Uber data Visualisation

Bernard Kymerlive

2024-10-10

Introduction

The Uber data visualization project aims to explore and analyze patterns in Uber trip data. By leveraging powerful data visualization techniques in R, such as ggplot2, we can gain insights into the temporal and spatial distribution of rides. This project focuses on creating meaningful visual representations of the data to better understand trends in ride frequency, peak times, and base-wise performance. With a vast dataset containing information on ride times, locations, and driver bases, this analysis is key in uncovering operational efficiencies and user behavior patterns

Problem Statement

Uber operates a complex system of ride-sharing services across various locations, and understanding the temporal and spatial distribution of trips is crucial for improving operational efficiency and customer satisfaction. However, due to the large volume of data, it becomes difficult to manually discern meaningful trends. This project addresses the problem by using data visualization to identify patterns such as peak ride times, the distribution of trips across different bases, and the busiest days of the week. By visualizing these patterns, Uber can make informed decisions to optimize driver allocation, enhance customer experience, and improve service performance across different regions and timeframes.

Significance of the Study

The significance of this study lies in its potential to provide actionable insights for Uber's operational strategy and decision-making processes. By visualizing key patterns in trip data, this study helps identify peak demand periods, busiest locations, and trends in ride frequency. These insights can aid Uber in optimizing driver distribution, reducing wait times for passengers, and enhancing overall service efficiency. Furthermore, the study provides a foundation for predictive modeling, allowing Uber to anticipate demand fluctuations and improve resource allocation. The findings can contribute to better customer satisfaction, cost management, and strategic planning within Uber's operational framework.

prerequisites.

```
library(tidyverse)
library(DT)
library(ggthemes)
library(scales)
library(RColorBrewer)
```

Data description and import

The data set used in this project contains a combination of the data describing uber trips from the month of April to September 2014. This final data set has 4,534,327 and four variables.

```
getwd()

## [1] "C:/Users/mcoast/Desktop/R projects/Projects"

setwd("C:/Users/mcoast/Desktop/R projects/data/Uber-dataset")
apr14 <- read.csv("apr14.csv")
aug14 <- read.csv("aug14.csv")
jul14 <- read.csv("jul14.csv")
may14 <- read.csv("may14.csv")
jun14 <- read.csv("jun14.csv")
sep14 <- read.csv("sep14.csv")

data_2014 <- rbind(apr14, may14, jun14, jul14, aug14, sep14)
head(data_2014)

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

Data cleaning

In this step i performed the appropriate formatting of Date.Time column. Then proceeded to create factors of time objects like day, month, year etc.

```
data_2014 <- data_2014 %>%
  rename(date_time = Date.Time) %>%
  rename_with(.fn = tolower) %>%
  mutate(date_time = parse_datetime(date_time, format = "%m/%d/%Y %H:%M:%S"),
         month = factor(month(date_time, label = T)),
         day = factor(day(date_time)),
```

```
         wday = factor(wday(date_time, label = T)),
         year = factor(year(date_time)),
         hour = factor(hour(date_time)),
         minute = factor(minute(date_time)),
         second = factor(second(date_time))
         )

head(data_2014)
```

```
##             date_time     lat      lon   base month day wday year hour
minute
## 1 2014-04-01 00:11:00 40.7690 -73.9549 B02512   Apr   1  Tue 2014    0
11
## 2 2014-04-01 00:17:00 40.7267 -74.0345 B02512   Apr   1  Tue 2014    0
17
## 3 2014-04-01 00:21:00 40.7316 -73.9873 B02512   Apr   1  Tue 2014    0
21
## 4 2014-04-01 00:28:00 40.7588 -73.9776 B02512   Apr   1  Tue 2014    0
28
## 5 2014-04-01 00:33:00 40.7594 -73.9722 B02512   Apr   1  Tue 2014    0
33
## 6 2014-04-01 00:33:00 40.7383 -74.0403 B02512   Apr   1  Tue 2014    0
33
##   second
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      0
```

Plotting the trips by hours in a day

In the next step of the R project, the ggplot2 function was used to plot the number of trips made by passengers in a day. The dplyr package was also utilized to aggregate the data. From the resulting visualizations, it was observed that the number of passengers fluctuated throughout the day, with a higher number of trips in the evening, around 5:00 and 6:00 PM

```
hour_data <- data_2014 %>%
  group_by(hour) %>%
  summarise(total = n())

datatable(hour_data)
```

Show 10 entries

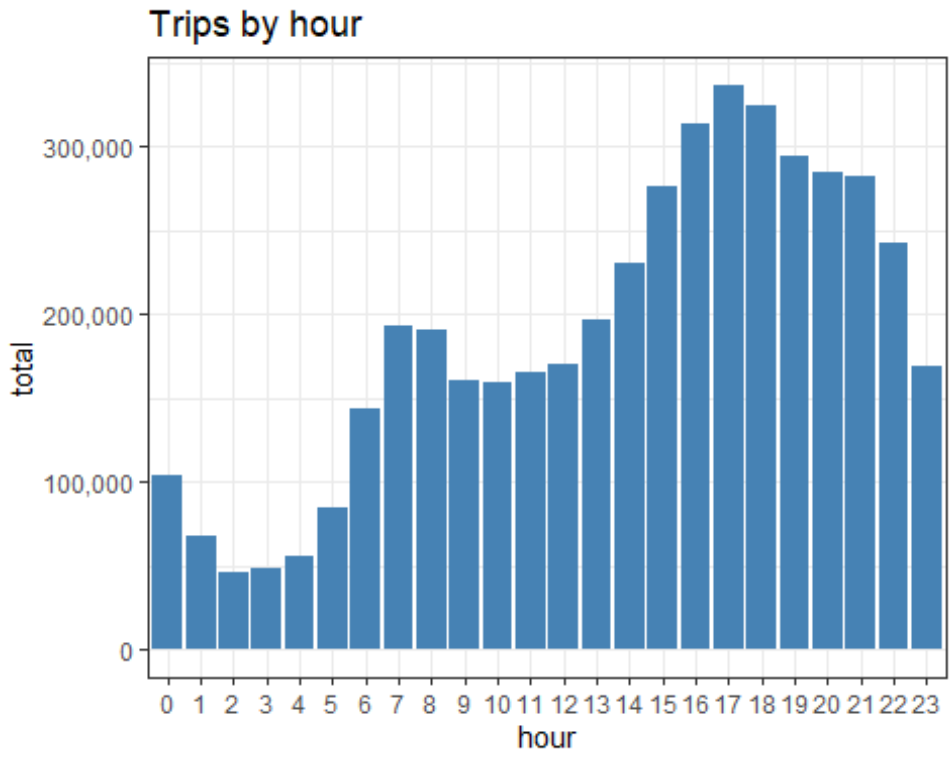
Search:

	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

```
ggplot(hour_data, aes(hour, total)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  ggtitle("Trips by hour") +
  scale_y_continuous(label = comma) +
  theme_bw()
```



PLotting data by trips during every day of the month

A visualisation of the trips made evryday of the month shows 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) %>%
  summarise(total = n())

datatable(day_group)
```

Show 10 entries

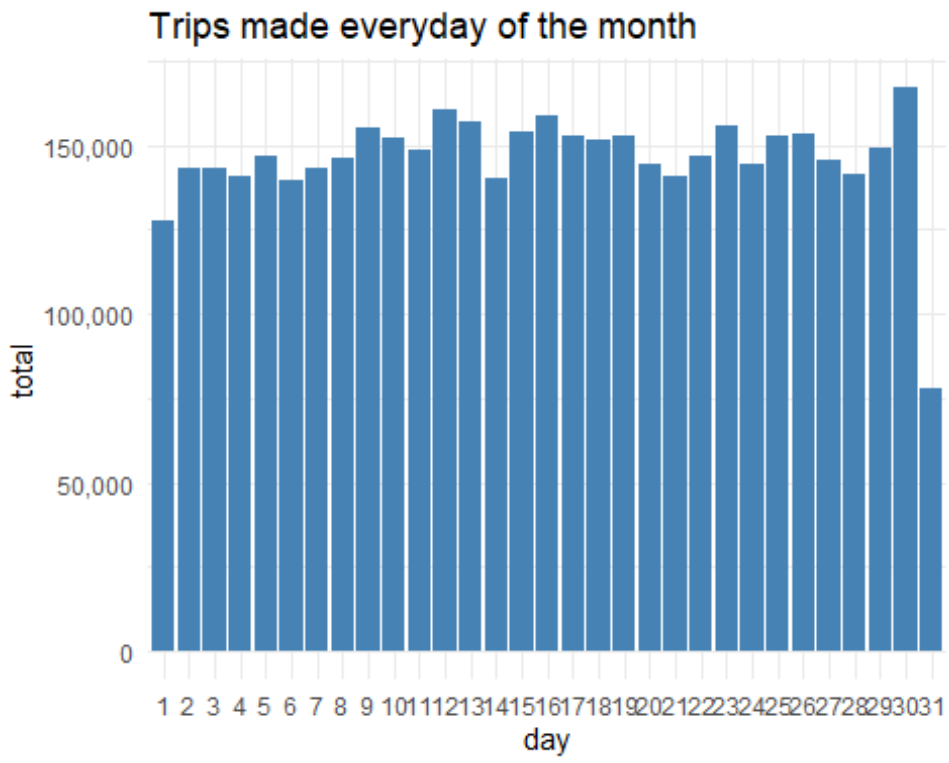
Search:

	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

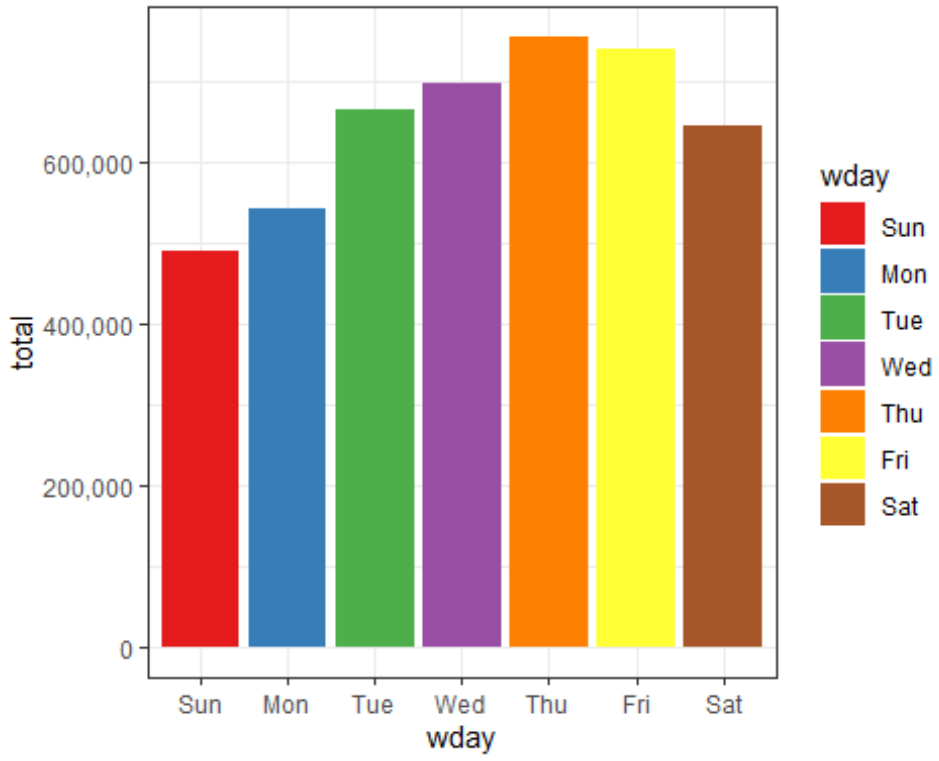
Previous 1 2 3 4 Next

```
ggplot(day_group, aes(day, total)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  scale_y_continuous(label = comma) +
  theme_minimal() +
  ggtitle("Trips made everyday of the month")
```



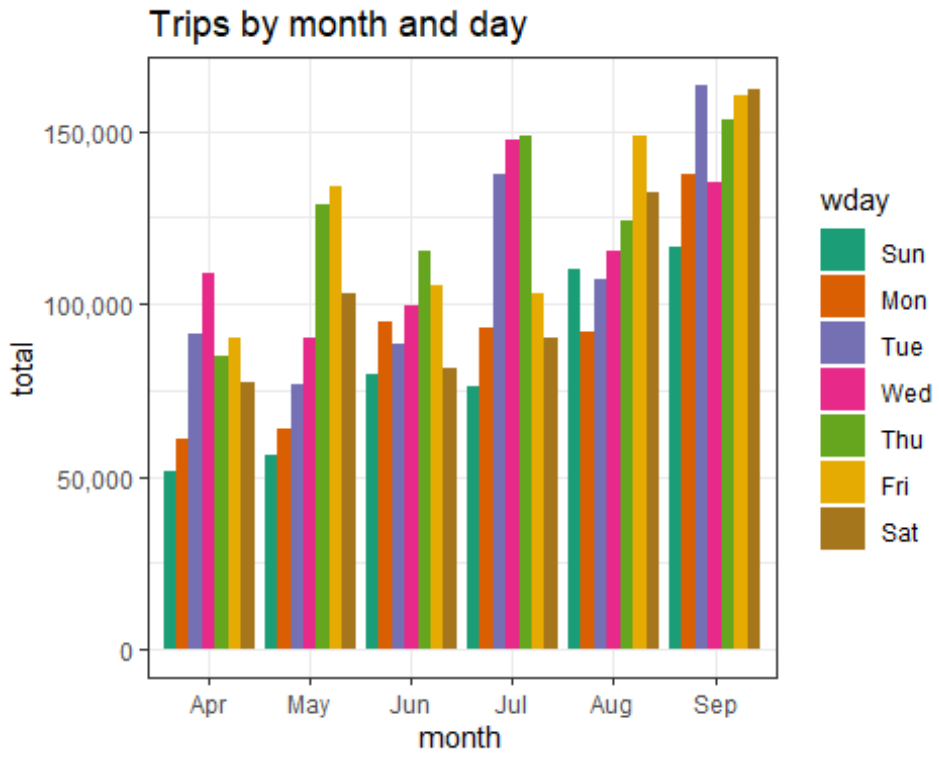
Plotting Trips made evry day of the week

```
day_of_week <- data_2014 %>%
  group_by(wday) %>%
  summarise(total = n())

ggplot(day_of_week, aes(wday, total)) +
  geom_bar(stat = "identity", aes(fill = wday)) +
  scale_fill_brewer(palette = "Set1") +
  scale_y_continuous(labels = comma) +
  theme_bw()
```



Clearly it can be seen that through out the year most trips are made on friday and thursday and that the least trips are made on sunday. Also a visualisation was made to investigate how trips are made during every day of the week for every month.

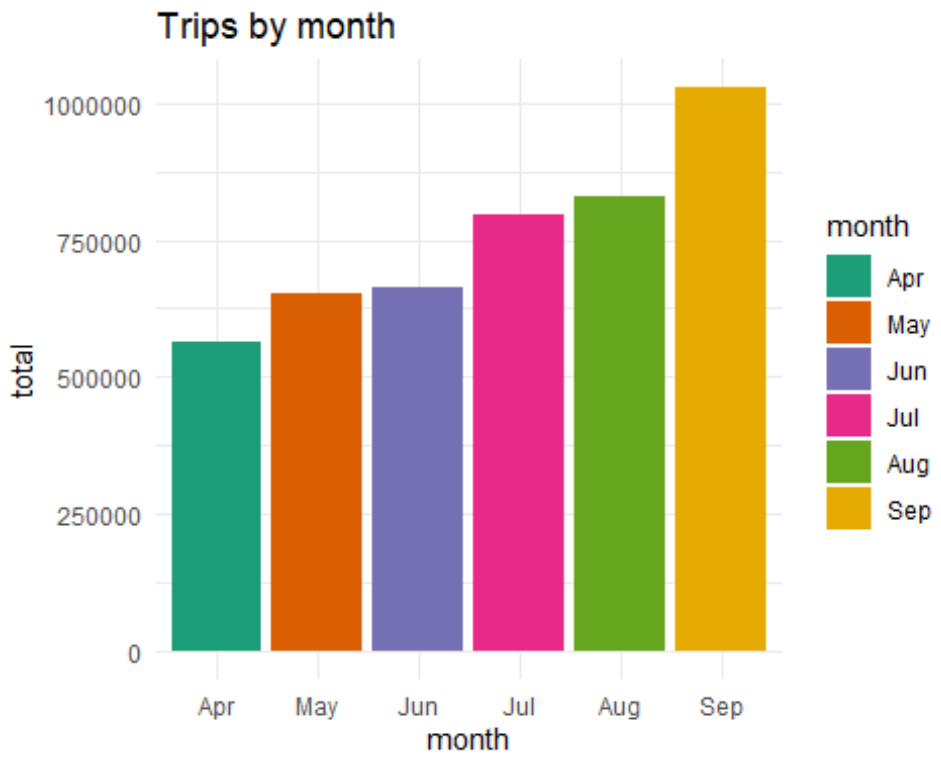
```
wday_month <- data_2014 %>%
  group_by(month, wday) %>%
  summarise(total = n())

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

ggplot(wday_month, aes(month, total)) +
  geom_bar(stat = "identity", aes(fill = wday), position = "dodge") +
  scale_y_continuous(labels = comma) +
  scale_fill_brewer(palette = "Dark2") +
  theme_bw() +
  ggtitle("Trips by month and day")
```



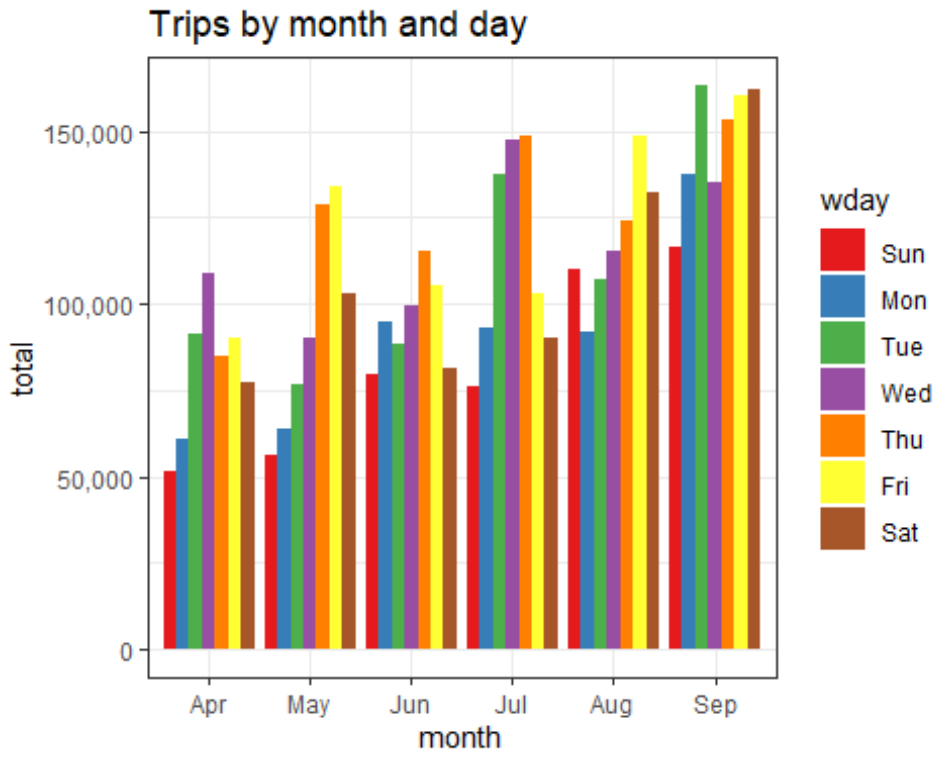
Number of Trips taking place during months in a year

In this section, the number of trips taking place each month of the year was visualized. In the output visualization, it was observed that most trips occurred during the month of September. Additionally, visual reports were generated showing the number of trips made on each day of the week.

```
month_group <- data_2014 %>%
  group_by(month) %>%
  summarise(total = n())

ggplot(month_group, aes(month, total)) +
  geom_bar(stat = "identity", aes(fill = month)) +
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal() +
  ggtitle("Trips by month")
```



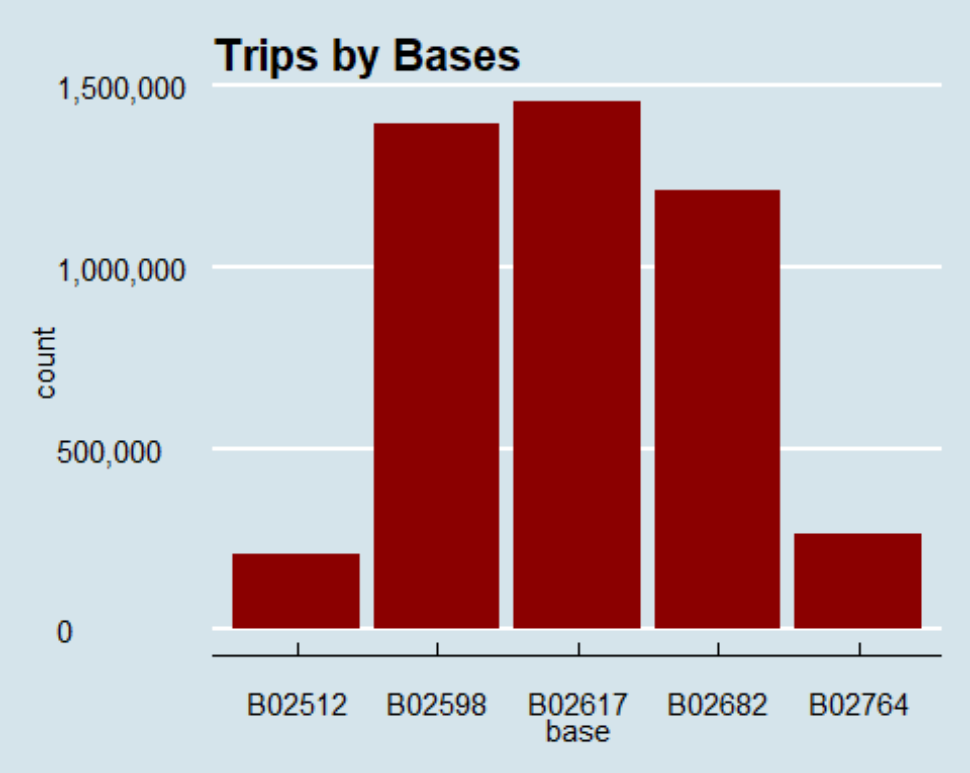

```
wday_month <- data_2014 %>%
  group_by(month, wday) %>%
  summarise(total = n())

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

ggplot(wday_month, aes(month, total)) +
  geom_bar(stat = "identity", aes(fill = wday), position = "dodge") +
  scale_y_continuous(labels = comma) +
  scale_fill_brewer(palette = "Set1") +
  theme_bw() +
  ggtitle("Trips by month and day")
```



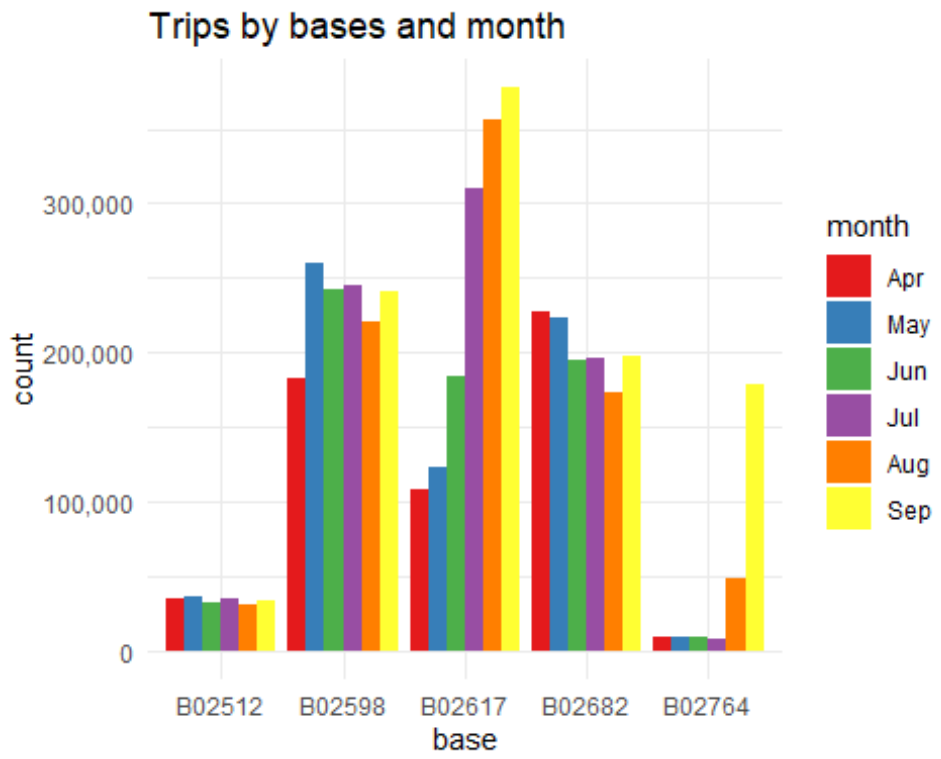
Finding out the number of trips by bases

In the following visualizations, the number of trips taken by passengers from each base was plotted. There are five bases in total, and it was observed that B02617 had the highest number of trips. Additionally, B02617 recorded the highest number of trips during the month. Thursday showed the highest number of trips for three bases: B02598, B02617, and B02682.

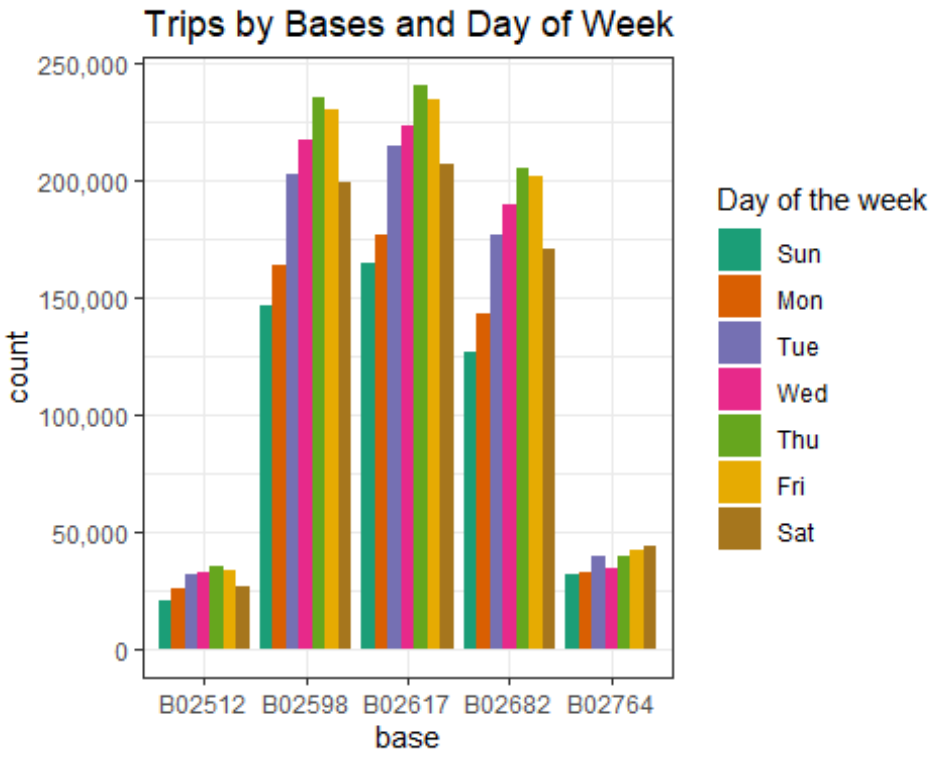
```
ggplot(data_2014, aes(base)) +
  geom_bar(fill = "darkred") +
  scale_y_continuous(label = comma) +
  ggtitle("Trips by Bases") +
  theme_economist()
```




```r
ggplot(data_2014, aes(base)) +
  geom_bar(aes(fill = month), position = "dodge") +
  scale_y_continuous(label = comma) +
  scale_fill_brewer(palette = "Set1") +
  theme_minimal() +
  ggtitle("Trips by bases and month")
```




```
ggplot(data_2014, aes(base)) +
  geom_bar(aes(fill = wday), position = "dodge") +
  scale_fill_brewer(palette = "Dark2") +
  theme_bw() +
 scale_y_continuous(labels = comma) +
 ggtitle("Trips by Bases and Day of Week") +
  labs(fill = "Day of the week")
```


Trips by Bases and Day of Week
250,000
200,000
150,000
100,000
50,000
0
count
B02512
B02598
B02617
B02682
B02764
base
Day of the week
Sun
Mon
Tue
Wed
Thu
Fri
Sat