Uber

Arun Solviya

7/11/2021

Introduction

This project is the analysis of UBER rides data.The analysis is performed on the public data set of UBER rides from April 2014 to September 2014. The analysis focuses to find the peak time of the rides made with respect to months, week days and hours of the day.

The analysis helps the audience to view and understand the pattern and trends of the rides made throughout the time frame through the visualizations like the bar charts, heat maps and map views.

Importing library

```
#install.packages("ggplot2")
#install.packages("ggthemes")
#install.packages("lubridate")
#install.packages("dplyr")
#install.packages("tidyr")
#install.packages("DT")
#install.packages("scales")
library(ggplot2) #data visualization
library(ggthemes)#add-on to our main ggplot2
library(lubridate)# time categories
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)#data manipulation
```

```
##
## 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)#tidy data
library(DT)#datatable
library(scales)#graphical scales
```

Read data

```
apr_data <- read.csv("C:/Users/ADMIN/Desktop/Uber_Dataset_EDA-master/uber-raw-data-apr14.csv"
)
may_data <- read.csv("C:/Users/ADMIN/Desktop/Uber_Dataset_EDA-master/uber-raw-data-may14.csv"
)
jun_data <- read.csv("C:/Users/ADMIN/Desktop/Uber_Dataset_EDA-master/uber-raw-data-jun14.csv"
)
jul_data <- read.csv("C:/Users/ADMIN/Desktop/Uber_Dataset_EDA-master/uber-raw-data-jul14.csv"
)
aug_data <- read.csv("C:/Users/ADMIN/Desktop/Uber_Dataset_EDA-master/uber-raw-data-aug14.csv"
)
sep_data <- read.csv("C:/Users/ADMIN/Desktop/Uber_Dataset_EDA-master/uber-raw-data-sep14.csv"
)
```

Combining all the data

```
data_2014 <- rbind(apr_data,may_data,jun_data,jul_data,aug_data,sep_data)
```

Print head of data

```
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
```

```
length(data_2014$Date.Time) #number of records
```

```
## [1] 4534327
```

As date-time

```
data_2014$Date.Time <- as.POSIXct(data_2014$Date.Time, format = "%m/%d/%Y %H:%M:%S")
```

Time

```
data_2014$Time <- format(as.POSIXct(data_2014$Date.Time, format = "%m/%d/%Y %H:%M:%S"), forma
t="%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)))
```

Print head of data

```
head(data_2014)
```

```
##             Date.Time     Lat      Lon   Base     Time day month year dayofweek
## 1 2014-04-01 00:11:00 40.7690 -73.9549 B02512 00:11:00   1   Apr 2014       Tue
## 2 2014-04-01 00:17:00 40.7267 -74.0345 B02512 00:17:00   1   Apr 2014       Tue
## 3 2014-04-01 00:21:00 40.7316 -73.9873 B02512 00:21:00   1   Apr 2014       Tue
## 4 2014-04-01 00:28:00 40.7588 -73.9776 B02512 00:28:00   1   Apr 2014       Tue
## 5 2014-04-01 00:33:00 40.7594 -73.9722 B02512 00:33:00   1   Apr 2014       Tue
## 6 2014-04-01 00:33:00 40.7383 -74.0403 B02512 00:33:00   1   Apr 2014       Tue
##   hour minute second
## 1    0     11      0
## 2    0     17      0
## 3    0     21      0
## 4    0     28      0
## 5    0     33      0
## 6    0     33      0
```

Plotting the trip by the hours in a day

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

Tabuler view

```
datatable(hour_data)
```

Show 10 **entries** **Search:**

	hour	Total
1	0	103836
2	1	67227

	hour	Total
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

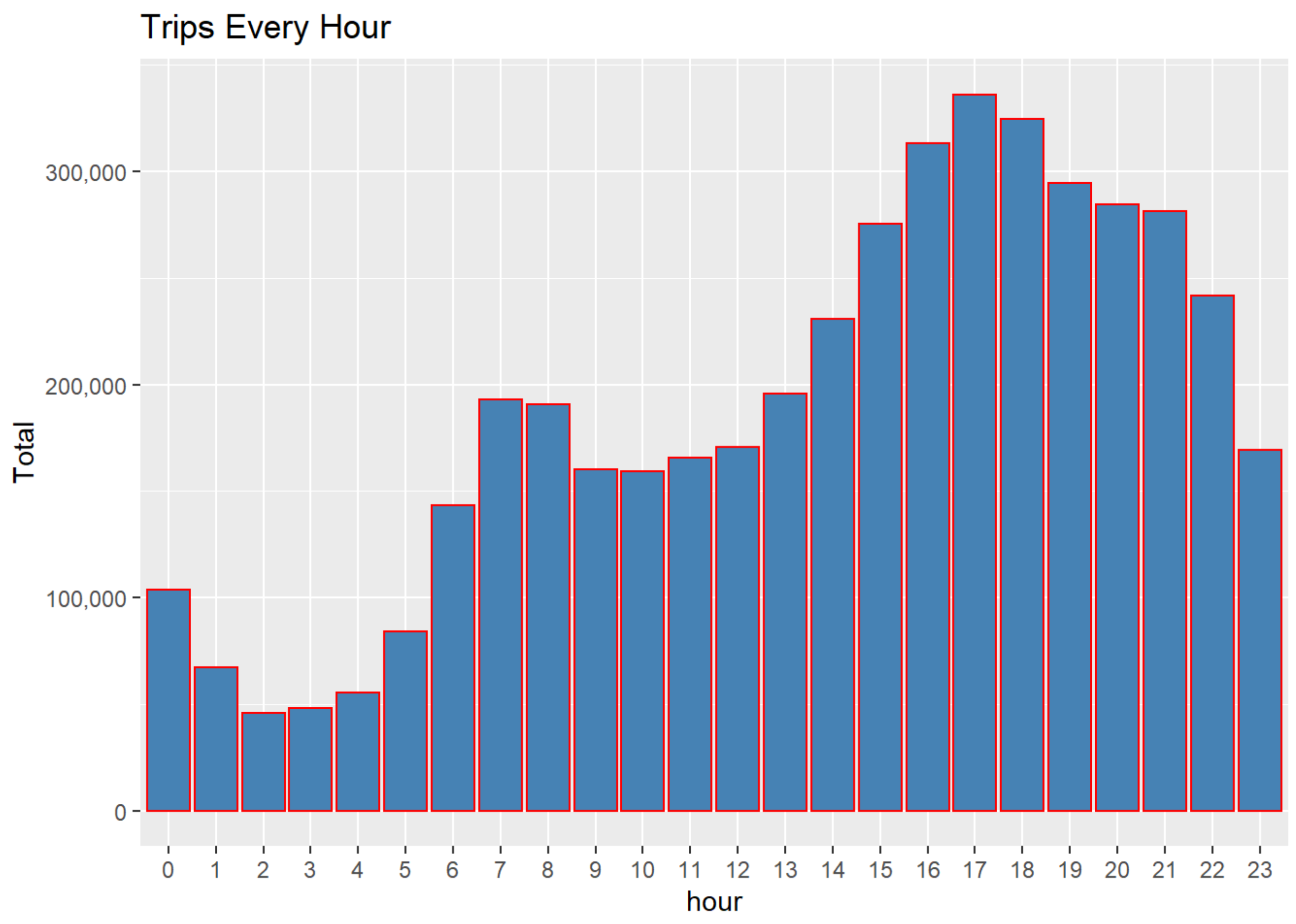
Normal view

```
head(hour_data)
```

```
## # A tibble: 6 x 2
##   hour   Total
##   <fct>  <int>
## 1 0     103836
## 2 1      67227
## 3 2      45865
## 4 3      48287
## 5 4      55230
## 6 5      83939
```

#visualization

```
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)
```



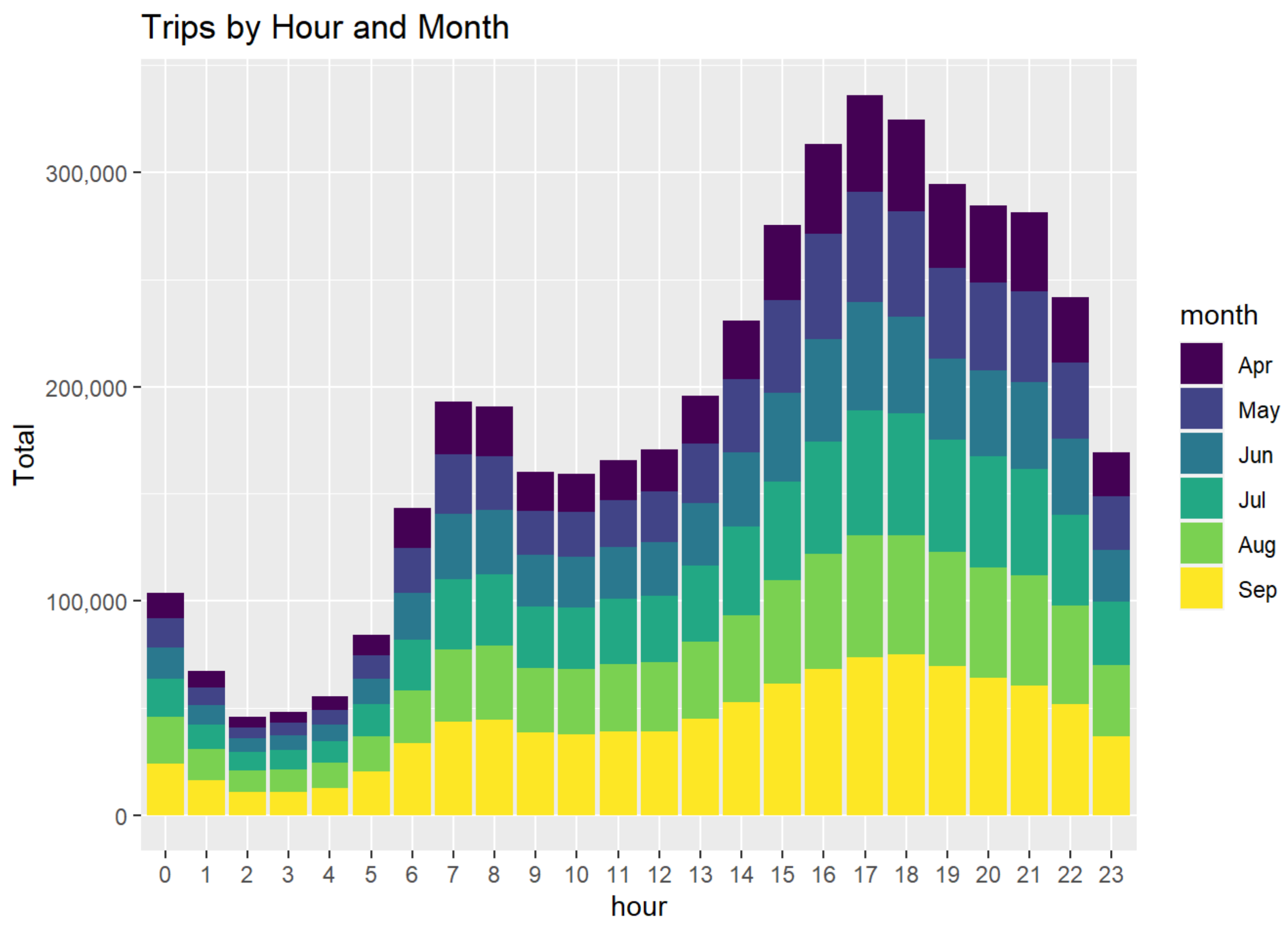
#

As you see in result the peak starts from 16 (4 pm) and continoue upto 21(9 pm)

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

```
## `summarise()` has grouped output by 'month'. You can override using the `.groups` argumen
t.
```

```
ggplot(month_hour, aes(hour, Total, fill = month)) +
  geom_bar( stat = "identity") +
  ggtitle("Trips by Hour and Month") +
  scale_y_continuous(labels = comma)
```

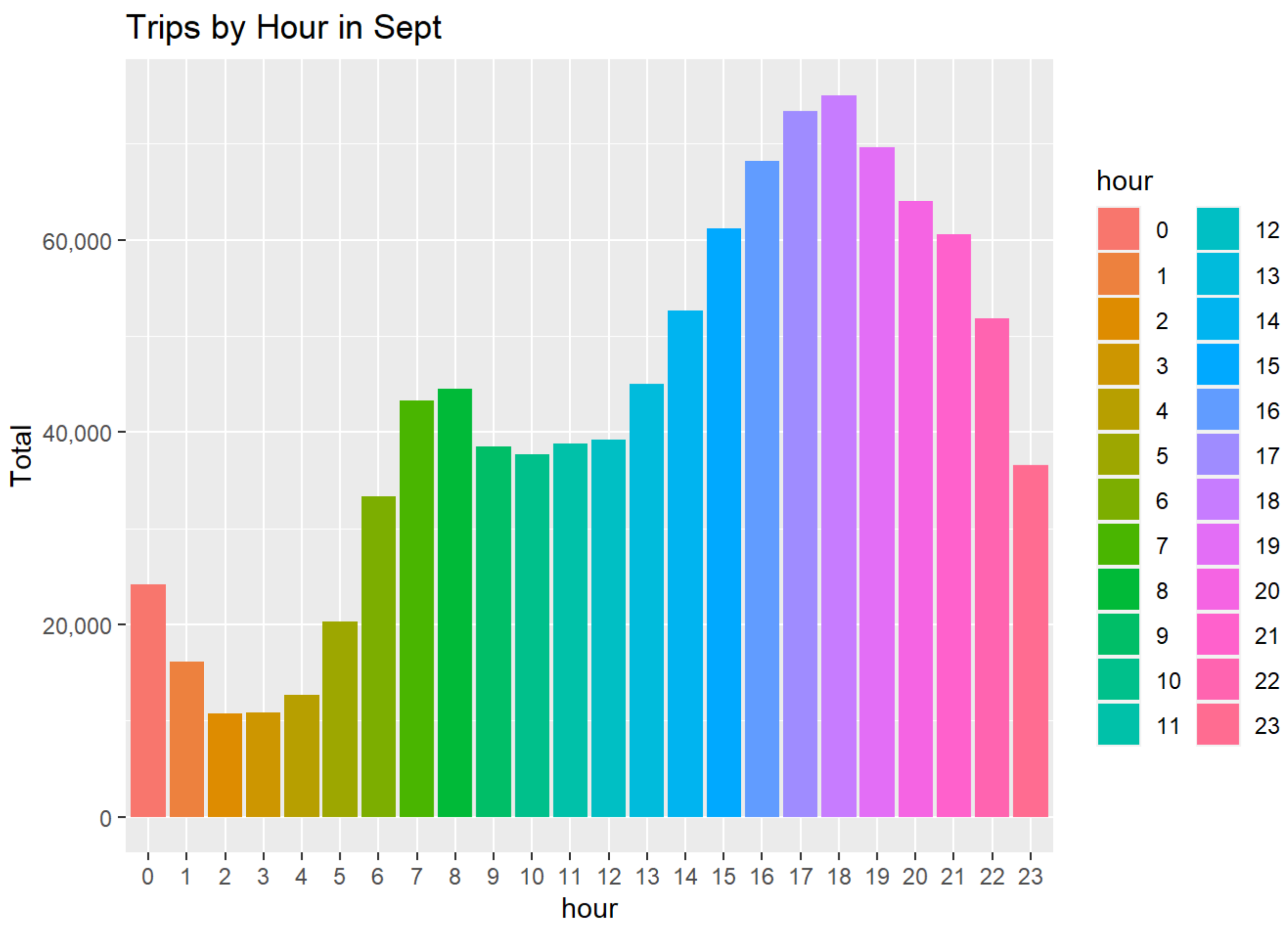


Let’s see september months data

```
sept_hour <- data_2014 %>%
  group_by(month, hour) %>%
  filter(month=="Sep") %>%
  summarise(Total =n())
```

```
## `summarise()` has grouped output by 'month'. You can override using the `.groups` argumen
t.
```

```
ggplot(sept_hour, aes(hour, Total, fill = hour)) +
  geom_bar( stat = "identity") +
  ggtitle("Trips by Hour in Sept") +
  scale_y_continuous(labels = comma)
```

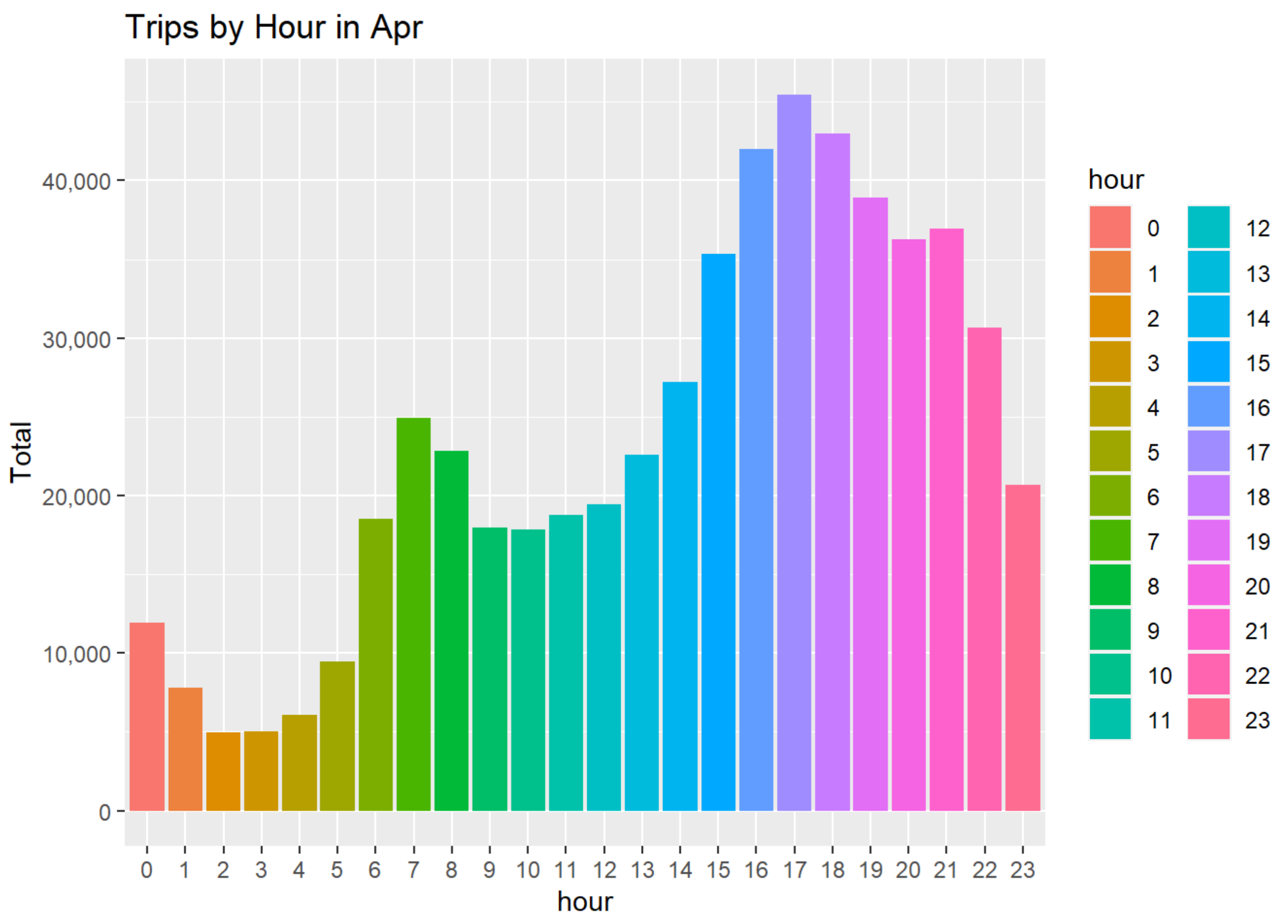


Let’s see April data

```
Apr_hour <- data_2014 %>%
  group_by(month, hour) %>%
  filter(month=="Apr") %>%
  summarise(Total =n())
```

```
## `summarise()` has grouped output by 'month'. You can override using the `.groups` argumen
t.
```

```
ggplot(Apr_hour, aes(hour, Total, fill = hour)) +
  geom_bar( stat = "identity") +
  ggtitle("Trips by Hour in Apr") +
  scale_y_continuous(labels = comma)
```



Plotting data by trips during every day of the month

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

Show 10 entries

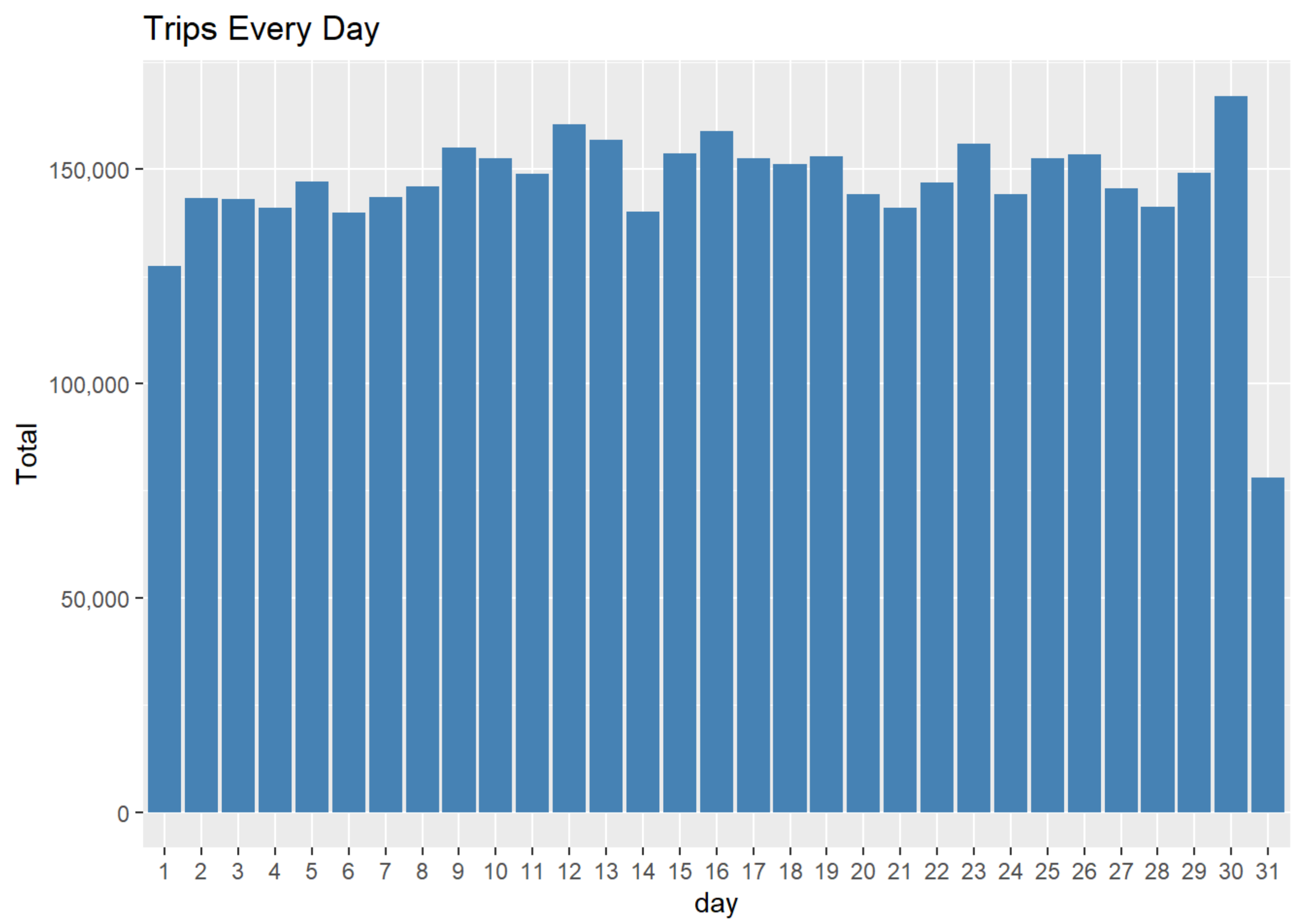
Search:

	day	Total
1	1	127430
2	2	143201
3	3	142983
4	4	140923
5	5	147054
6	6	139886
7	7	143503

	day	Total
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") +
  ggtitle("Trips Every Day") +
  theme(legend.position = "none") +
  scale_y_continuous(labels = comma)
```

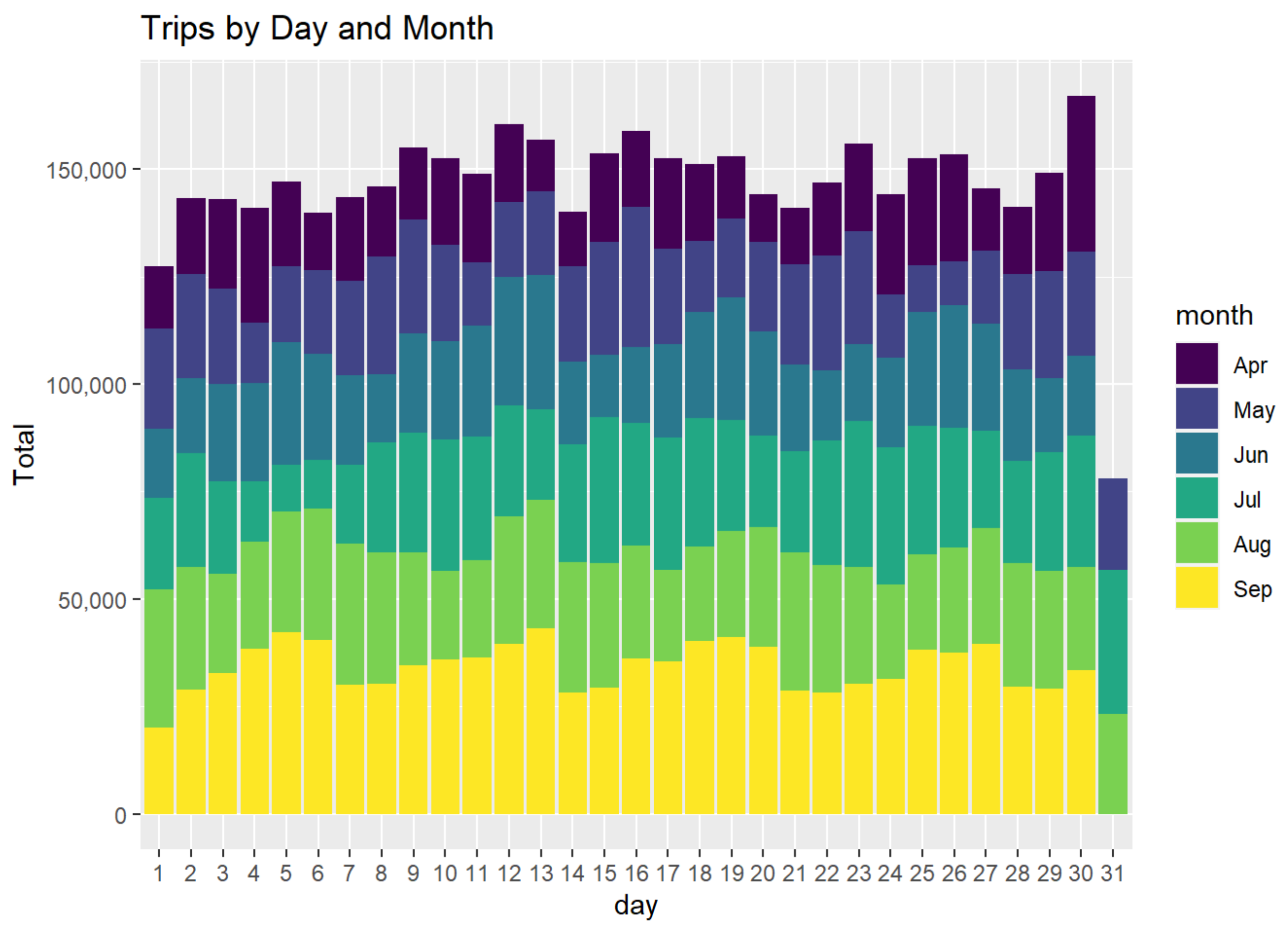


Last day of month has less value it may be because not all the month has 31 days

```
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` argumen
t.
```

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



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

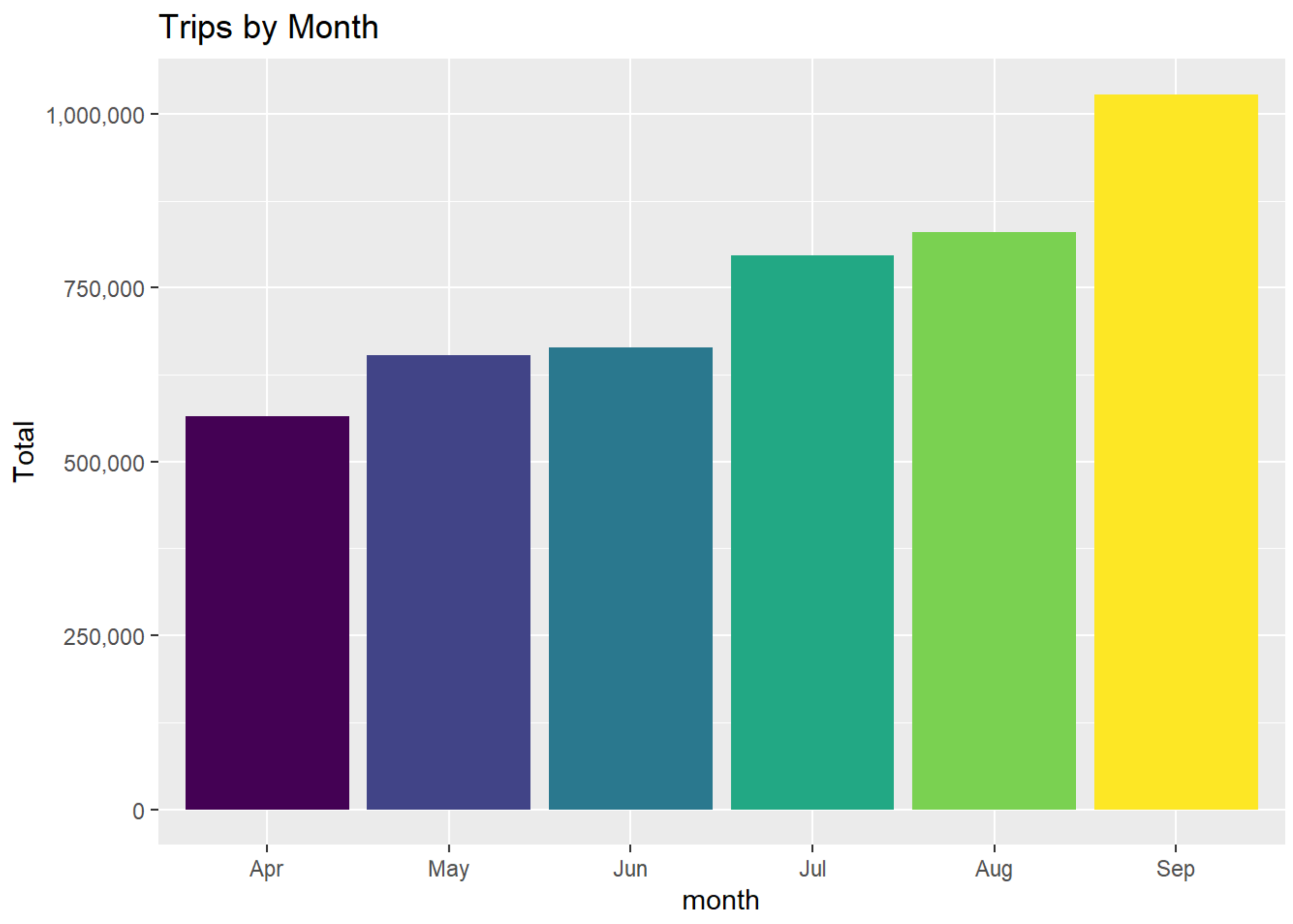
Show 10 entries Search:

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

	month	Total
6	Sep	1028136

Showing 1 to 6 of 6 entries Previous 1 Next

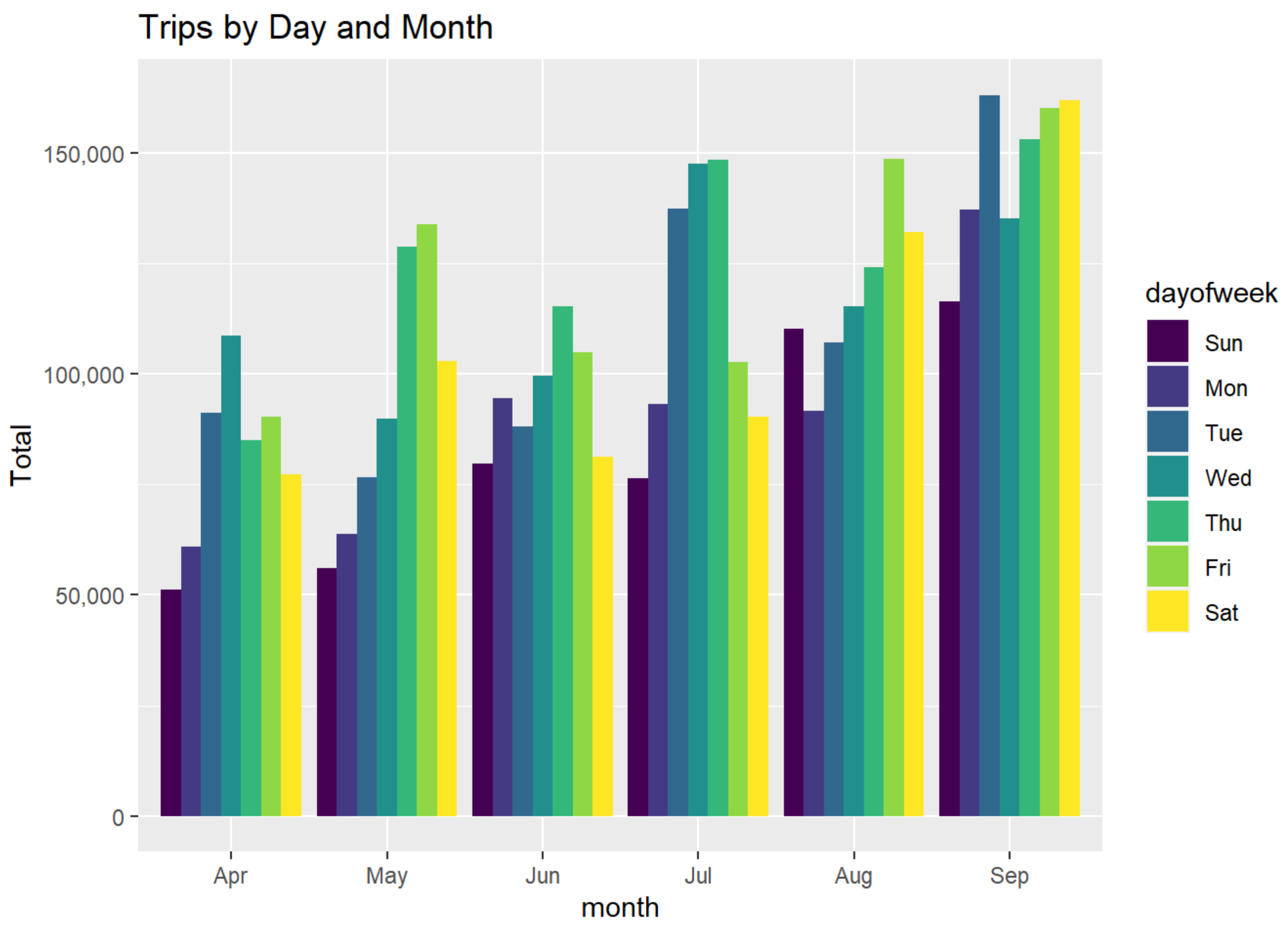
```
ggplot(month_group , aes(month, Total, fill = month)) +
  geom_bar( stat = "identity") +
  ggtitle("Trips by Month") +
  theme(legend.position = "none") +
  scale_y_continuous(labels = comma)
```



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

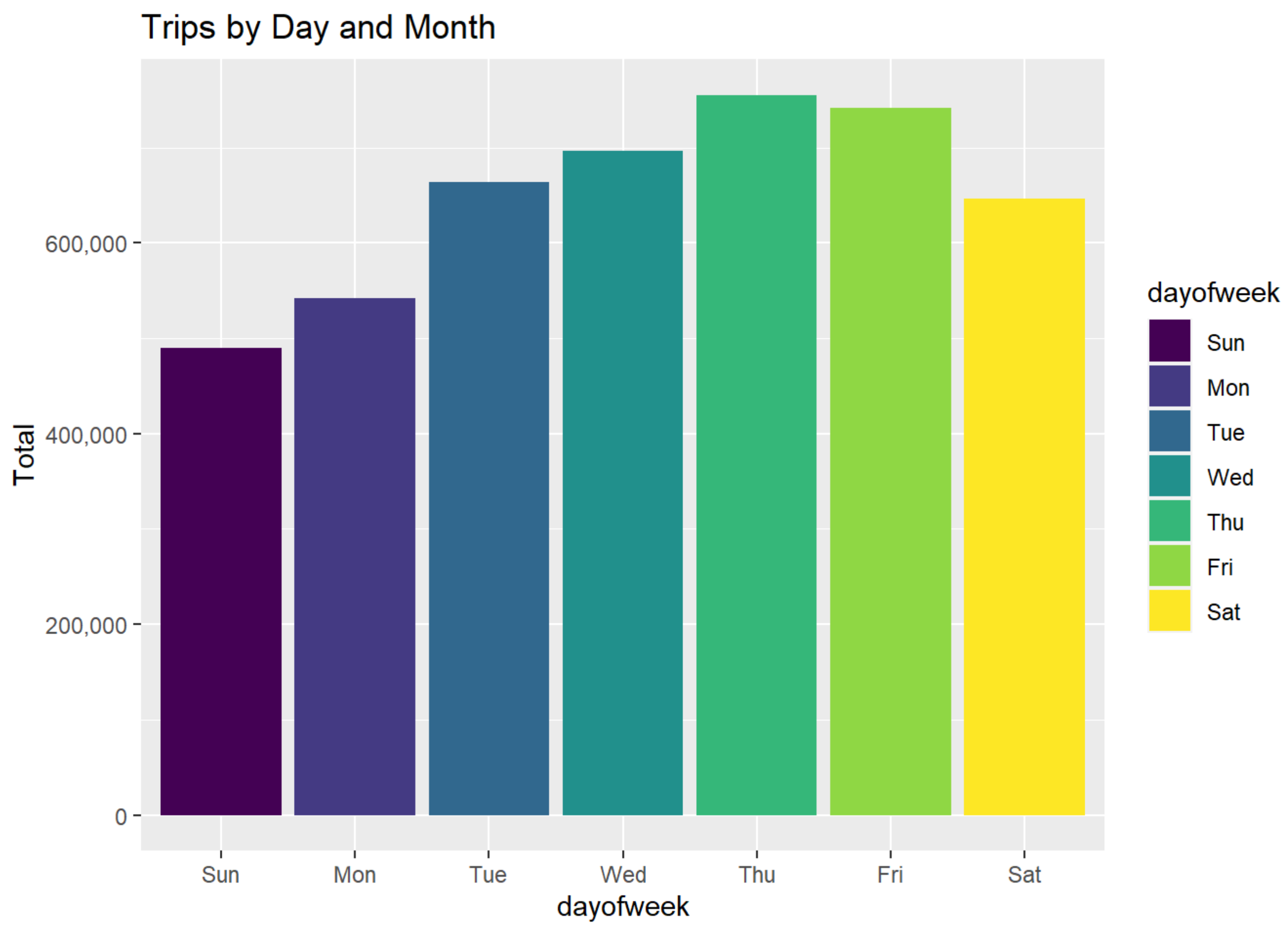
```
## `summarise()` has grouped output by 'month'. You can override using the `.groups` argumen
t.
```

```
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)# +
```



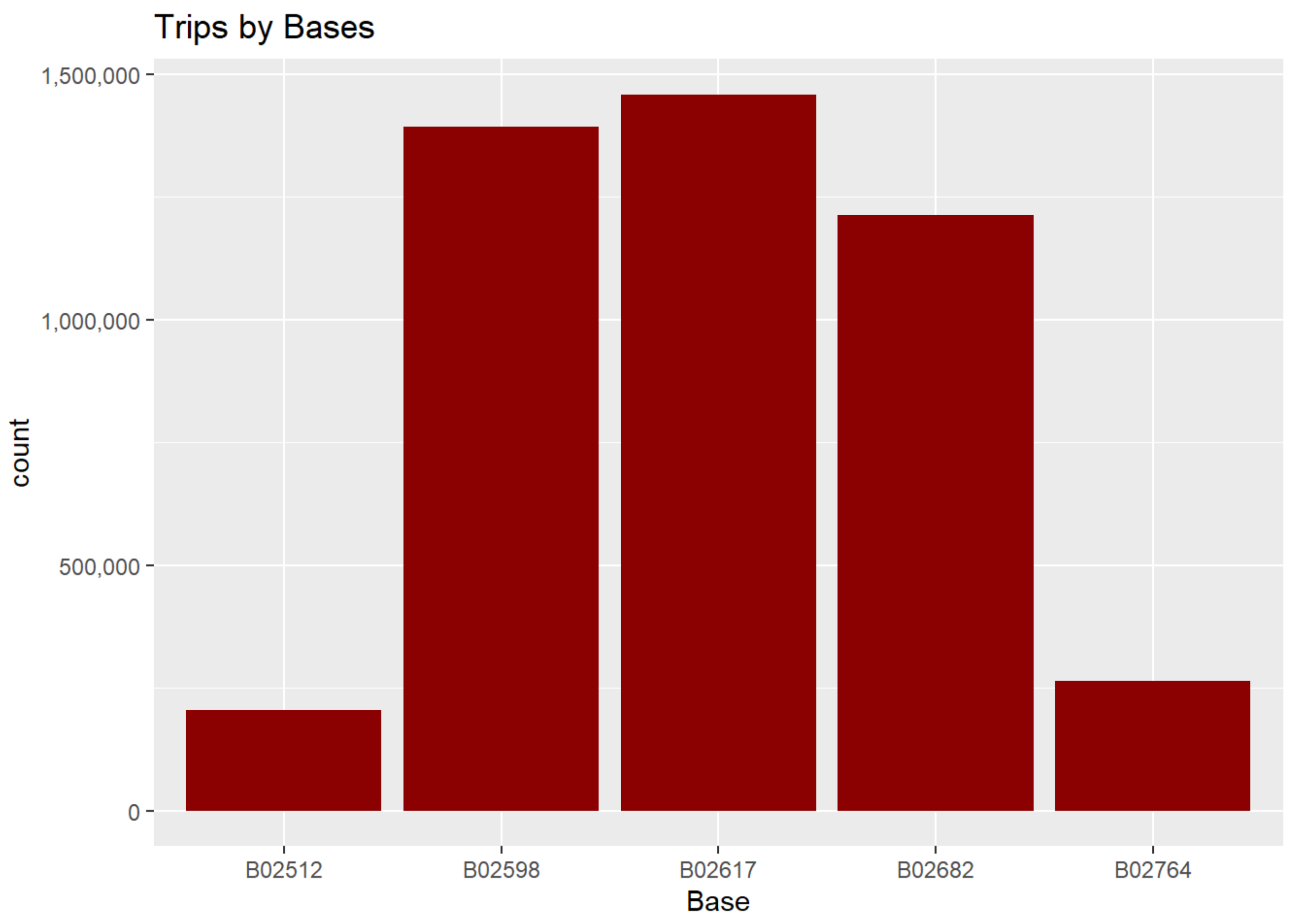
```
#scale_fill_manual
```

```
weekday <- data_2014 %>%
  group_by(dayofweek) %>%
  dplyr::summarize(Total = n())
ggplot(weekday, aes(dayofweek,Total, fill = dayofweek)) +
  geom_bar( stat = "identity", position = "dodge") +
  ggtitle("Trips by Day and Month") +
  scale_y_continuous(labels = comma)# +
```

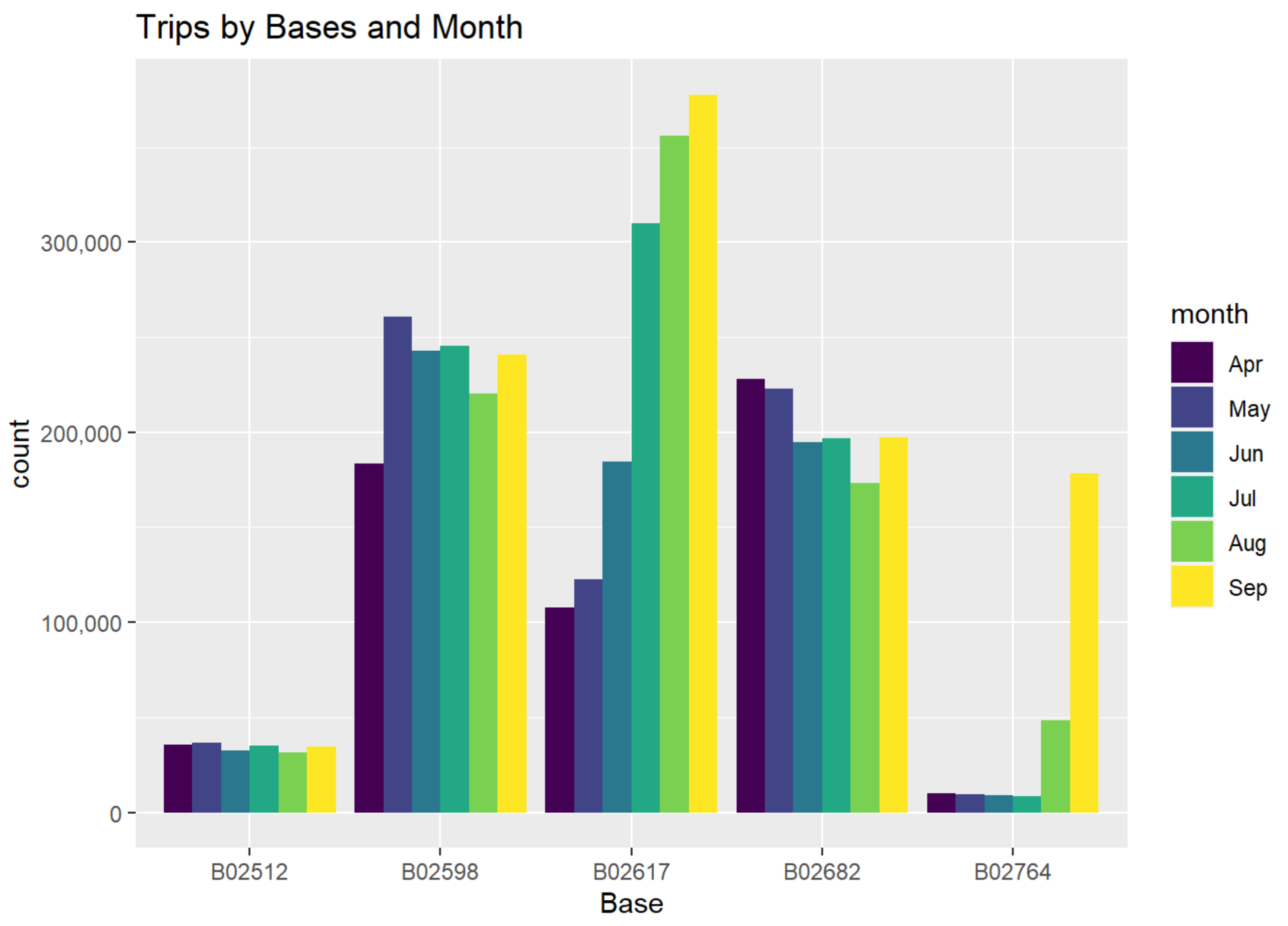


```
#scale_fill_manual
```

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

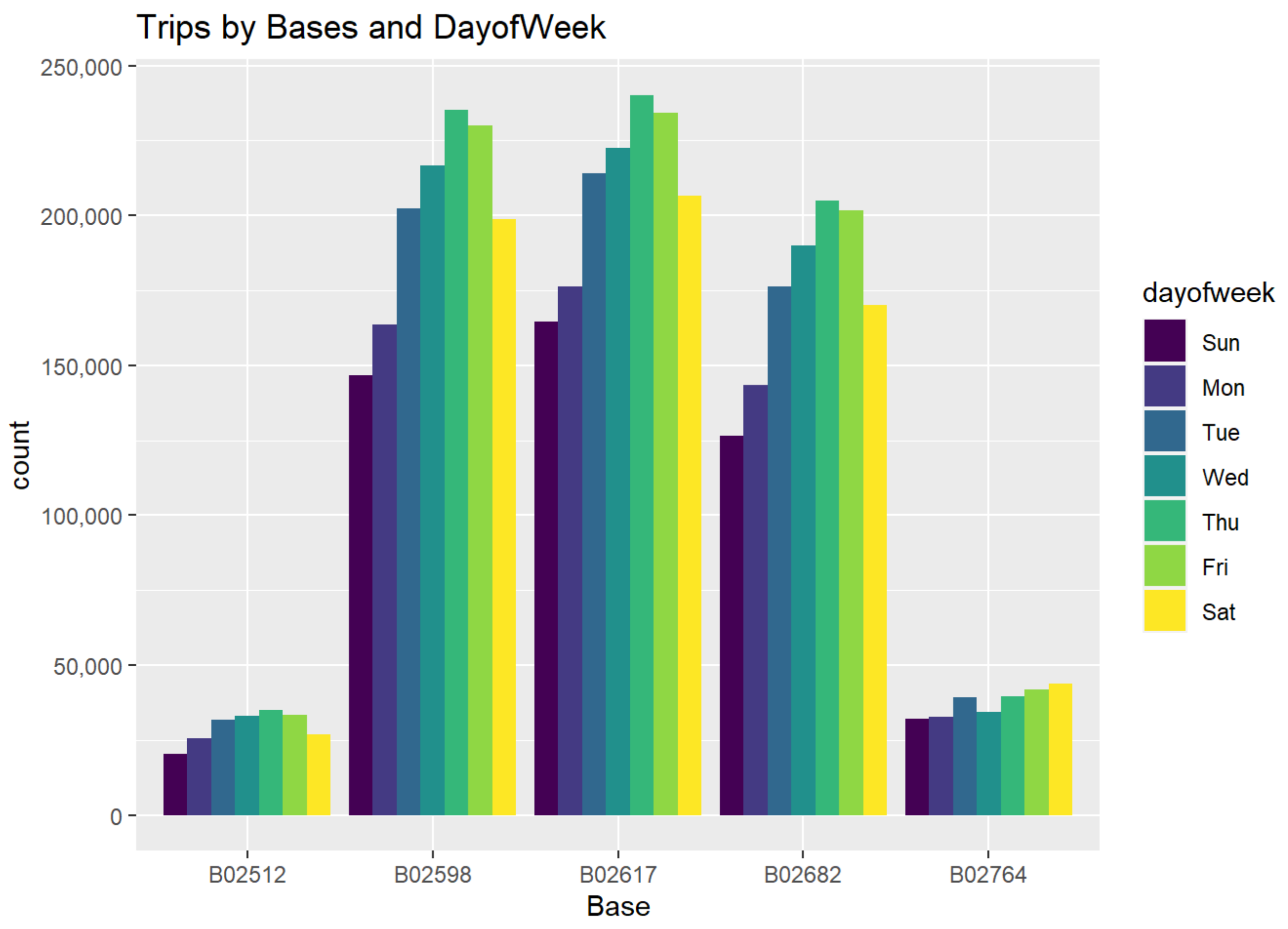


```
ggplot(data_2014, aes(Base, fill = month)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = comma) +
  ggtitle("Trips by Bases and Month")#+
```



```
  #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)
```

```
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)
```

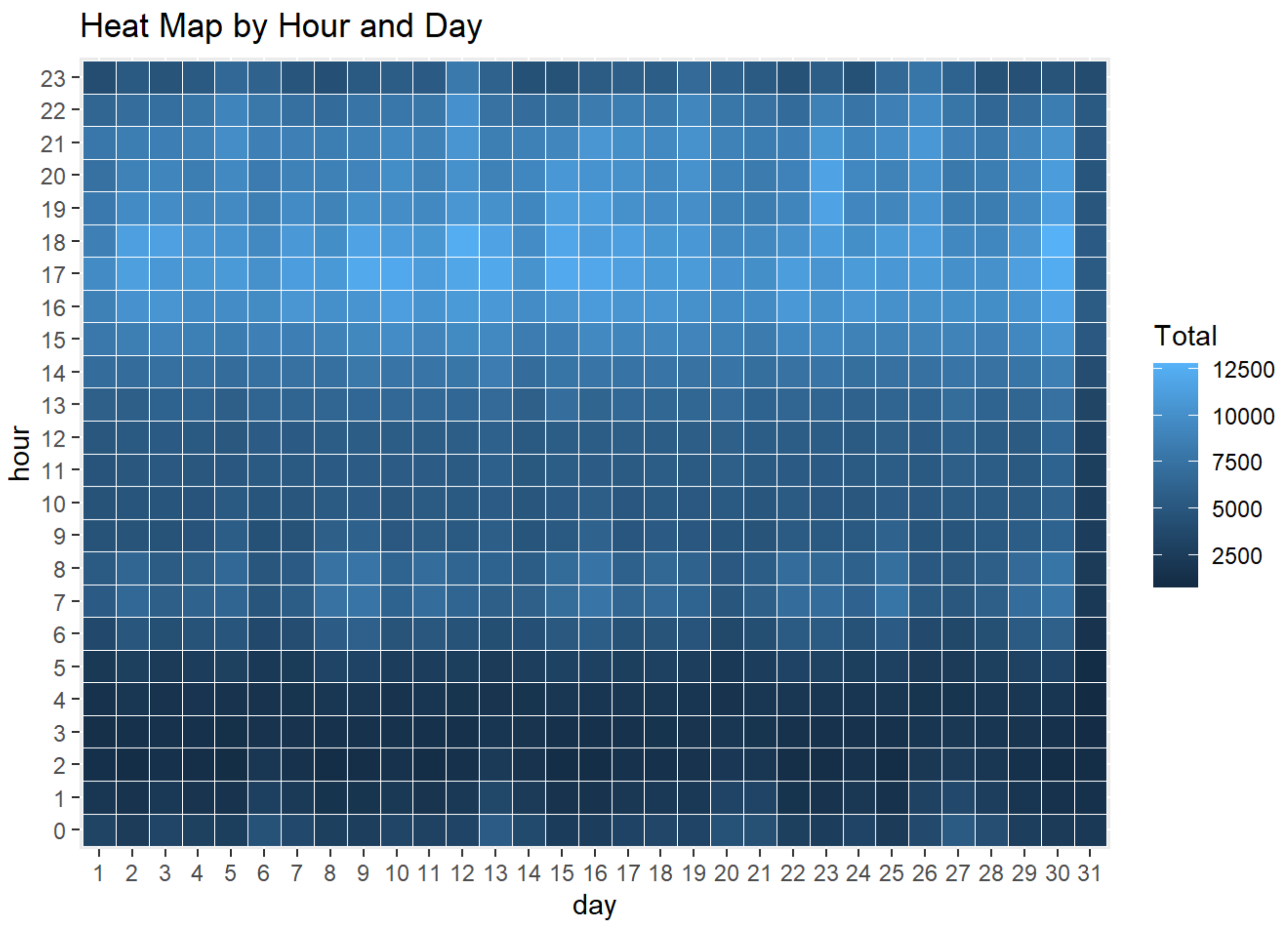
Show 10 **entries** **Search:**

	day	hour	Total
1	1	0	3247
2	1	1	1982
3	1	2	1284
4	1	3	1331
5	1	4	1458
6	1	5	2171

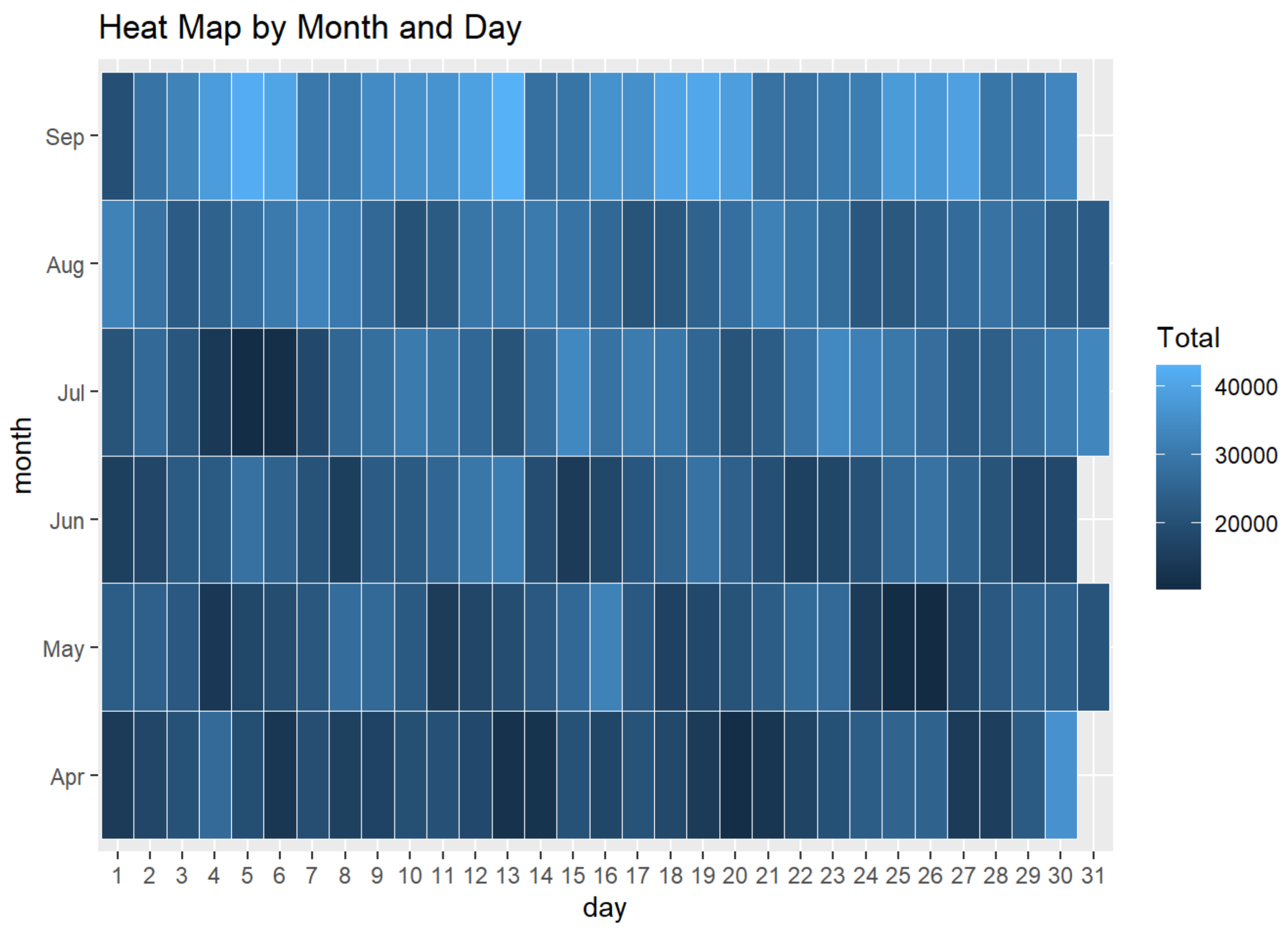
	day	hour	Total
7	1	6	3717
8	1	7	5470
9	1	8	5376
10	1	9	4688

Showing 1 to 10 of 744 entries Previous 1 2 3 4 5 … 75 Next

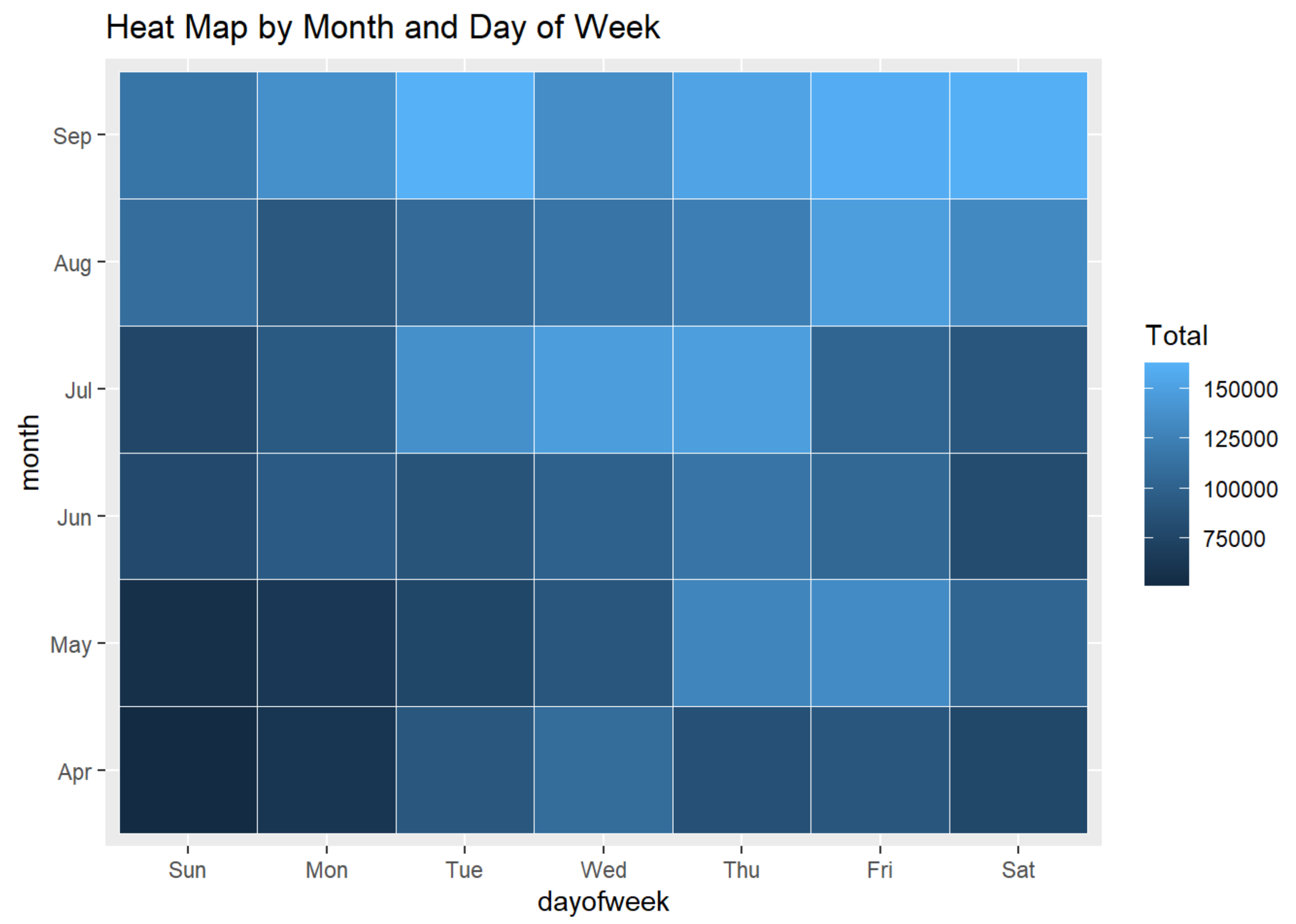
```
ggplot(day_and_hour, aes(day, hour, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Hour and Day")
```



```
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")
```



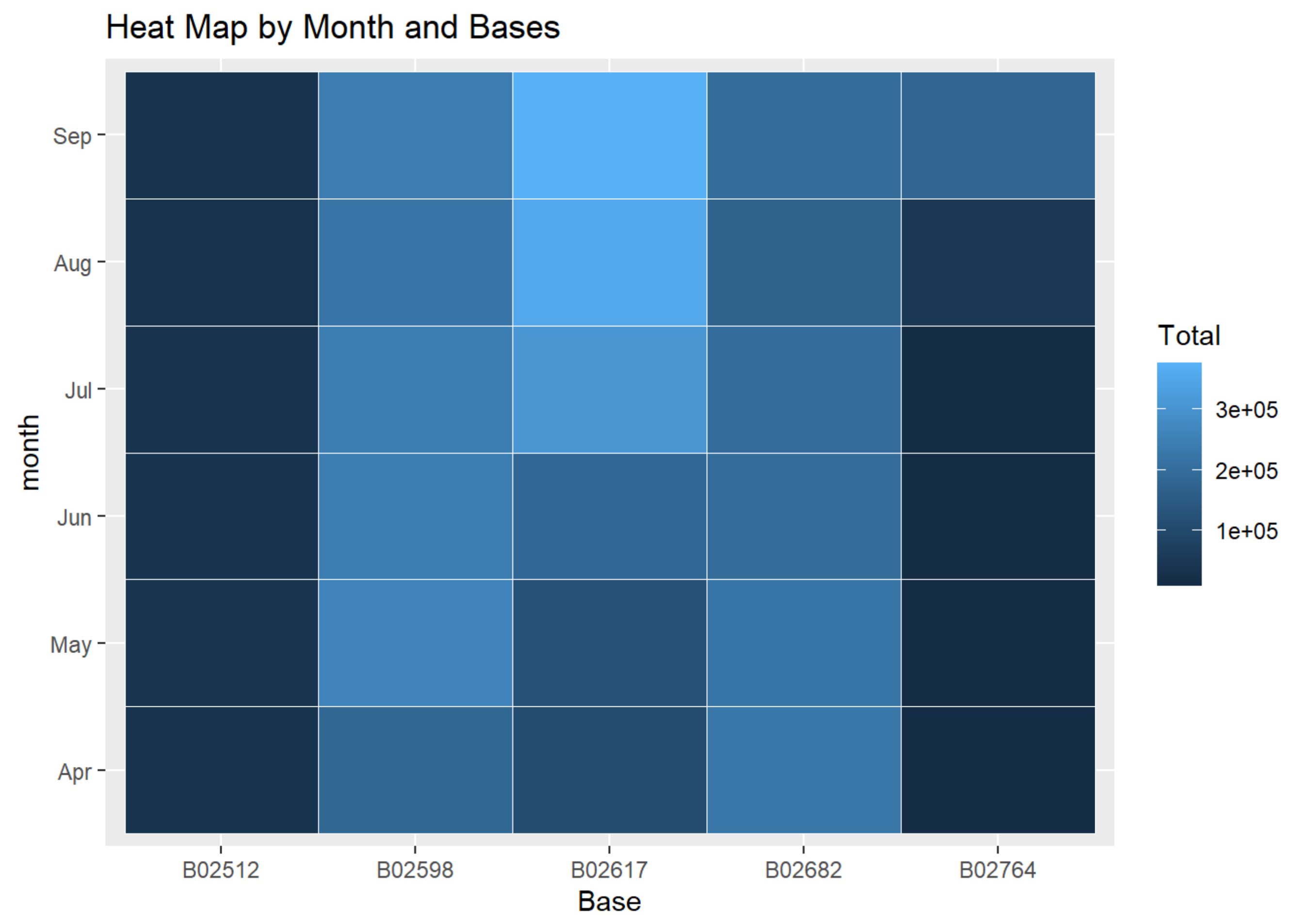
```
month_base <-  data_2014 %>%
  group_by(Base, month) %>%
  dplyr::summarize(Total = n())
```

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

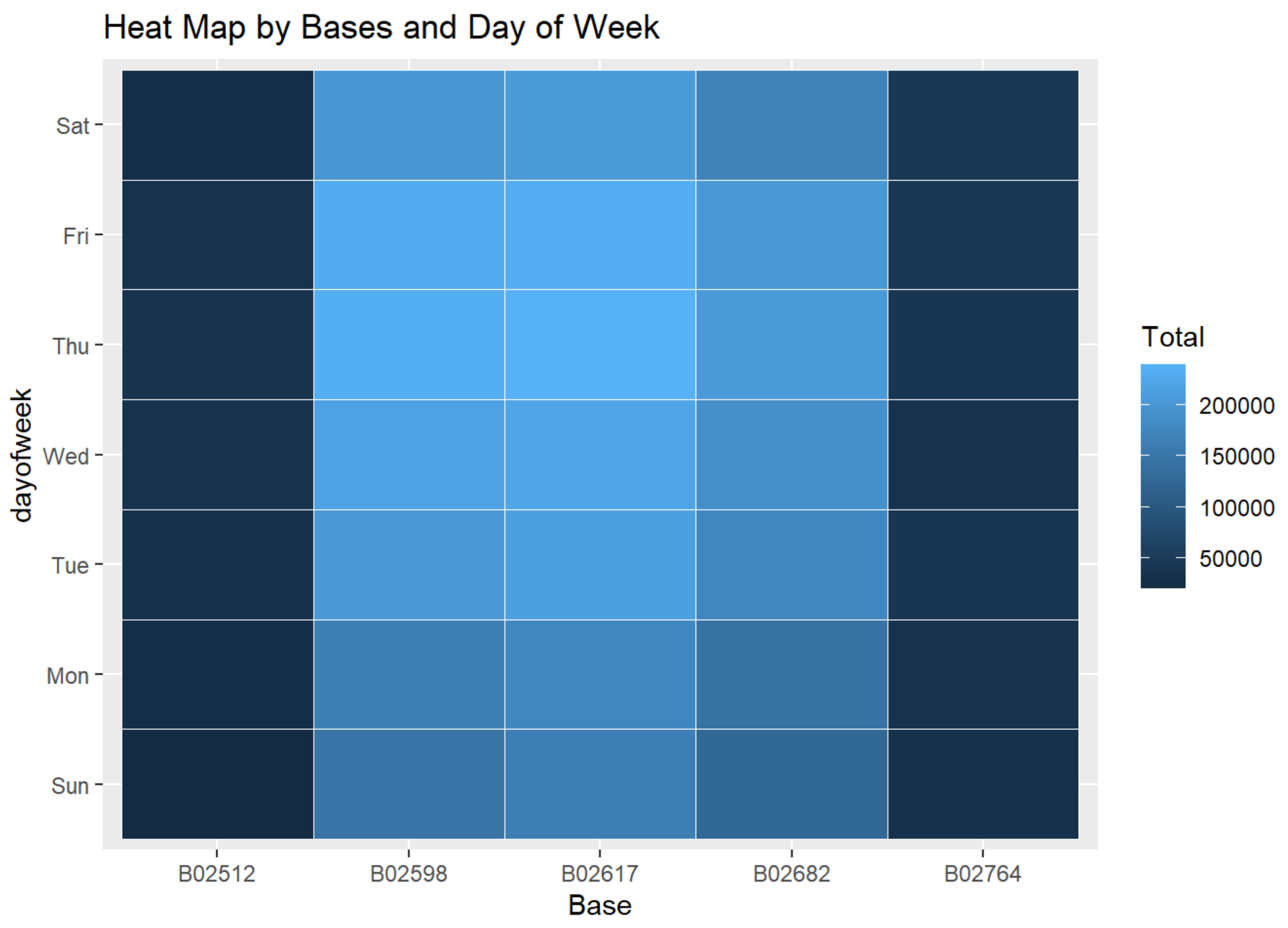
```
day0fweek_bases <-  data_2014 %>%
  group_by(Base, dayofweek) %>%
  dplyr::summarize(Total = n())
```

```
## `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")
```



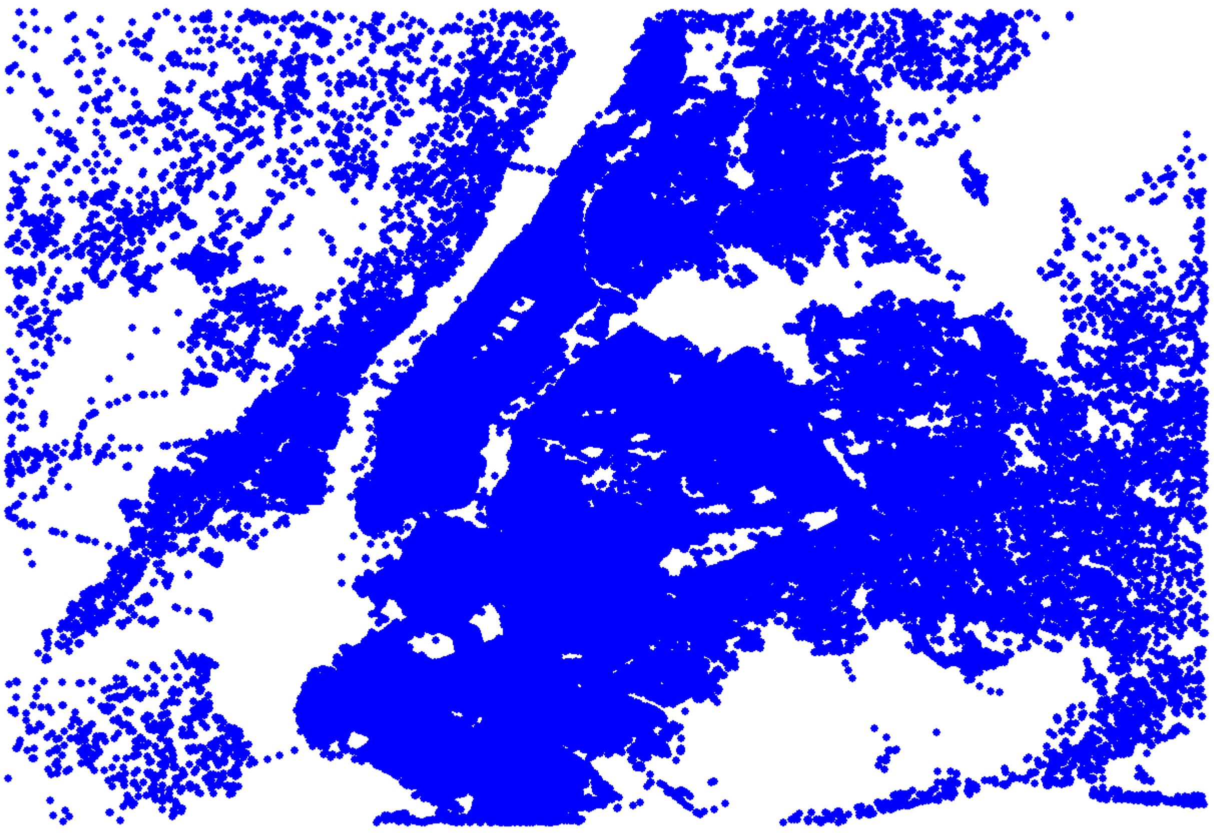
```
ggplot(day0fweek_bases, aes(Base, dayofweek, fill = Total)) +
  geom_tile(color = "white") +
  ggtitle("Heat Map by Bases and Day of Week")
```



```
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).
```

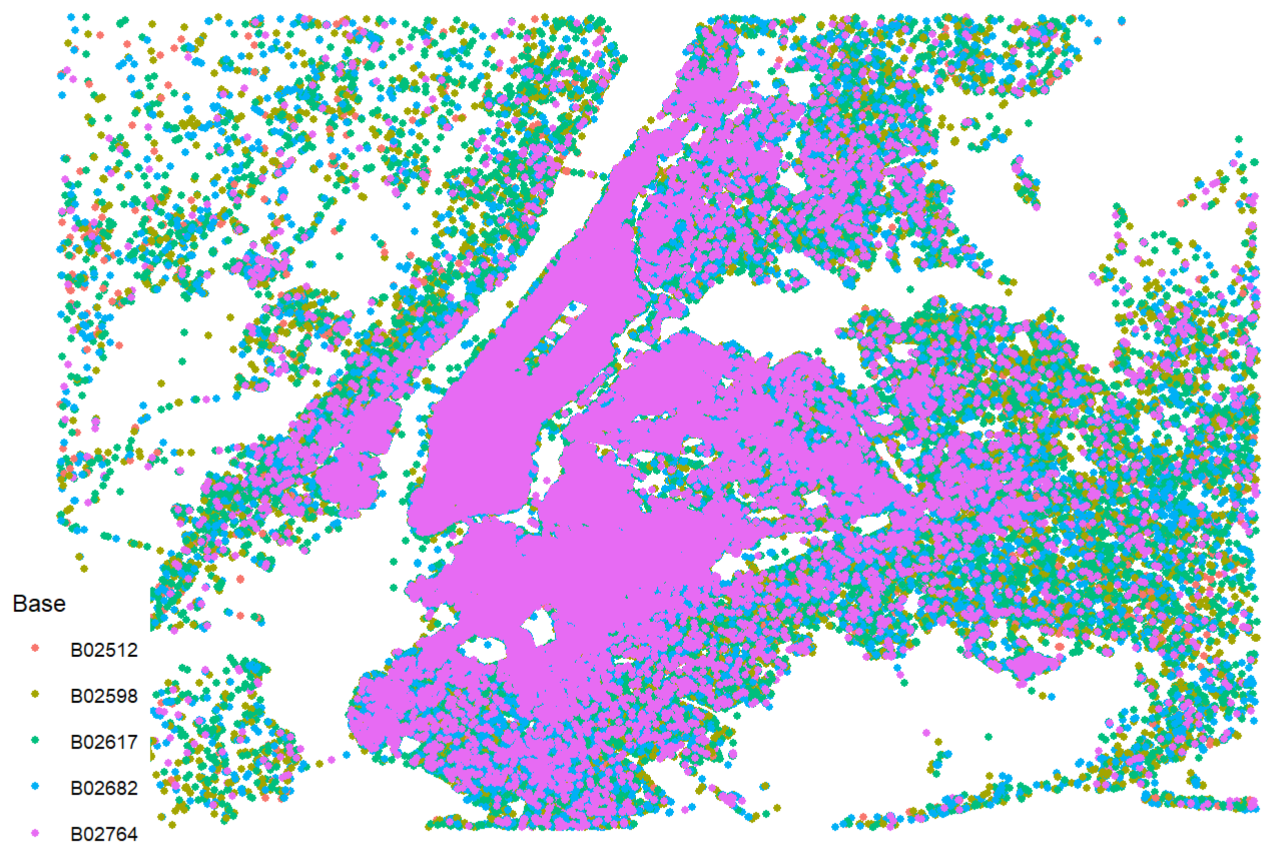
NYC MAP BASED ON UBER RIDES DURING 2014 (APR-SEP)



```
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")
```

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

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


Base
B02512
B02598
B02617
B02682
B02764