Project Based Learning Report

on

**Uber Dataset Time Series Analysis in R**

Submitted in the partial fulfillment of the requirements

For the Project based learning in (**Essentials of Data Science**)

in

Electronics & Communication Engineering

By

**2014111101 Ansh Sahu**

**2014111094 Anshuman Raj**

**2014111088 Shubham Parkhi**

Under the guidance of Course In-charge

Prof. Dnyanesh S.Lavhkare

Department of Electronics & Communication Engineering

Bharati Vidyapeeth

(Deemed to be University)

College of Engineering,

Pune – 4110043

**Academic Year: 2021-22**

**Bharati Vidyapeeth**

**(Deemed to be University)**

**College of Engineering,**

**Pune – 411043**

**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**CERTIFICATE**

Certified that the Project Based Learning report entitled **Uber Dataset Time Series Analysis in R**

is work done by

**2014111101 Ansh Sahu**

**2014111094 Anshuman Raj**

**2014111088 Shubham Parkhi**

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in **Essentials of Data Science Course** of Bachelor of Technology Semester IV, Electronics & Communication Engineering.

**Date: 21 May 2022**

**Prof. Dnyanesh S.Lavhkare Dr. Tanuja S.Dhope**

**Course In-charge PBL Co-Ordinator**

**Dr. Arundhati A.Shinde**

**Professor & Head**

**ELECTRONICS & COMMUNICATION ENGINEERING**

|  |  |
| --- | --- |
| **Index: -** | |
| Page No. | Contents |
| 1-1 | Problem Statement with Solution |
| 2-5 | Description about project |
| 6-9 | Software Used |
| 10-13 | Results with Analysis |
| 14-14 | Conclusion & Outcome |

**Problem Statement :-**

What is Data Science? Why learn Data Science?

**Solution :-**

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models. The data used for analysis can come from many different sources and presented in various formats.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science practitioners apply [machine learning](https://www.datarobot.com/wiki/machine-learning/) [algorithms](https://www.datarobot.com/wiki/algorithm/) to numbers, text, images, video, audio, and more to produce [artificial intelligence (AI)](https://www.datarobot.com/wiki/artificial-intelligence/) systems to perform tasks that ordinarily require human intelligence. In turn, these systems generate [insights](https://www.datarobot.com/wiki/insights/) which analysts and business users can translate into tangible business value.

Reasons to learn Data Science are: -

1. Learning about data science provides an opportunity for you to recreate yourself.
2. **We live in a digital world, everything is data-driven.** There is data science in **business, accounting, education, science, engineering, healthcare, technology, energy sector, government**, and so on.
3. **Data science is also a very promising field with lots of high paying job opportunities.**
4. **Basic data science skills are important for personal use.**
5. Great potential to branch out with different options.
6. Become a decision-maker, not every job opportunity will give you the power to make informed business decisions. For a data scientist, that is the core responsibility.
7. Less competitive because it is a highly analytical role, competition is less, but demand is not. With a limited talent pool, there is always a challenge for businesses to hire in these roles.

**1**

# **Project in R – Uber Data Analysis Project**

Step by step guide to building a Uber Data Time Series Analysis

Talking about our Uber data analysis project, data storytelling is an important component of [***Machine Learning***](https://techvidvan.com/tutorials/machine-learning-tutorial/) 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.

### **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.

* lubridate

Our dataset involves various time-frames. In order to understand our data in separate time categories, we will make use of the lubridate package.

* dplyr

This package is the lingua franca of [***data manipulation in R***](https://data-flair.training/blogs/manipulating-and-processing-data-in-r/).

* tidyr

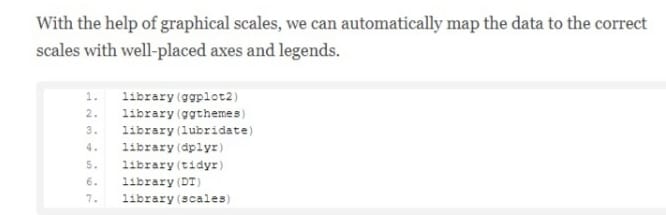
This package will help you to tidy your data. The basic principle of tidyr is to tidy the columns where each variable is present in a column, each observation is represented by a row and each value depicts a cell.

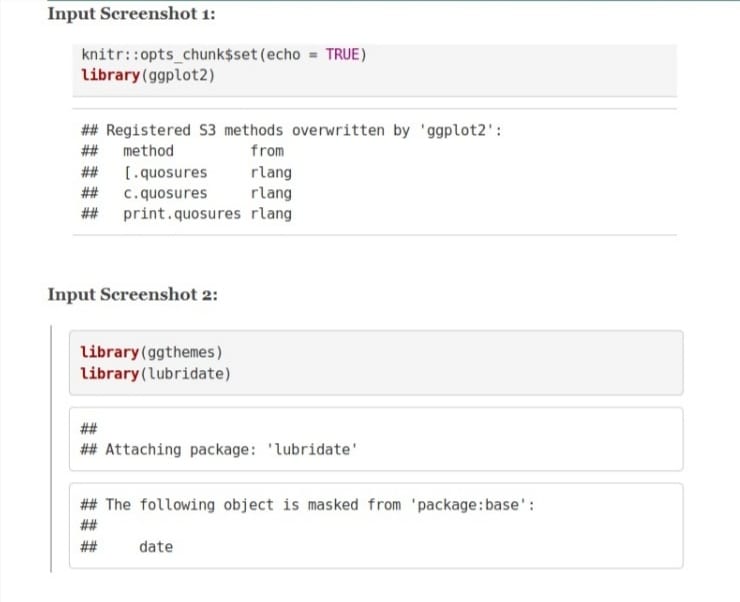
* DT

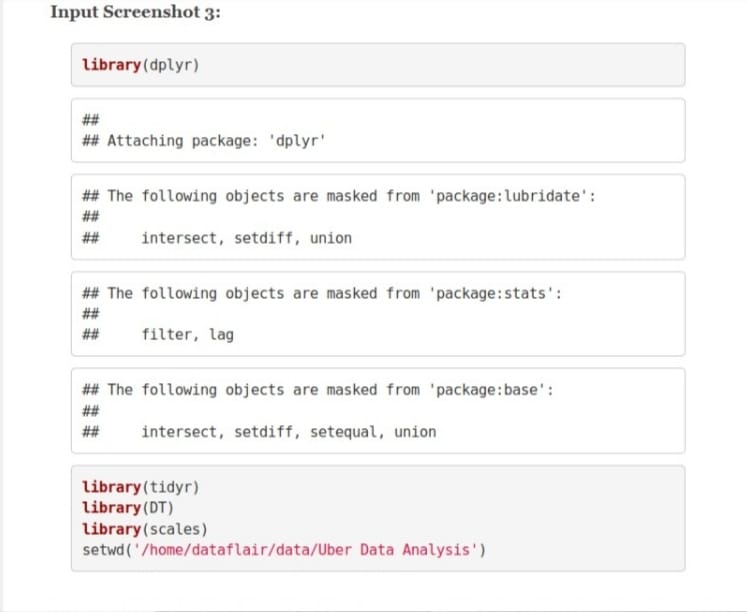
With the help of this package, we will be able to interface with the [***JavaScript***](https://data-flair.training/blogs/javascript-tutorials-home/)Library called – Datatables.

* scales

With the help of graphical scales, we can automatically map the data to the correct scales with well-placed axes and legends.







### **2. Creating vector of colors to be implemented in our plots**

In this step of data science project, we will create a vector of our colors that will be included in our plotting functions. You can also select your own set of colors.

**Code:**

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

**Input Screenshot 4:**

[create vector of colors](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/create-vector-of-colors.jpg)

### **3. 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’.

Then, in the next step, we will perform the appropriate formatting of Date.Time column. Then, we will proceed to create factors of time objects like day, month, year etc.

**Code:**

apr\_data <- read.csv("uber-raw-data-apr14.csv")

may\_data <- read.csv("uber-raw-data-may14.csv")

jun\_data <- read.csv("uber-raw-data-jun14.csv")

jul\_data <- read.csv("uber-raw-data-jul14.csv")

aug\_data <- read.csv("uber-raw-data-aug14.csv")

sep\_data <- read.csv("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**))

**Input Screenshot 5:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/2.1-Reading-FIle.png)

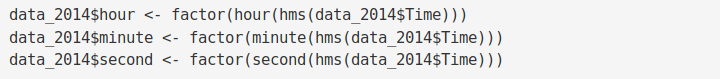
**Code:**

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)))

**Input Screenshot 6:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/2.2-Reading-FIle.png)

### **Plotting the trips by the hours in a day**

In the next step or R project, we will use 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.

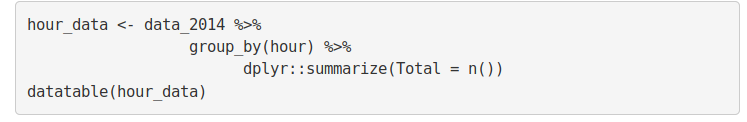
hour\_data <- data\_2014 %>%

group\_by(hour) %>%

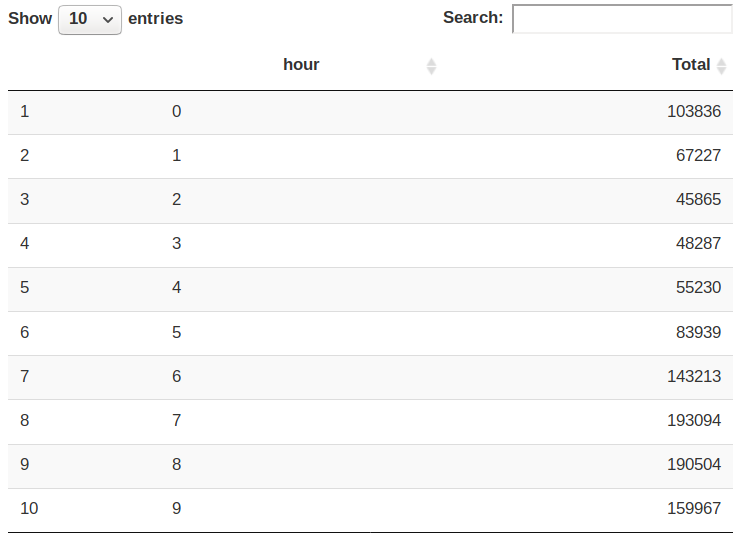
dplyr::summarize(Total = n())

datatable(hour\_data)

**Input Screenshot 7:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.0-Plotting-the-trips-by-the-hours-in-a-day-Code.png)

**Output Screenshot:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.1-Plotting-the-trips-by-the-hours-in-a-day-Output.png)

**Code:**

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())

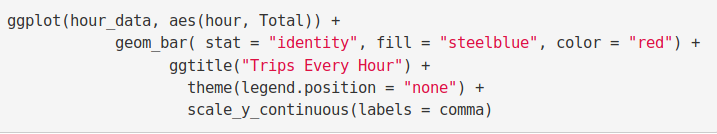
ggplot(month\_hour, aes(hour, Total, fill = month)) +

geom\_bar( stat = "identity") +

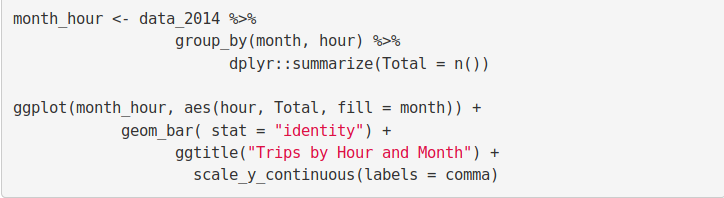
ggtitle("Trips by Hour and Month") +

scale\_y\_continuous(labels = comma)

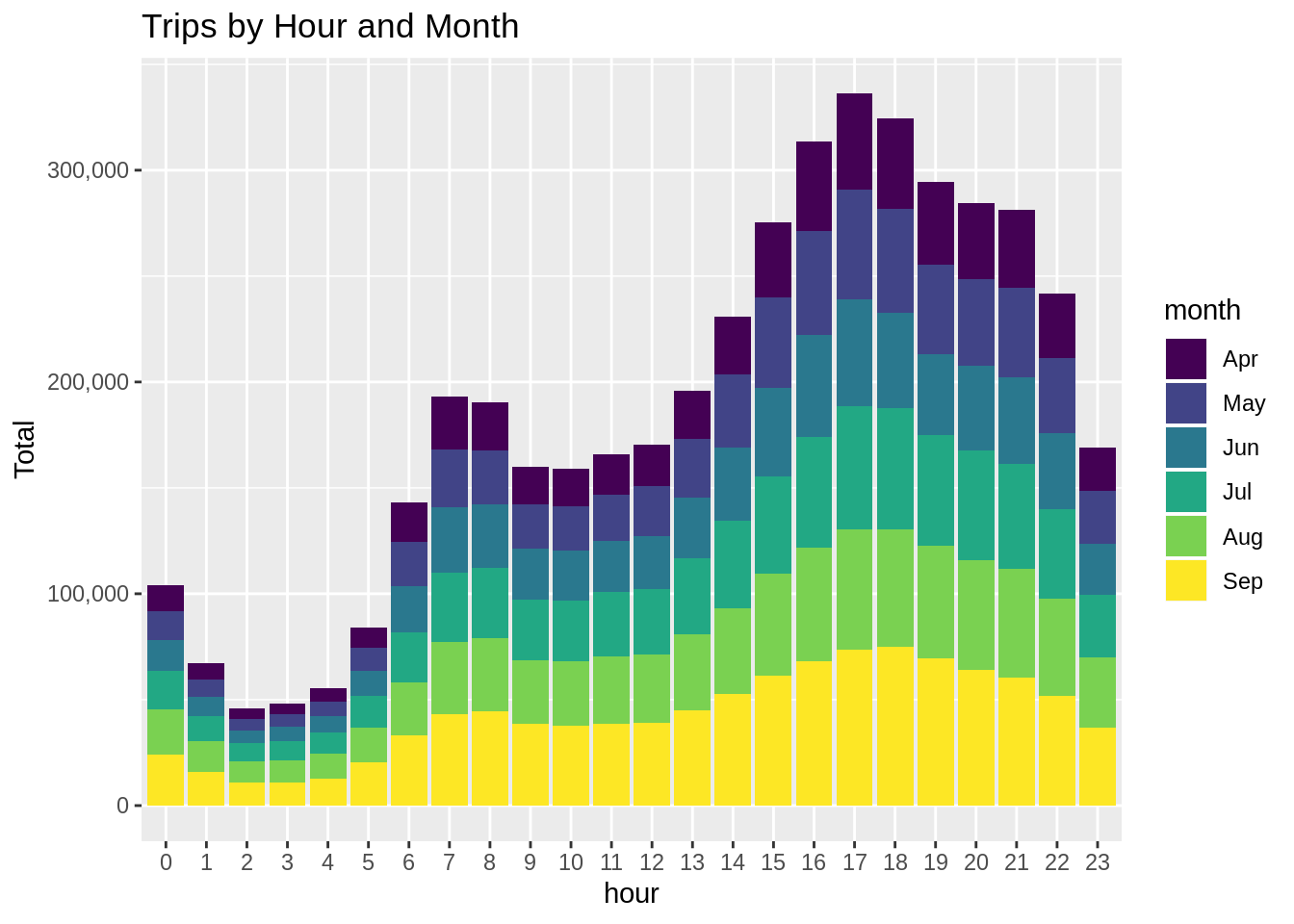
**Input Screenshot 8:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.2-Trips-every-Hour-Plot-Code.png)

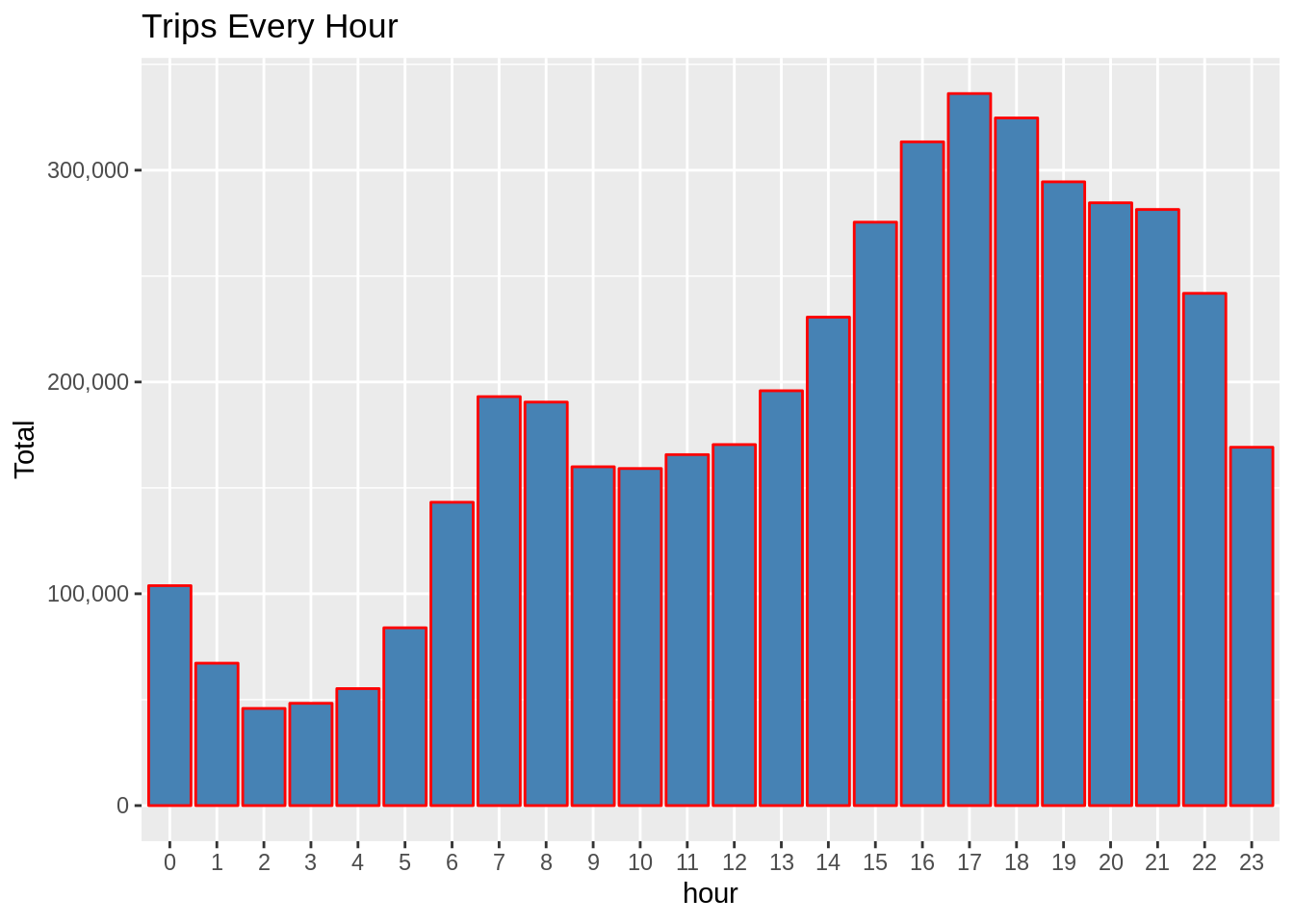
**Input Screenshot 9:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.3-month_hour.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Hours-and-Months.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Hours-in-a-Day-Plot-Output.png)

### **Plotting data by trips during every day of the month**

In this section of DataFlair 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.

**Code:**

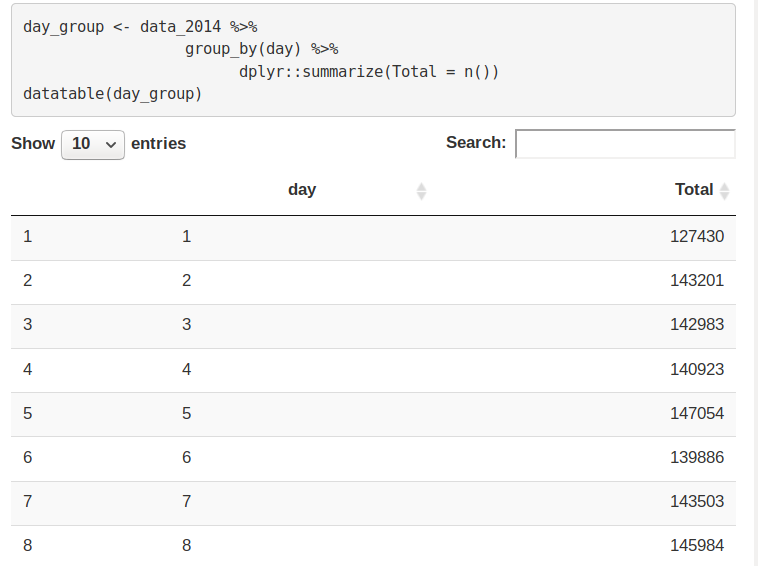
day\_group <- data\_2014 %>%

group\_by(day) %>%

dplyr::summarize(Total = n())

datatable(day\_group)

**Output Screenshot:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.4-day_group.png)

**Code:**

ggplot(day\_group, aes(day, Total)) +

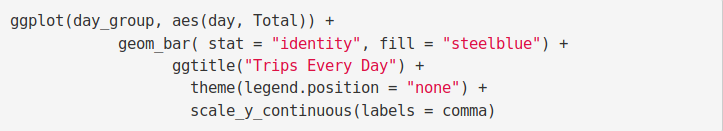
geom\_bar( stat = "identity", fill = "steelblue") +

ggtitle("Trips Every Day") +

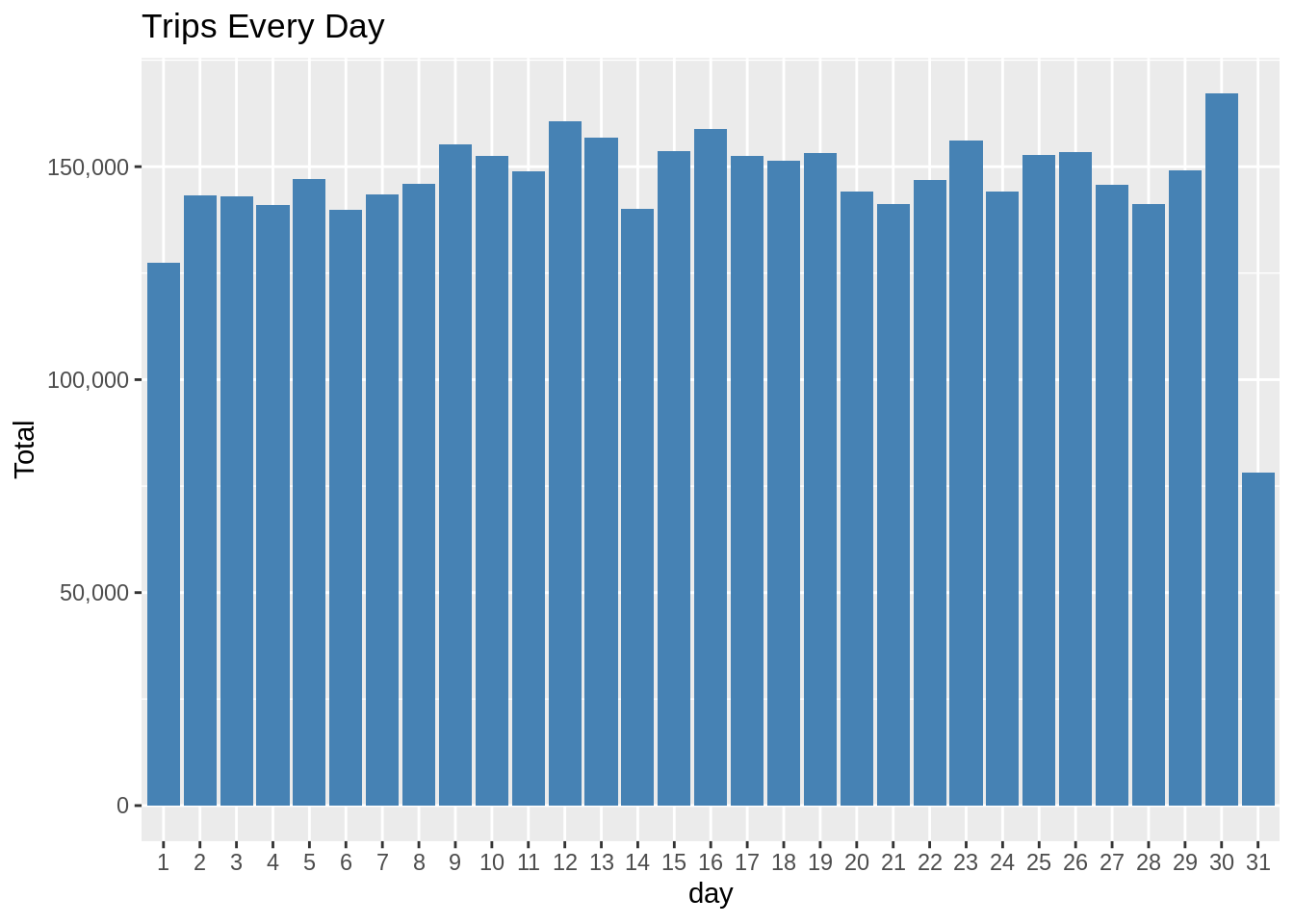
theme(legend.position = "none") +

scale\_y\_continuous(labels = comma)

**Input Screenshot 10:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.5-Trips-Everyday-Code-Plot.png)

**Output:**



**5**

**Code:**

day\_month\_group <- data\_2014 %>%

group\_by(month, day) %>%

dplyr::summarize(Total = n())

ggplot(day\_month\_group, aes(day, Total, fill = month)) +

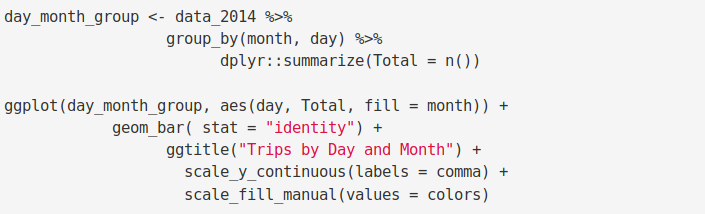
geom\_bar( stat = "identity") +

ggtitle("Trips by Day and Month") +

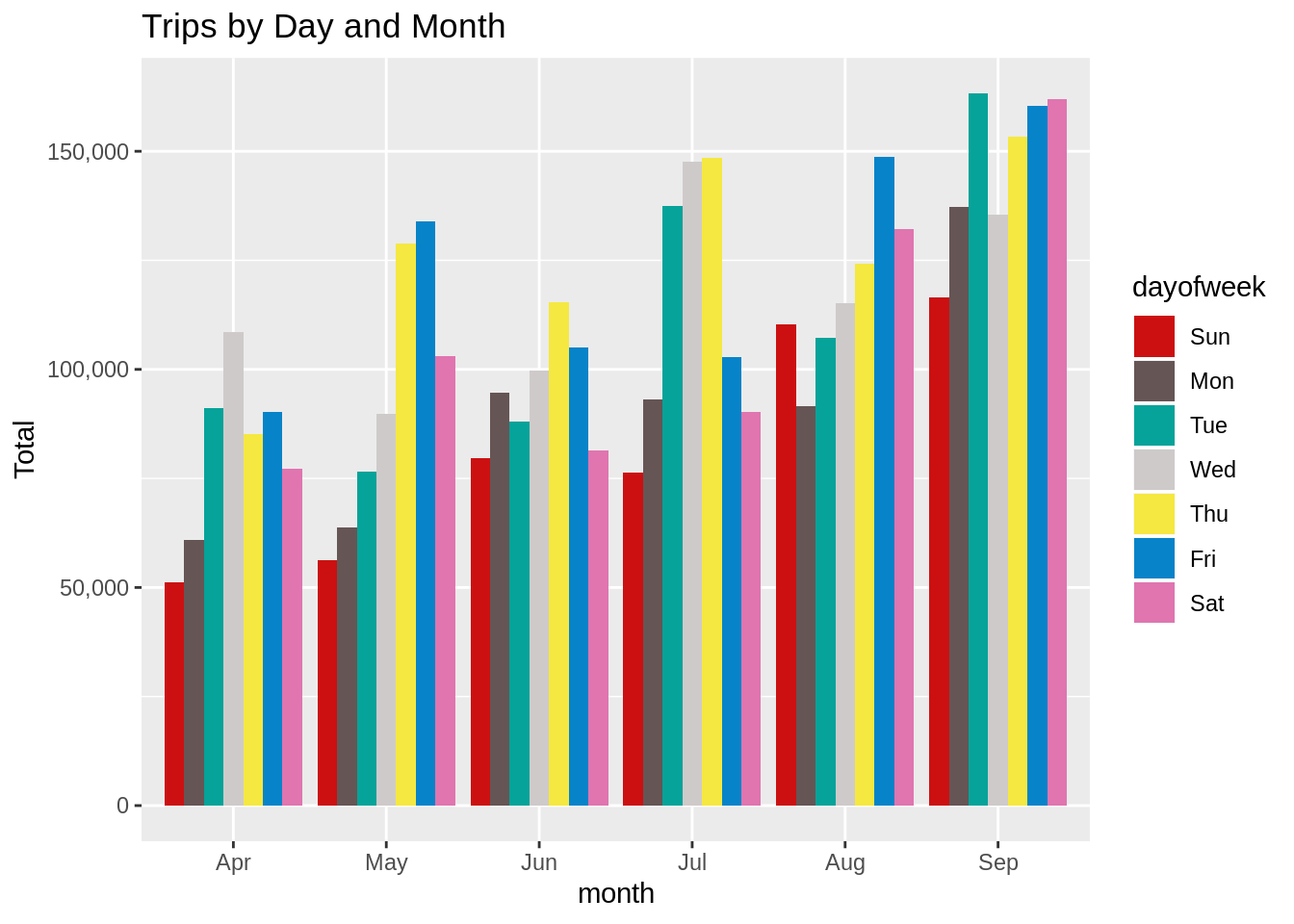
scale\_y\_continuous(labels = comma) +

scale\_fill\_manual(values = colors)

**Input Screenshot 11:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.6-day-month-group.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Trips-by-Days-and-Months1.png)

### **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.

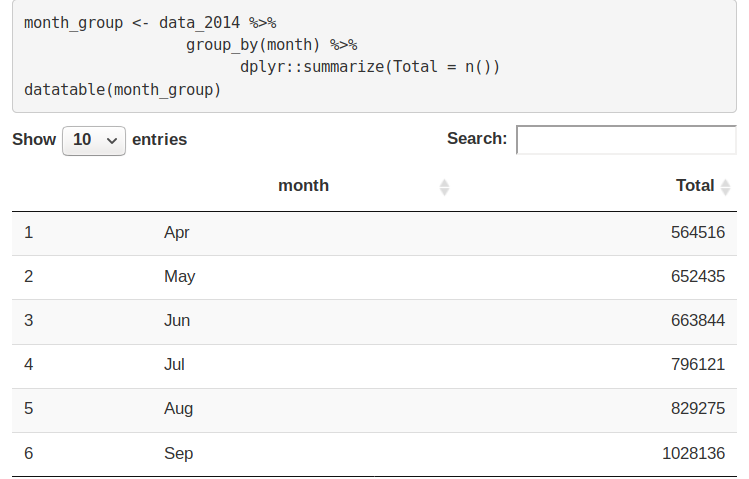
**Code:**

month\_group <- data\_2014 %>%

group\_by(month) %>%

dplyr::summarize(Total = n())

datatable(month\_group)

**Output Screenshot:**[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.7-month_group.png)

**Code:**

ggplot( , aes(month, Total, fill = month)) +

geom\_bar( stat = "identity") +

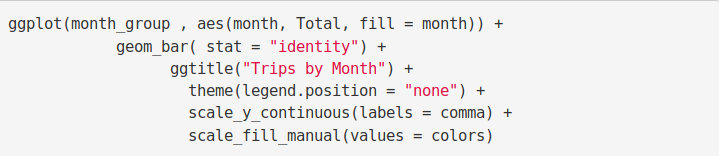
ggtitle("Trips by Month") +

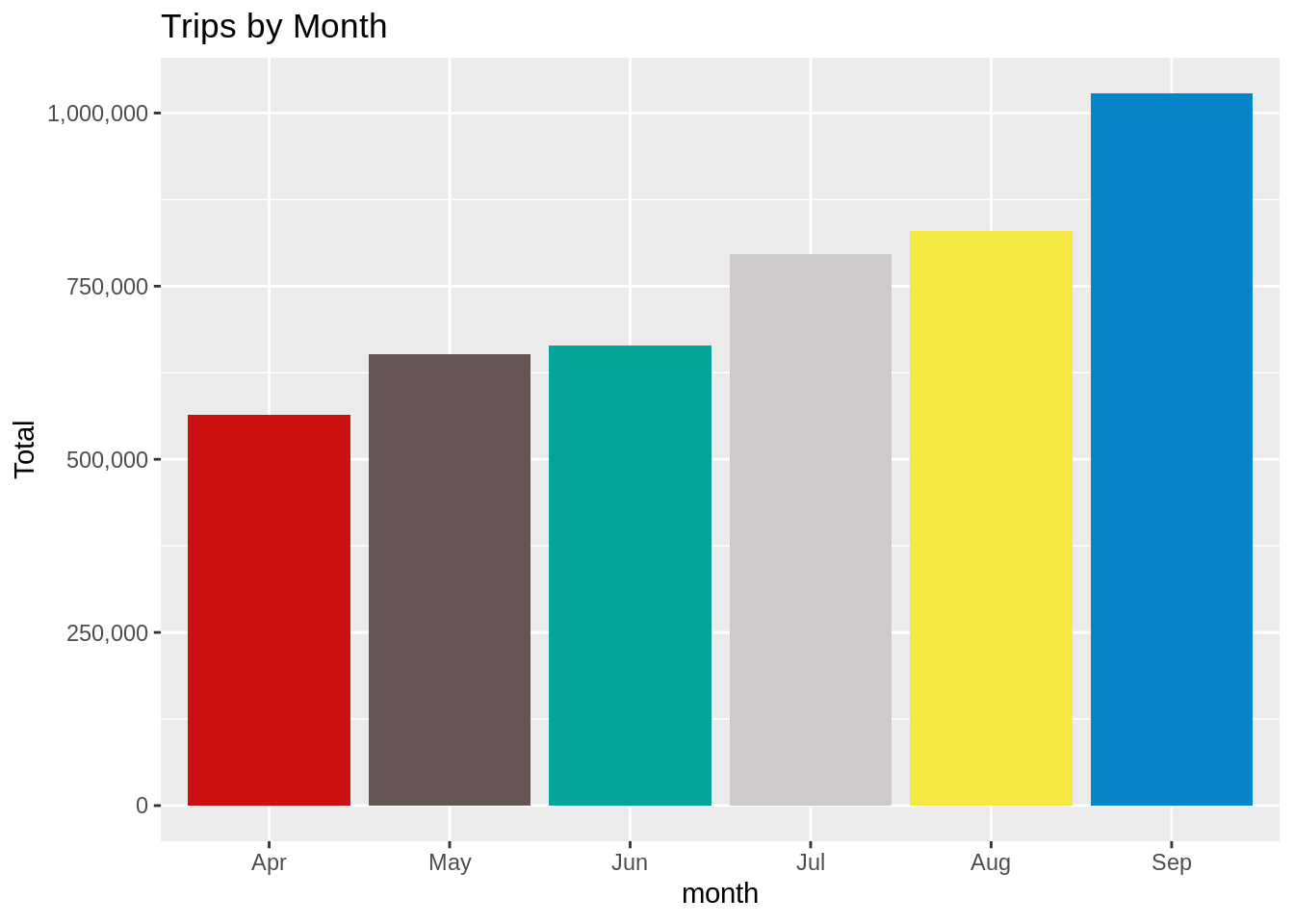
theme(legend.position = "none") +

scale\_y\_continuous(labels = comma) +

scale\_fill\_manual(values = colors)

**Input Screenshot 12:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.8-Trips-by-Month-Plot.png)

**Output:**[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Trips-By-Month.png)

month\_weekday <- data\_2014 %>%

group\_by(month, dayofweek) %>%

dplyr::summarize(Total = n())

ggplot(month\_weekday, aes(month, Total, fill = dayofweek)) +

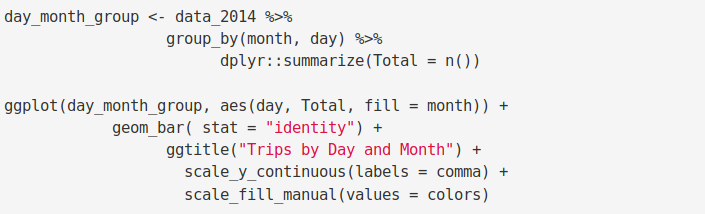
geom\_bar( stat = "identity", position = "dodge") +

ggtitle("Trips by Day and Month") +

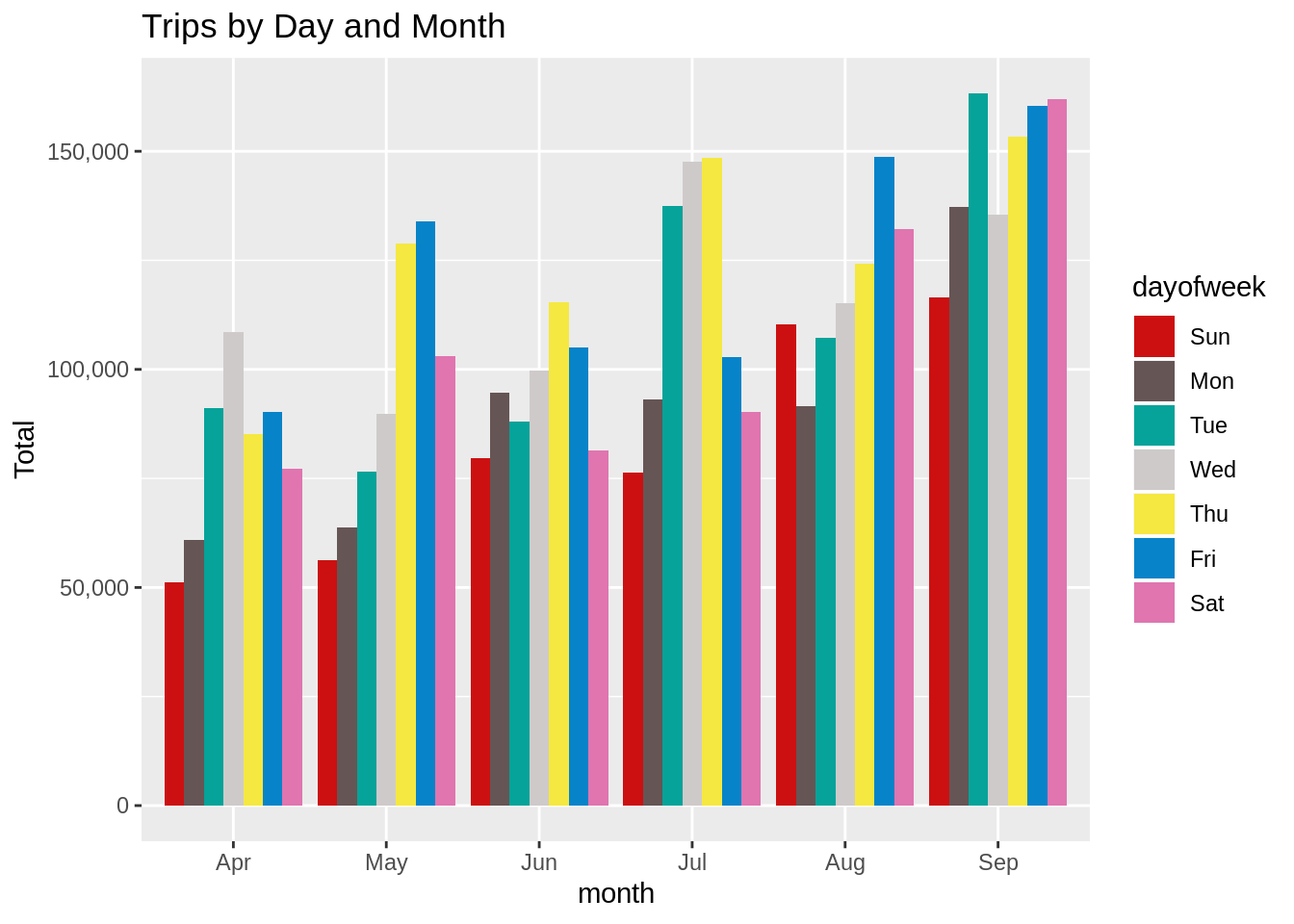
scale\_y\_continuous(labels = comma) +

scale\_fill\_manual(values = colors)

**Input Screenshot 13:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/3.6-day-month-group-1.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Trips-by-Days-and-Months.png)

### **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.

**Code:**

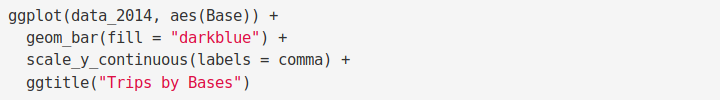
ggplot(data\_2014, aes(Base)) +

geom\_bar(fill = "darkred") +

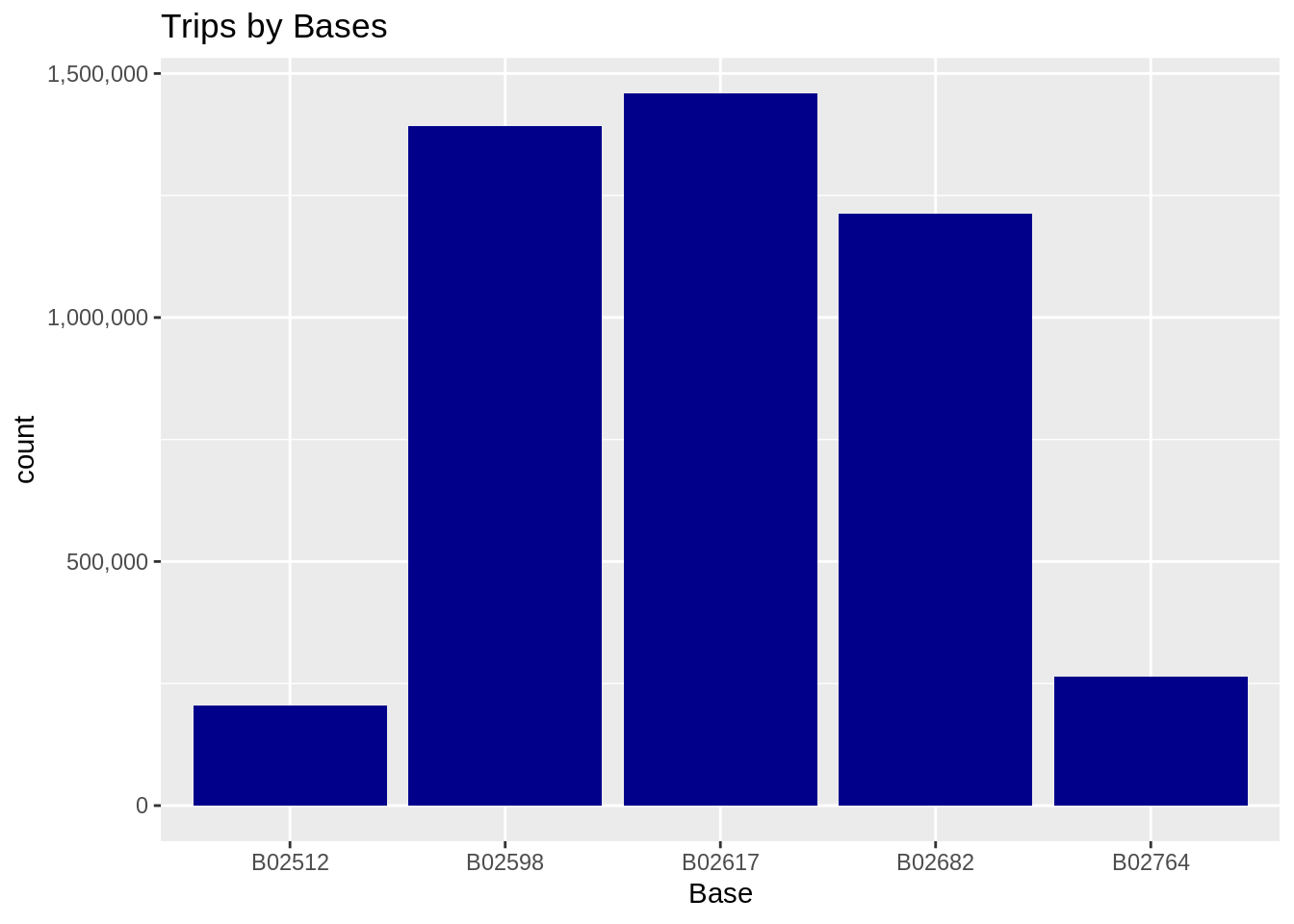
scale\_y\_continuous(labels = comma) +

ggtitle("Trips by Bases")

**Input Screenshot 14:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/4.0-Trips-by-Bases.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Number-of-Trips-by-Bases-Plot.png)

**Code:**

ggplot(data\_2014, aes(Base, fill = month)) +

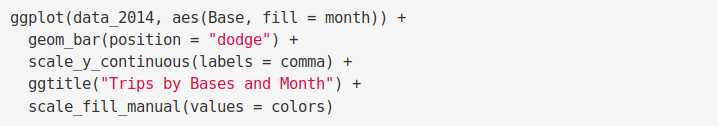
geom\_bar(position = "dodge") +

scale\_y\_continuous(labels = comma) +

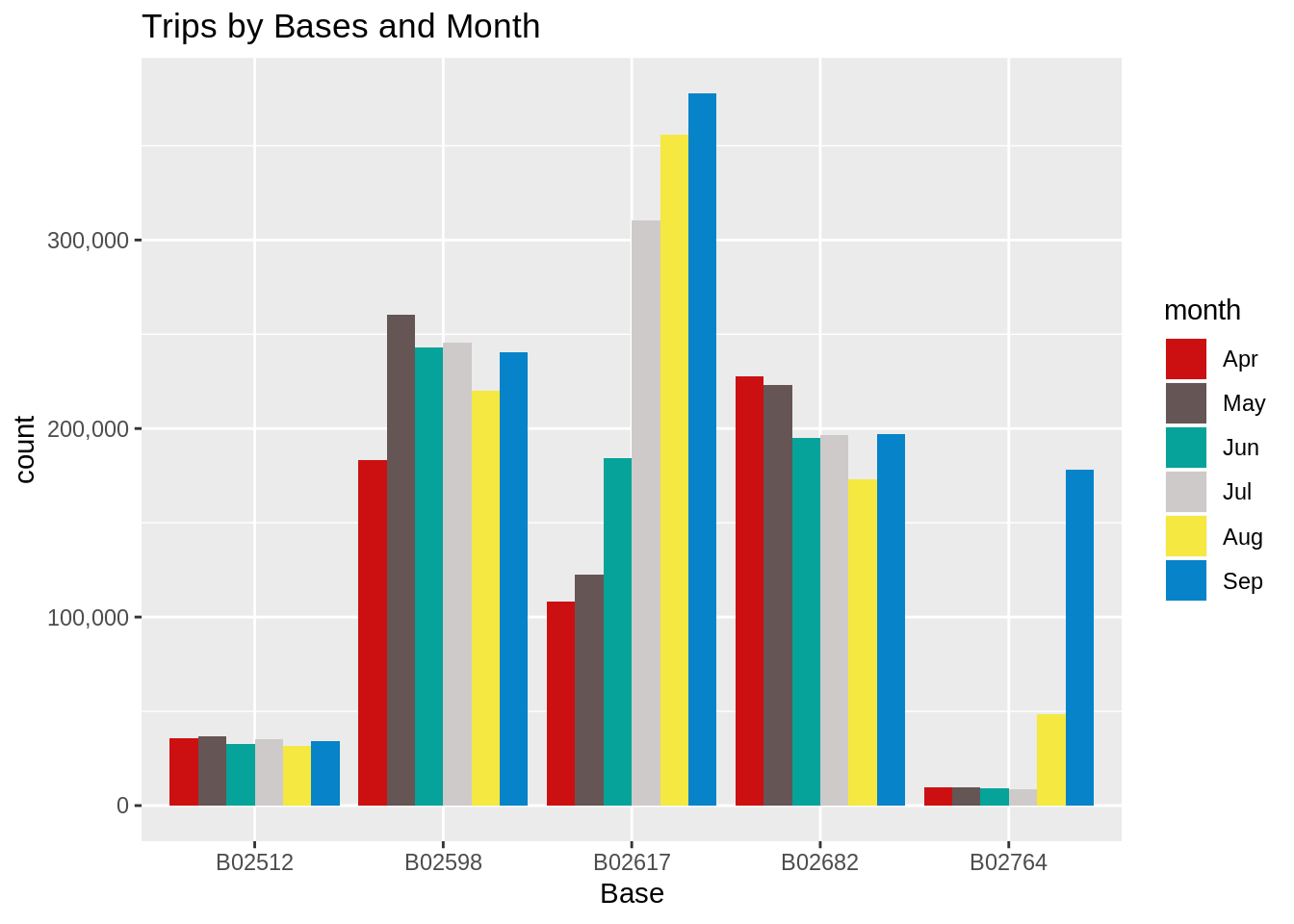
ggtitle("Trips by Bases and Month") +

scale\_fill\_manual(values = colors)

**Input Screenshot 15:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/4.1-Trips-by-Bases-and-Months.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Trips-by-Bases-and-Months-Plot-Output1.png)

**Code:**

ggplot(data\_2014, aes(Base, fill = dayofweek)) +

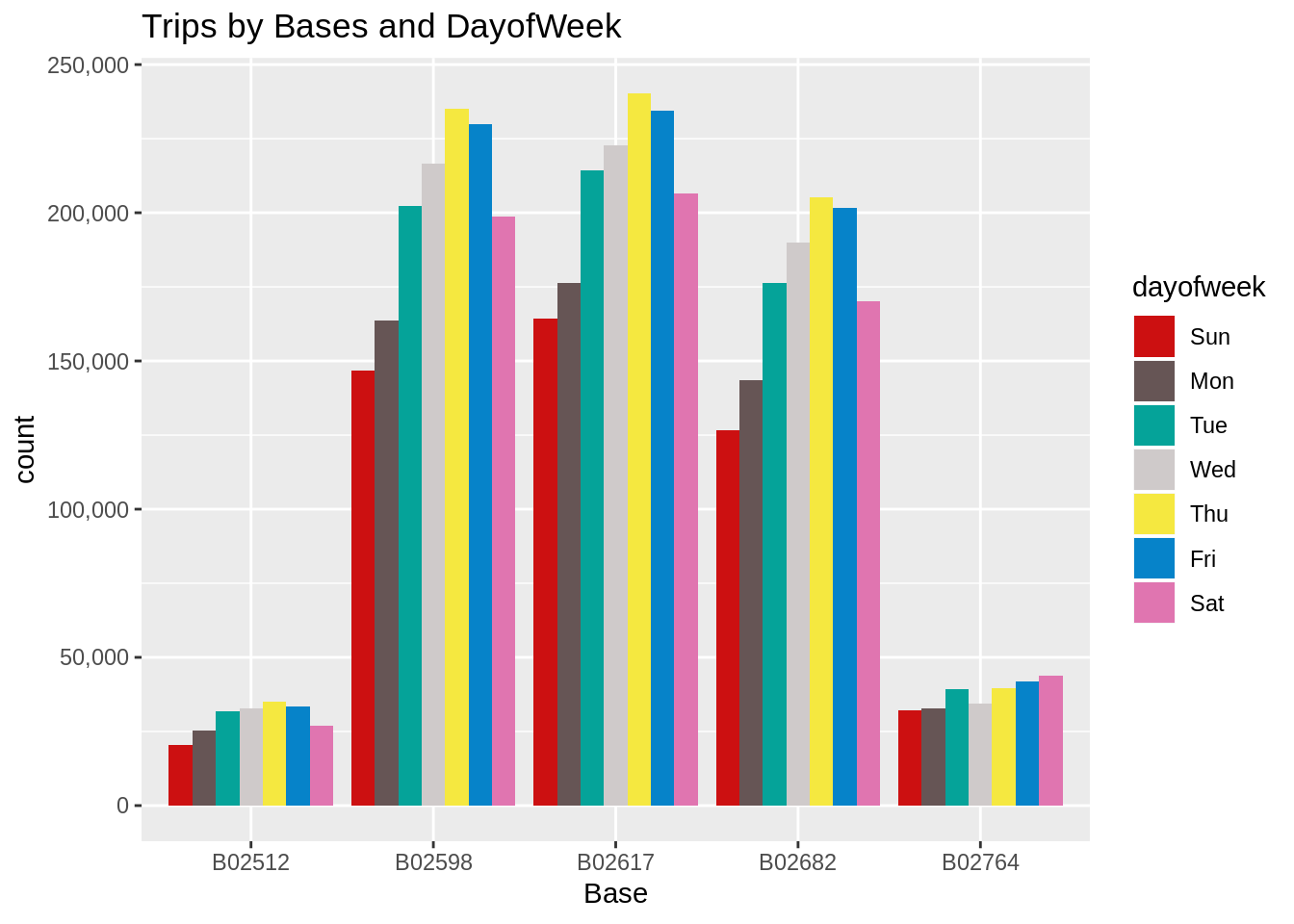
geom\_bar(position = "dodge") +

scale\_y\_continuous(labels = comma) +

ggtitle("Trips by Bases and DayofWeek") +

scale\_fill\_manual(values = colors)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Trips-by-Bases-and-Days-of-the-Week-Output.png)

### **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](https://en.wikipedia.org/wiki/Heat_map" \t "_blank) 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.

**Code:**

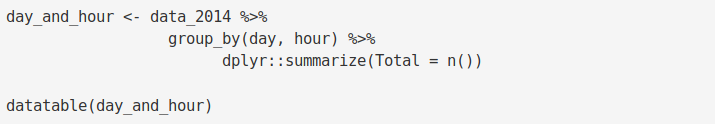
day\_and\_hour <- data\_2014 %>%

group\_by(day, hour) %>%

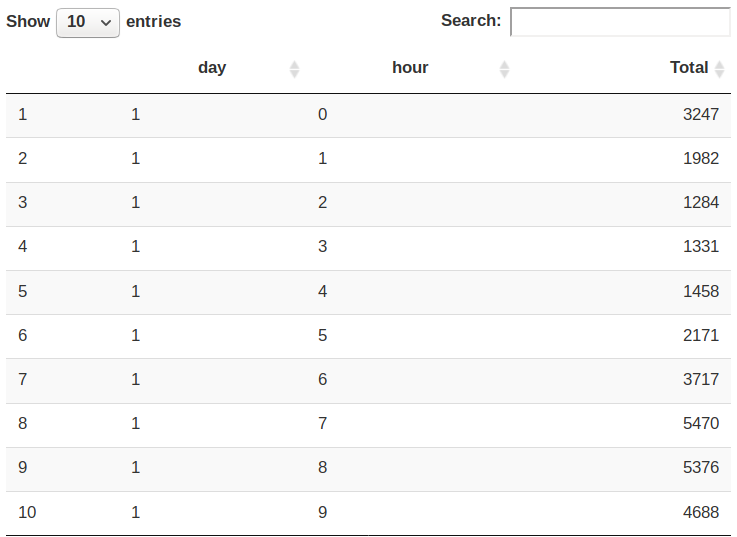
dplyr::summarize(Total = n())

datatable(day\_and\_hour)

**Input Screenshot 16:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/4.2-day_hour_month-code.png)

**Output Screenshot:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/4.2-day_hour_month-code-Output.png)

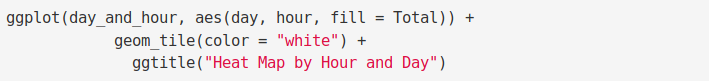
**Code:**

ggplot(day\_and\_hour, aes(day, hour, fill = Total)) +

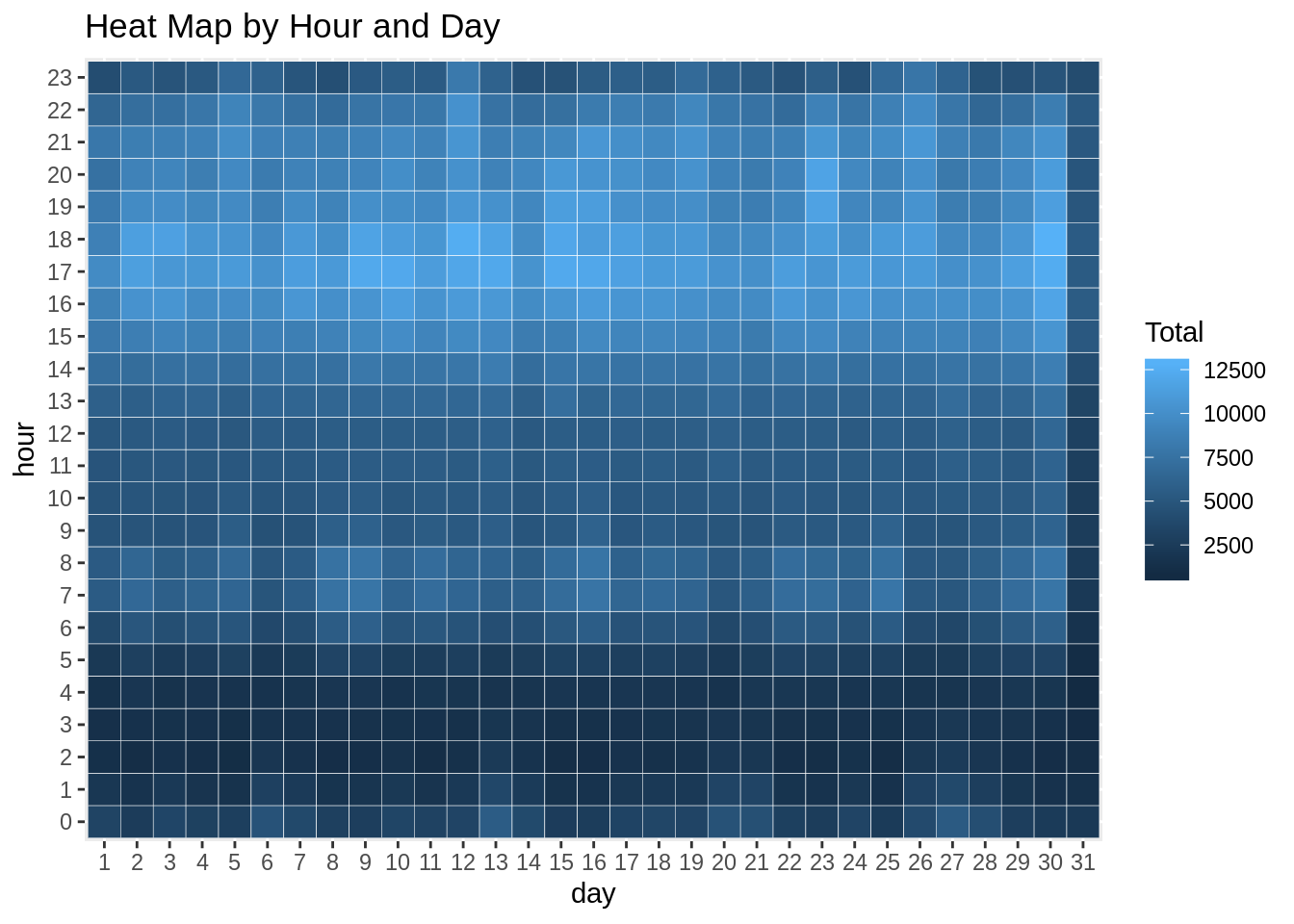
geom\_tile(color = "white") +

ggtitle("Heat Map by Hour and Day")

**Input Screenshot 17:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/5.0-heatmap-hour_day.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/HeatMap-Hour-and-Day-Output.png)

**Code:**

ggplot(day\_month\_group, aes(day, month, fill = Total)) +

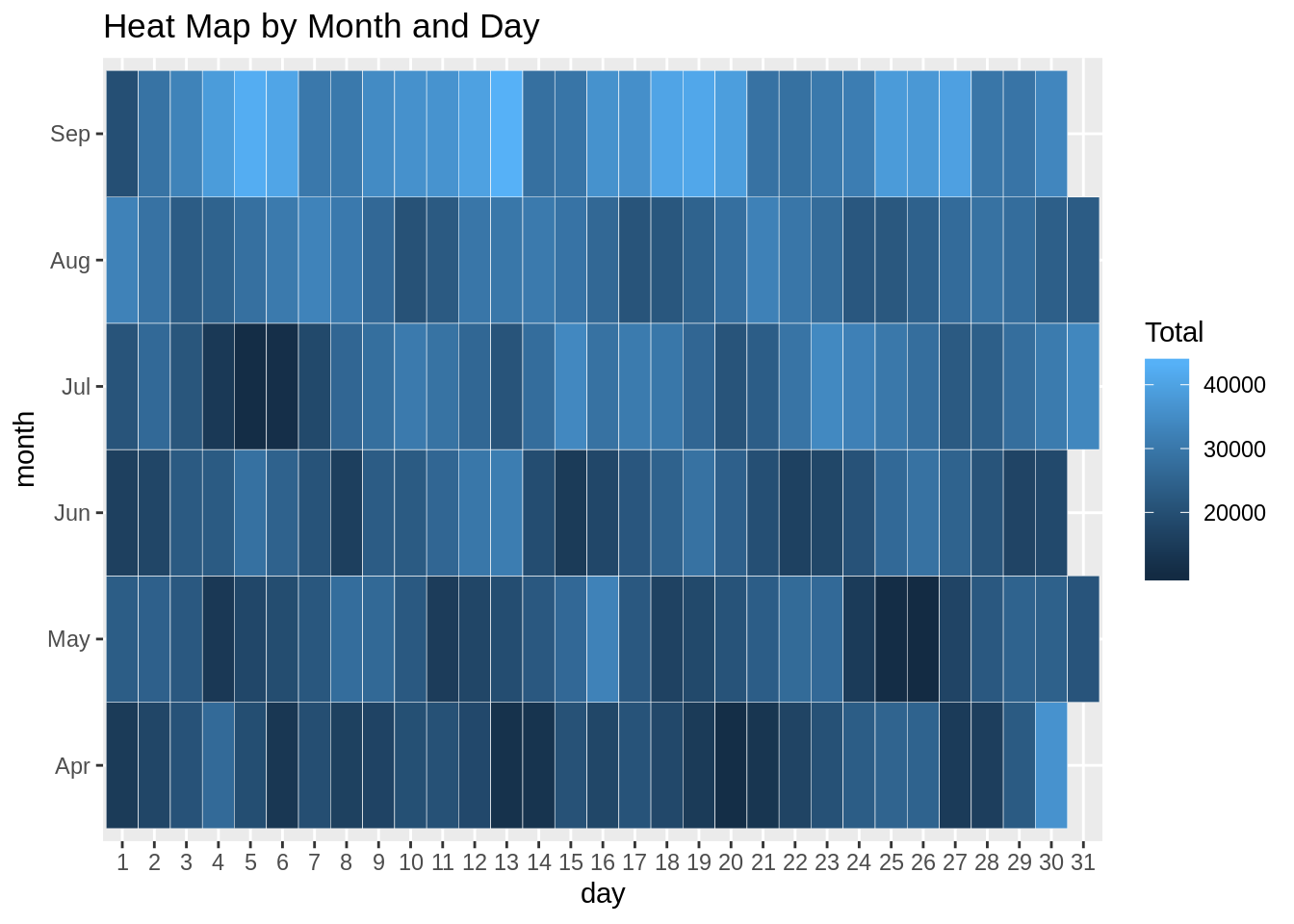
geom\_tile(color = "white") +

ggtitle("Heat Map by Month and Day")

**Input Screenshot 18:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/5.1-Heatmap-Month-and-Day.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Month-and-Day-Output.png)

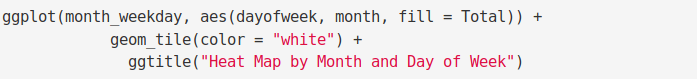
**Code:**

ggplot(month\_weekday, aes(dayofweek, month, fill = Total)) +

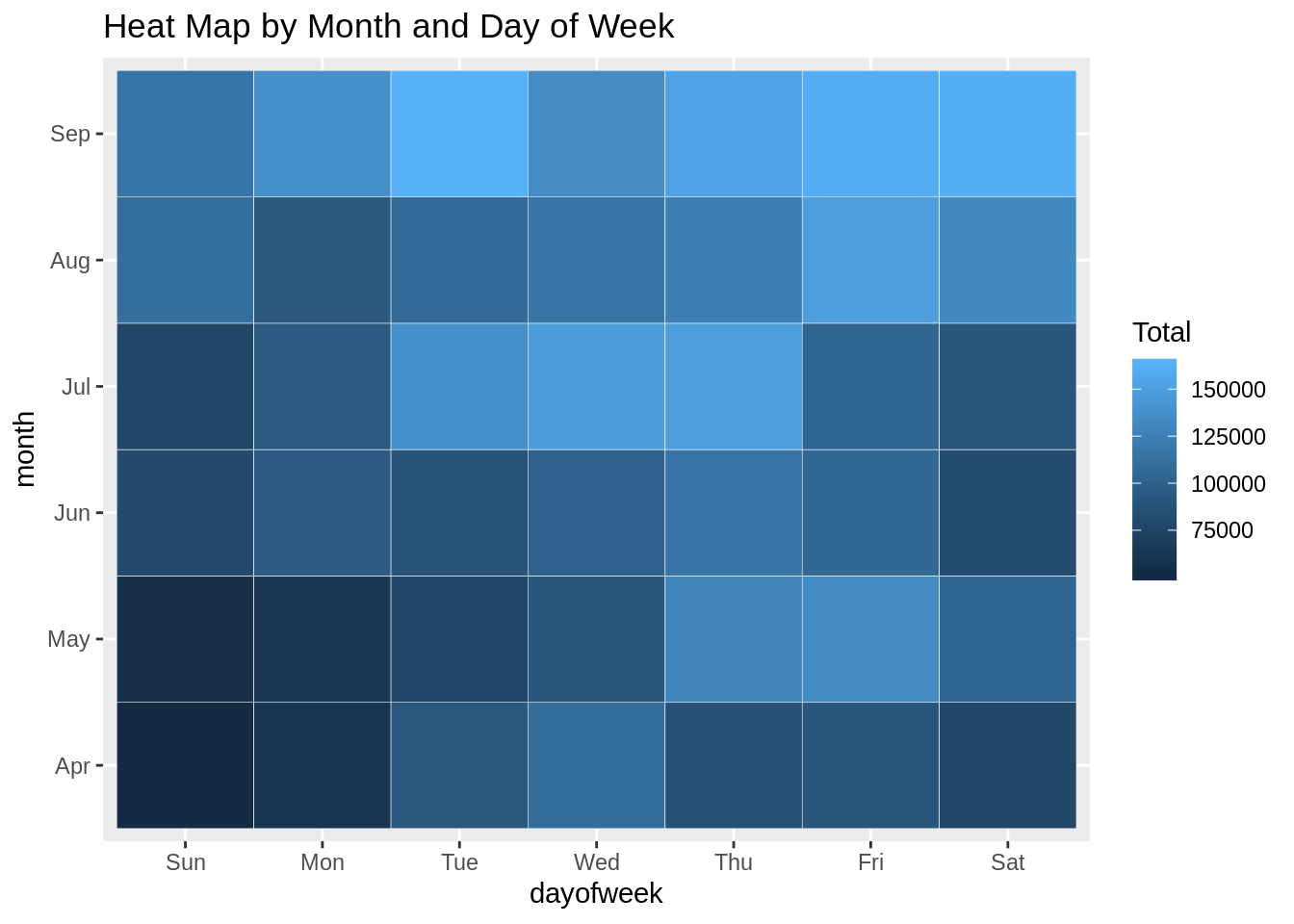
geom\_tile(color = "white") +

ggtitle("Heat Map by Month and Day of Week")

**Input Screenshot 19:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/5.2-Heatmap-Month-and-Day-of-Week.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Month-and-Day-of-Week-Output.png)

**Code:**

month\_base <- data\_2014 %>%

group\_by(Base, month) %>%

dplyr::summarize(Total = n())

day0fweek\_bases <- data\_2014 %>%

group\_by(Base, dayofweek) %>%

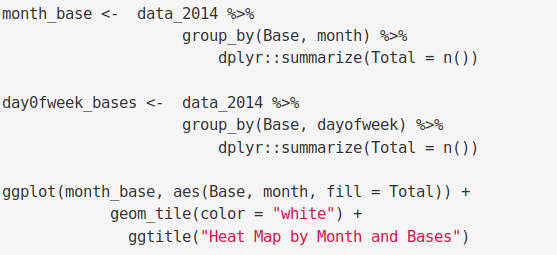
dplyr::summarize(Total = n())

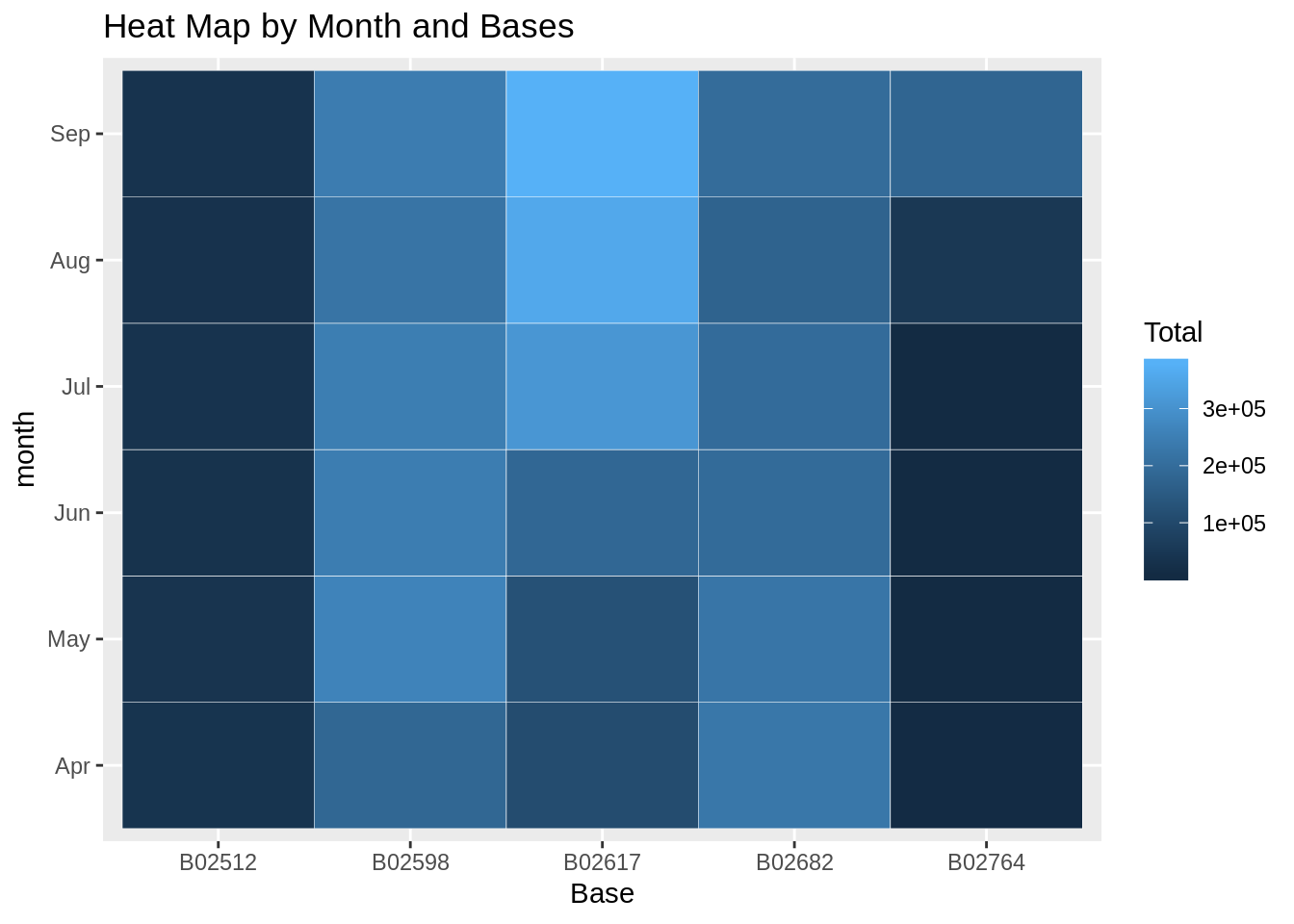
ggplot(month\_base, aes(Base, month, fill = Total)) +

geom\_tile(color = "white") +

ggtitle("Heat Map by Month and Bases")

**Input Screenshot 20:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/5.3-Heatmap-Month-and-Bases.png)  
**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Month-and-Bases-Output.png)

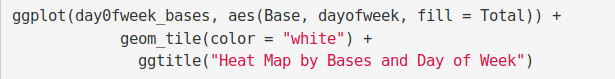
**Code:**

ggplot(day0fweek\_bases, aes(Base, dayofweek, fill = Total)) +

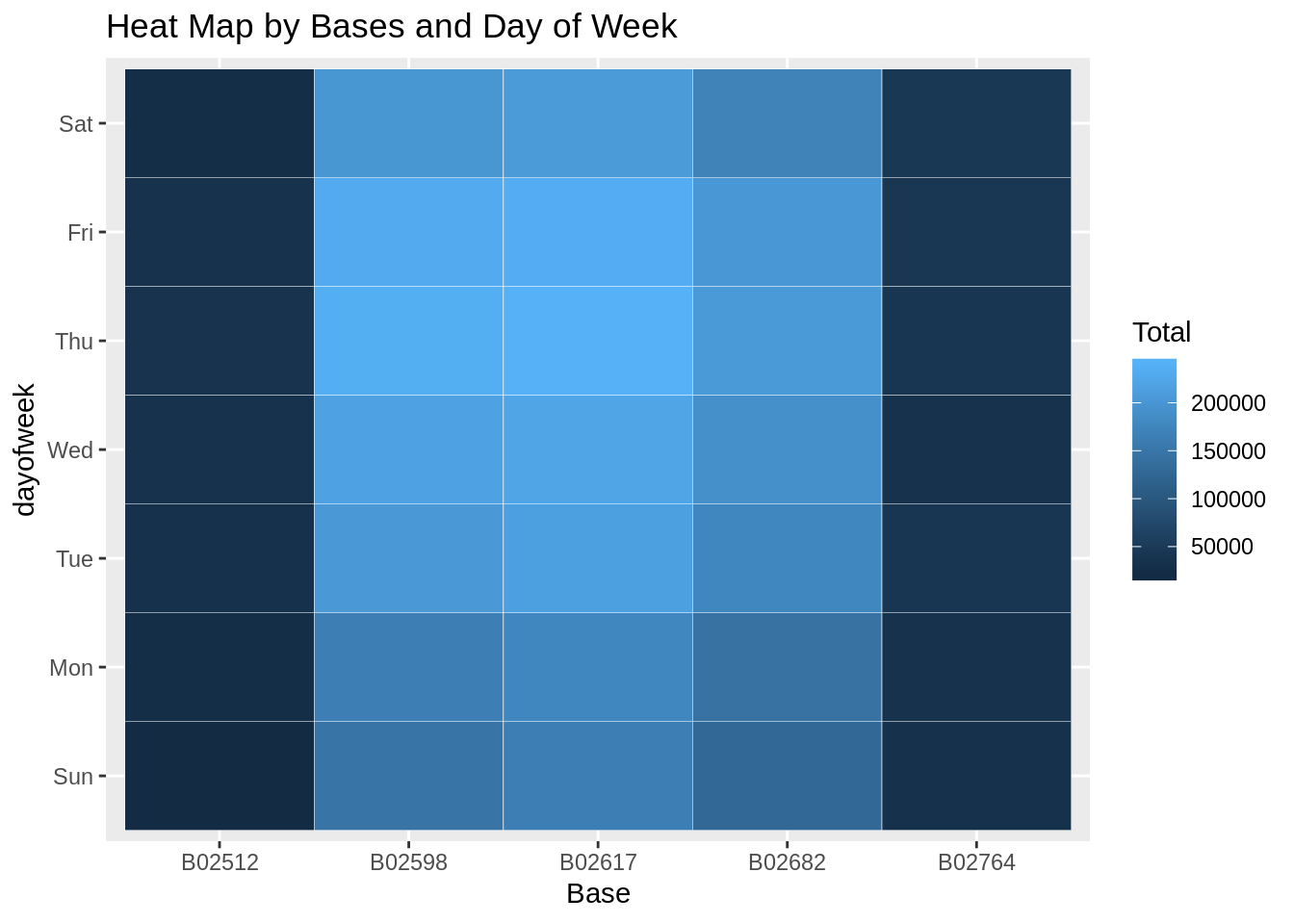
geom\_tile(color = "white") +

ggtitle("Heat Map by Bases and Day of Week")

**Input Screenshot 21:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/5.4-Days-and-Base-of-Week.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Bases-and-Days-of-Weeks-Output.png)

### **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.

**Code:**

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) +

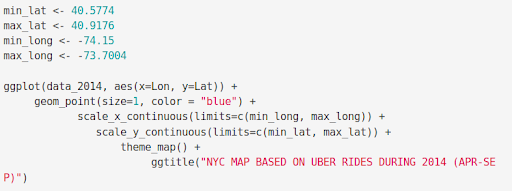
scale\_x\_continuous(limits=c(min\_long, max\_long)) +

scale\_y\_continuous(limits=c(min\_lat, max\_lat)) +

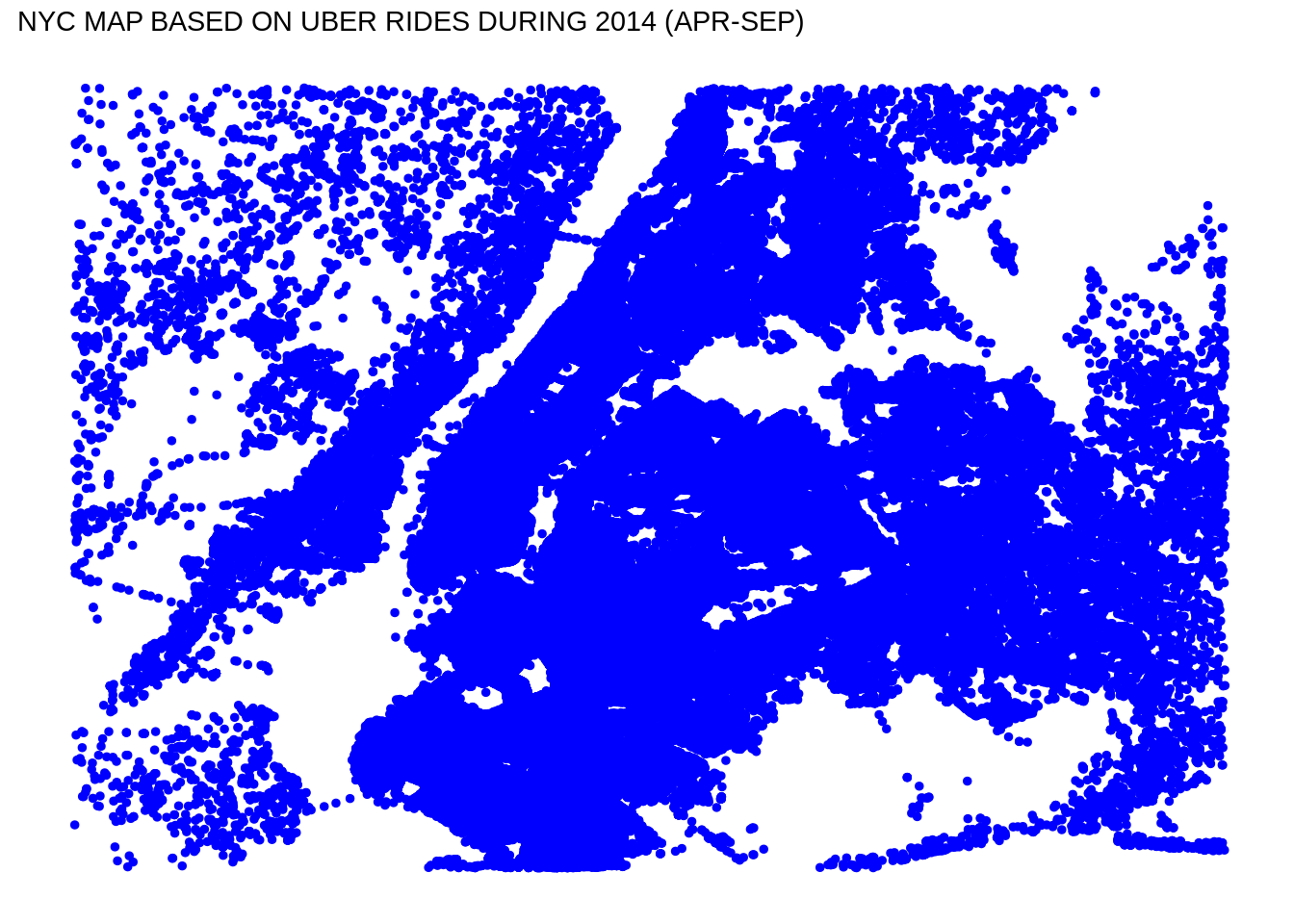
theme\_map() +

ggtitle("NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP) by BASE")

**Input Screenshot 22:**

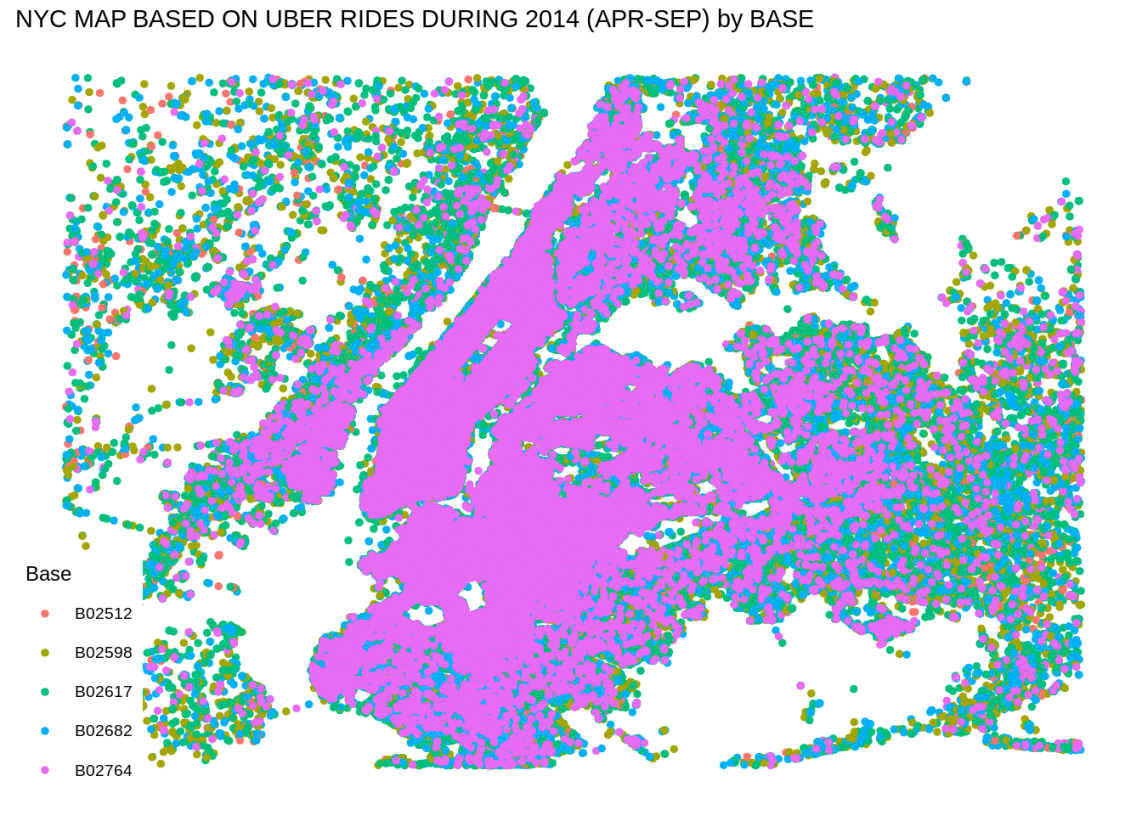
[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/creating-a-map-visualization.png)

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/NYC-Map-1-Output.png)

*Uber data analysis using R*

**Output:**

[](https://data-flair.training/blogs/wp-content/uploads/sites/2/2019/07/Map-2-Output.png)

**Which algorithm does Uber use for Data Analysis?**

Every ride booked on Uber gives their team a large amount of information, including the riders booking preferences, pickup, and drop-off trends, availability of drivers in the area, traffic patterns, ride ETA, duration, speed, weather factors, and more. Uber uses this data to train a multitude of machine learning algorithms like the ones discussed in this blog for various purposes. Some popular uses include calculating a competitive fare to maximize profits (using predictive modelling algorithms), estimating surge prices (using a model called “Geosurge”), tuning the requirements of drivers in a particular region, catching fake rides, and fake drivers, and estimating ride info like ETA.

**Project Outcome: -**

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.

**Project 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.

**14**