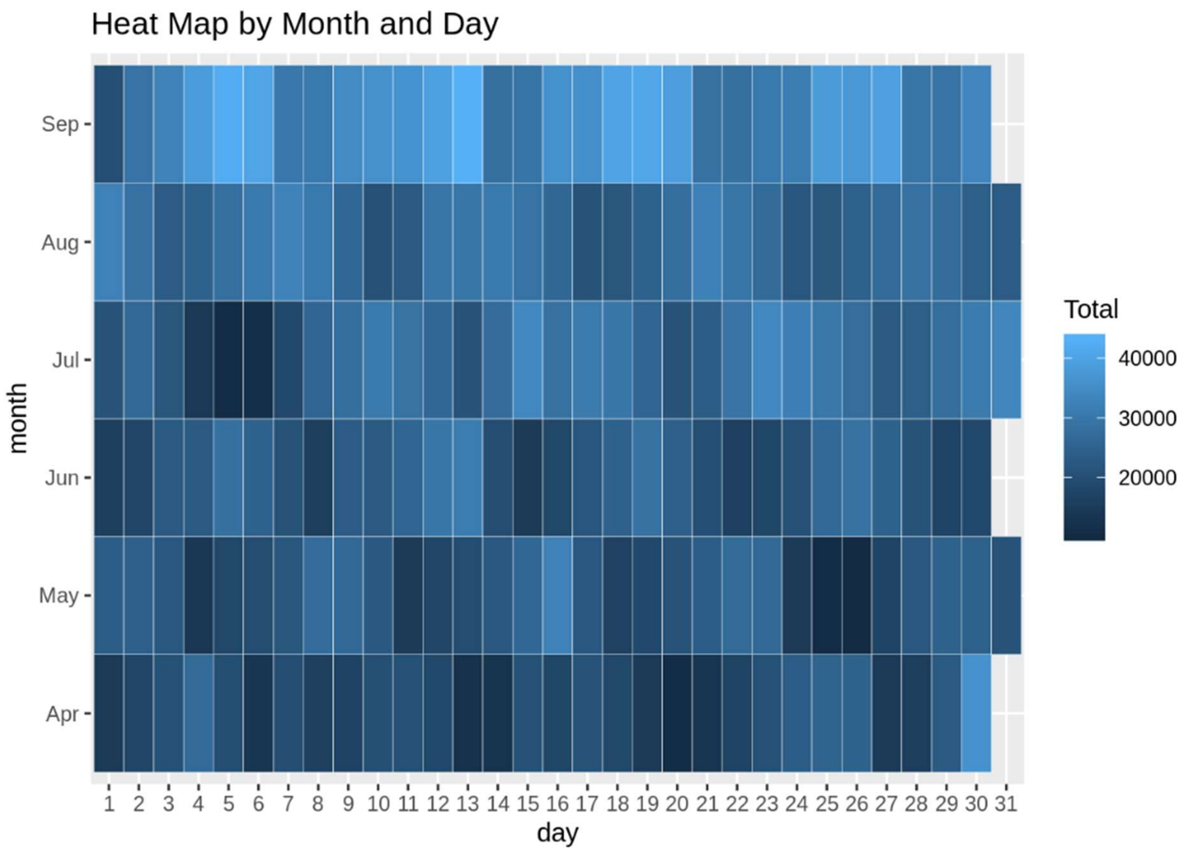


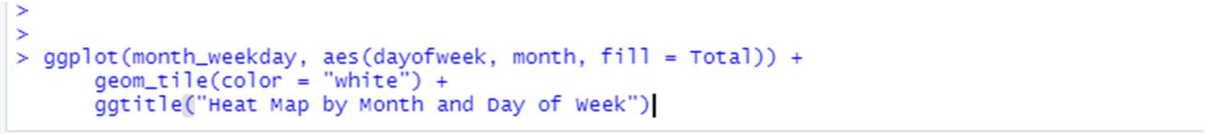
```
>
> ggplot(day_and_hour, aes(day, hour, fill = Total)) +
     geom_tile(color = "white") +
     ggtitle("Heat Map by Hour and Day")
```



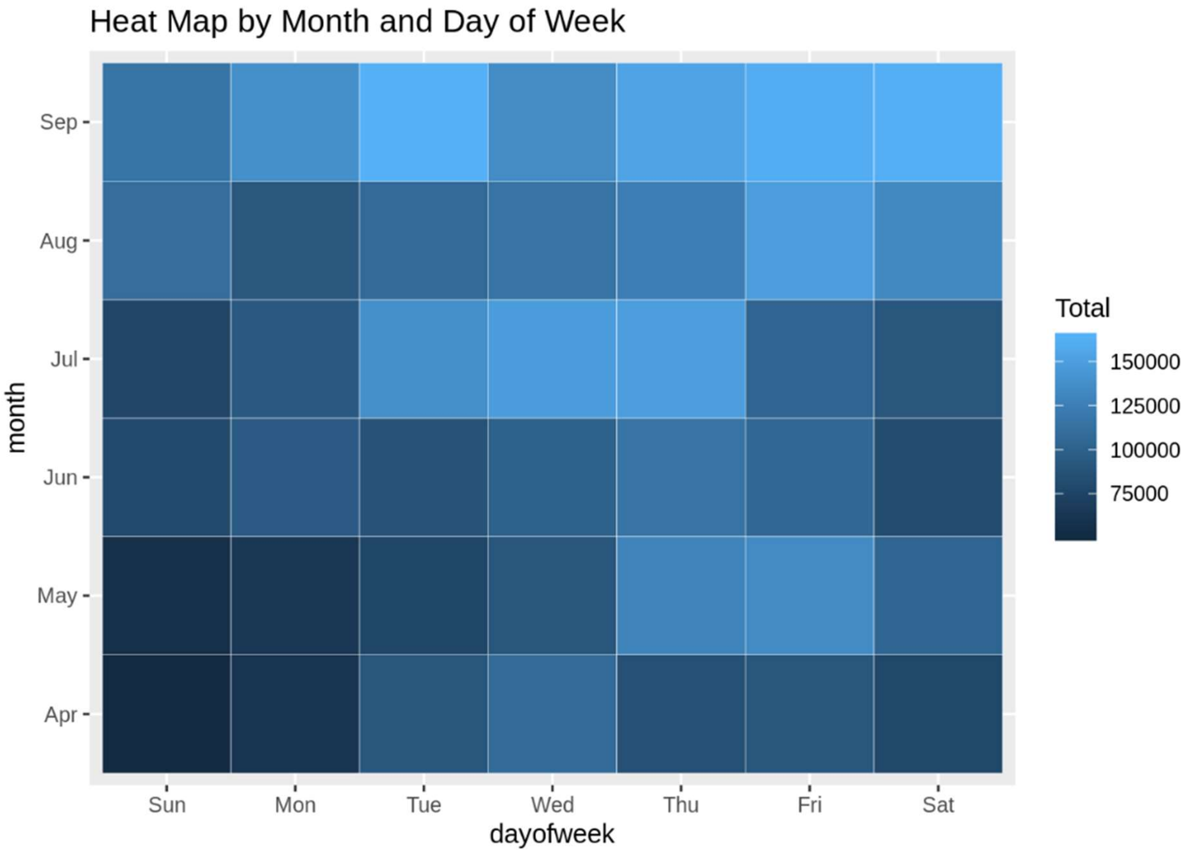


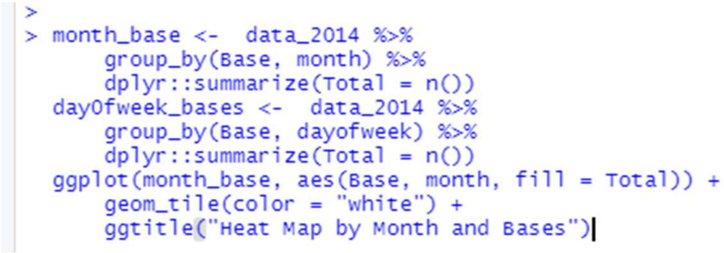
```
>
> ggplot(day_month_group, aes(day, month, fill = Total)) +
      geom_tile(color = "white") +
      ggtitle("Heat Map by Month and Day")
```



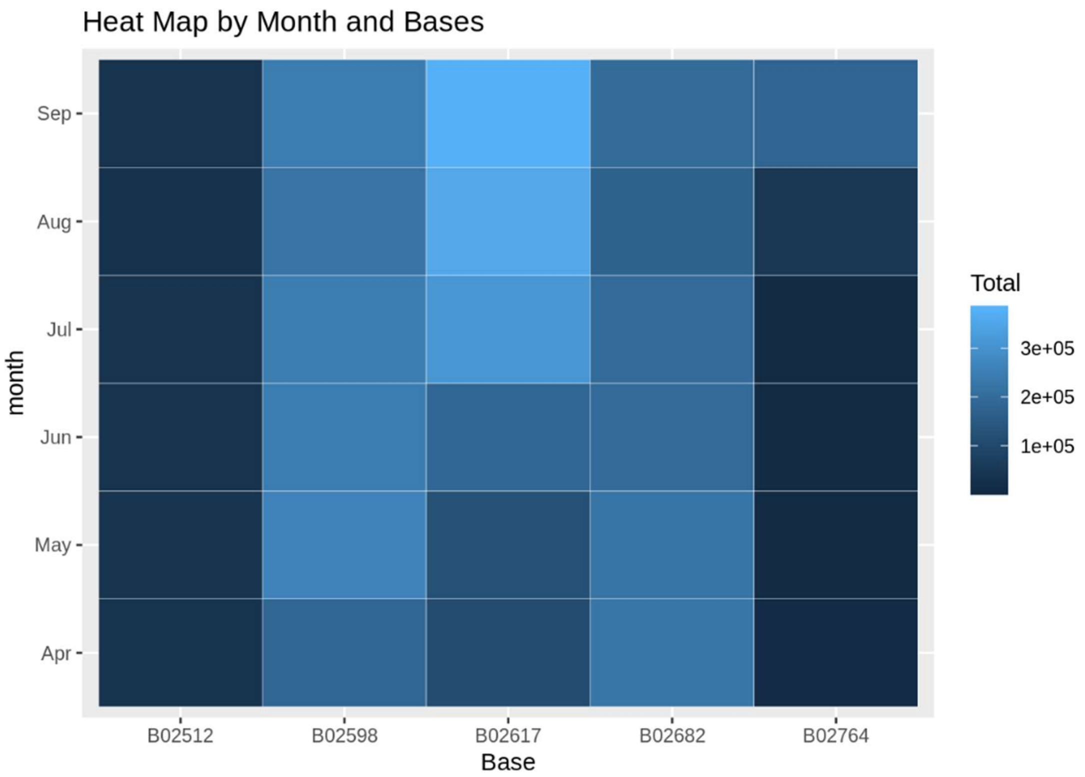


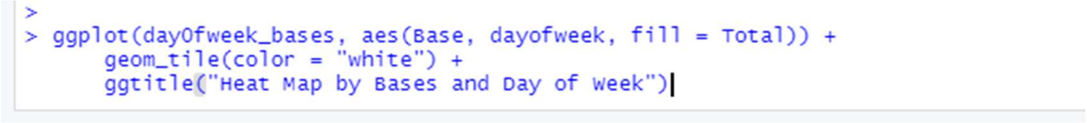
```
>
>
> ggplot(month_weekday, aes(dayofweek, month, fill = Total)) +
      geom_tile(color = "white") +
      ggtitle("Heat Map by Month and Day of Week")
```



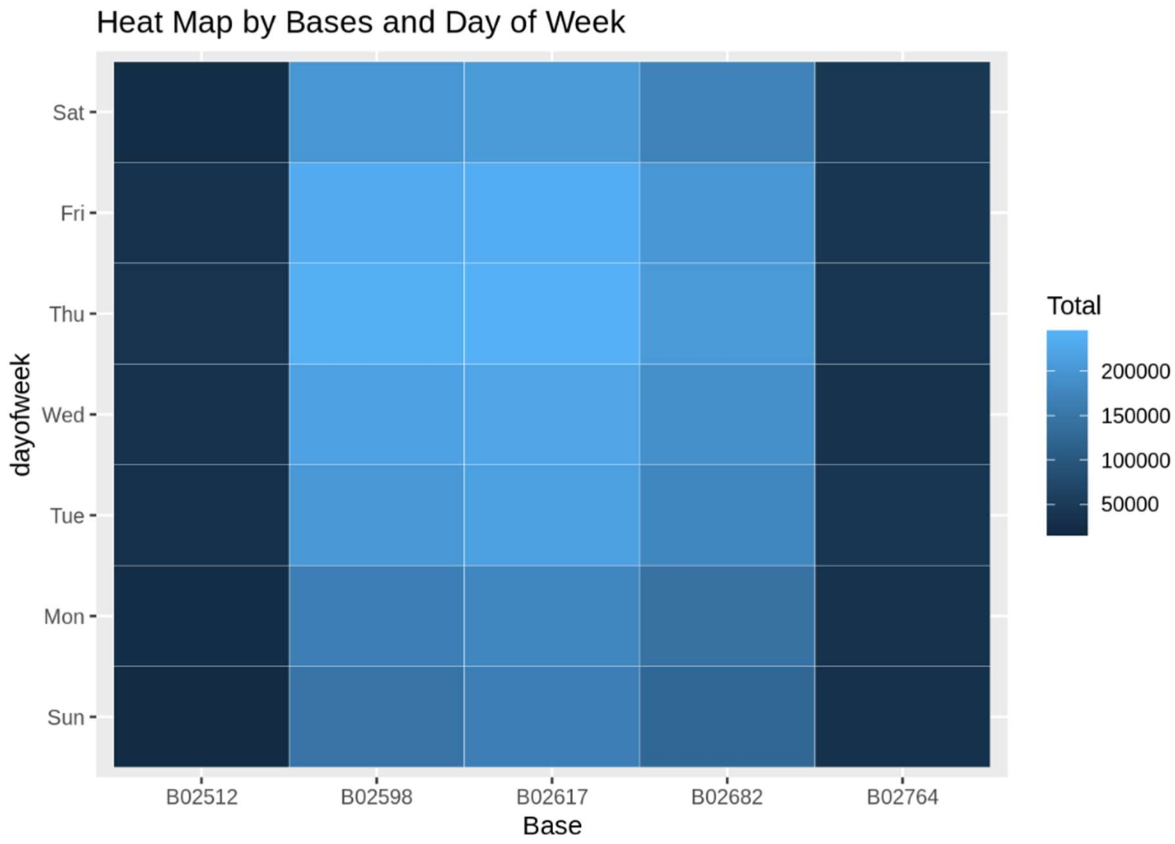


```
>
> month_base <-  data_2014 %>%
      group_by(Base, month) %>%
      dplyr::summarize(Total = n())
  dayOfweek_bases <-  data_2014 %>%
      group_by(Base, dayofweek) %>%
      dplyr::summarize(Total = n())
  ggplot(month_base, aes(Base, month, fill = Total)) +
      geom_tile(color = "white") +
      ggtitle("Heat Map by Month and Bases")
```



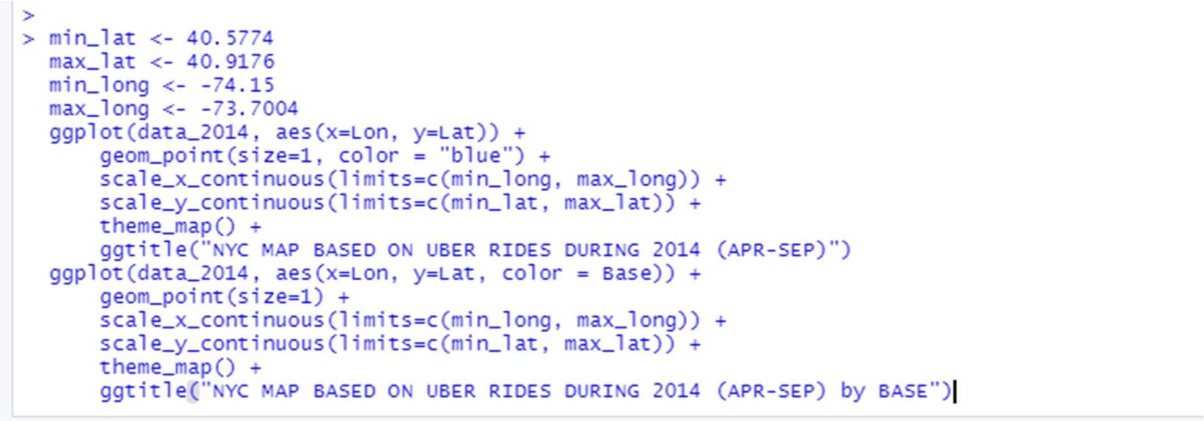


```
>
> ggplot(dayOfweek_bases, aes(Base, dayofweek, fill = Total)) +
     geom_tile(color = "white") +
     ggtitle("Heat Map by Bases and Day of Week")
```



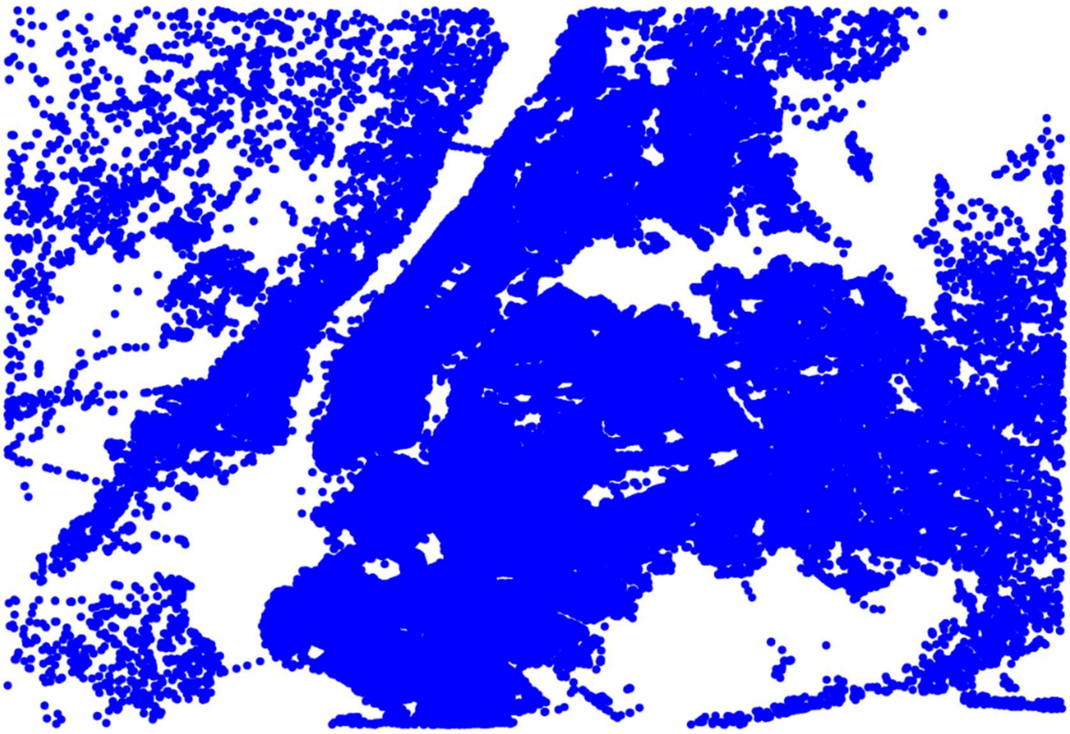
Creating a map visualization of rides in New York

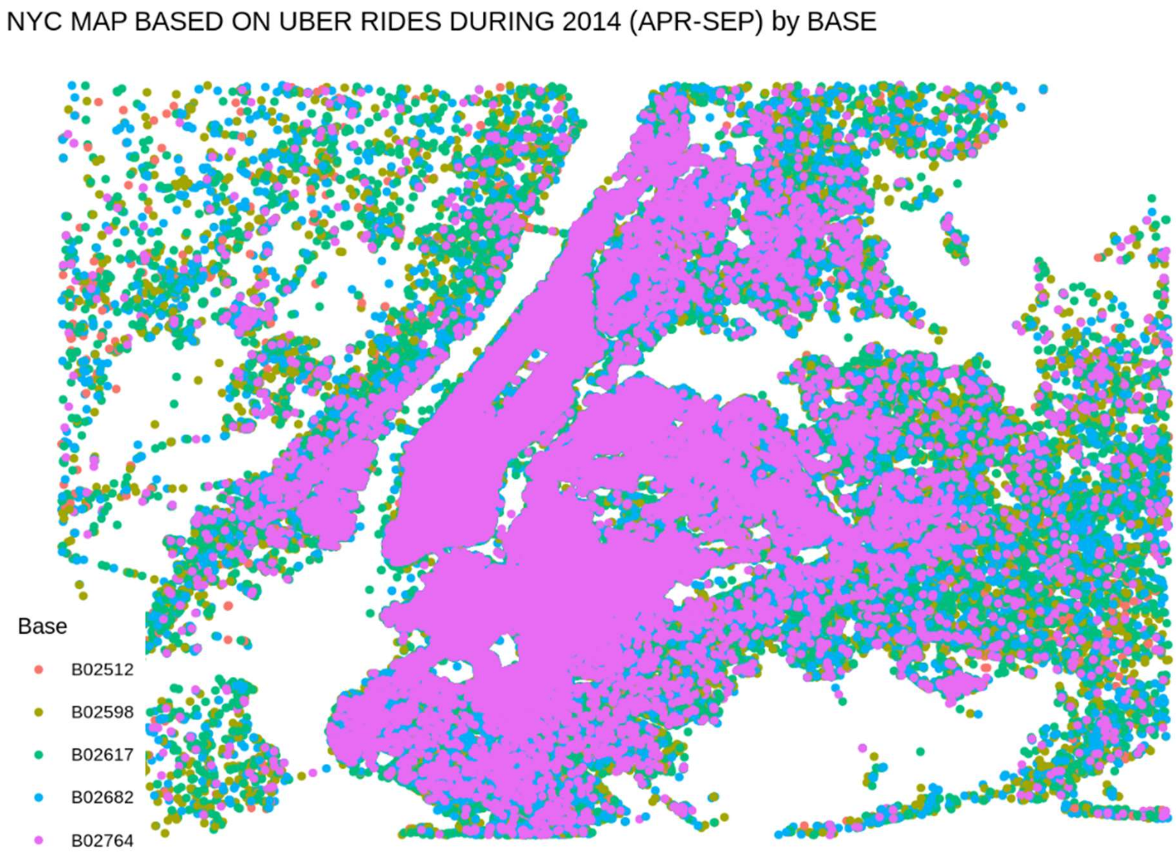
In the final section, we will visualize the rides in New York city by creating a geo-plot that will help us to visualize the rides during 2014 (Apr – Sep) and by the bases in the same period.



```
>
> 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)")
  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")
```

NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP)

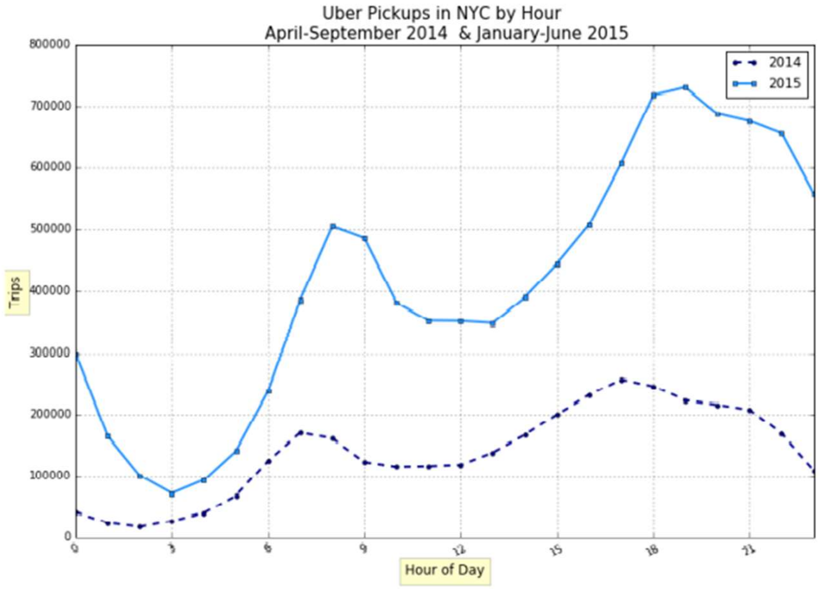




At the end of the Uber data analysis R project, we observed how to create data visualizations. We made use of packages like ggplot2 that allowed us to plot various types of visualizations that pertained to several time-frames of the year. With this, we could conclude how time affected customer trips.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

Insights:

- From 2014 to 2015, Uber trips increased dramatically by 10 million (223.3%), while taxi trips (include both yellow and green taxis) decreased slightly by 0.8 million (1.0%).



(a)

Summary:

Toward the finish of the Uber data analysis R project, we saw how to make Data visualizations. We utilized packages like ggplot2 that permitted us to plot different kinds of representations that related to a few time periods of the year. With this, we could finish up what time meant for client trips. At last, we made a geo plot of New York City that gave us the brief idea of how different clients made trips from various bases.

Conclusion:

At the end of this Uber data analysis R project, we studied how to create data visualizations. We used package ggplot2 that helped us to plot various types of visualizations that pertained to several time-frames of the year. With this, we conclude how time and place affected customer trips.

Finally, we made visualization a Geo plot of New York that provided us with the details of how various users made trips from different bases.

Future Scope:

We can use this data for training a model using ML and building a smart AI based predictive system. Model can automatically send the insights to the authorities or drivers related to areas having most trips and passenger count in certain areas. This big data can be used to study passenger’s behaviour.

References:

[1] https://ggplot2.tidyverse.org/

[2] **https://www.kaggle.com/datasets/fivethirtyeight/uber-pickups-in-new-york-city**

[3] **https://www.skyfilabs.com/project-ideas/uber-data-analysis**

[4] **https://ieeexplore.ieee.org/abstract/document/7584941**

[5] **L. K. Poulsen et al., "Green Cabs vs. Uber in New York City," 2016 IEEE International Congress on Big Data (BigData Congress), 2016, pp. 222-229, doi: 10.1109/BigDataCongress.2016.35.**

[6] **https://posit.co/blog/introducing-tidyr/#:~:text=tidyr%20is%20new%20package%20that,Each%20column%20is%20a%20variable.**