Google Data Analytics

Meet

2023-04-02

Importing Library

```r
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.1     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the  ]8;;http://conflicted.r-lib.org/ conflicted package ]8;;  to force all conflicts to become errors
```

```r
library(ggplot2)
library(lubridate)
library(chron) # For time
```

```
##
## Attaching package: 'chron'
##
## The following objects are masked from 'package:lubridate':
##
##     days, hours, minutes, seconds, years
```

Importing Data

```r
march_22 <- read.csv("Biker_03_2022.csv")
april_22 <- read.csv("Biker_04_2022.csv")
may_22 <- read.csv("Biker_05_2022.csv")
june_22 <- read.csv("Biker_06_2022.csv")
july_22 <- read.csv("Biker_07_2022.csv")
august_22 <- read.csv("Biker_08_2022.csv")
september_22 <- read.csv("Biker_09_2022.csv")
october_22 <- read.csv("Biker_10_2022.csv")
november_22 <- read.csv("Biker_11_2022.csv")
december_22 <- read.csv("Biker_12_2022.csv")
january_23 <- read.csv("Biker_01_2023.csv")
february_23 <- read.csv("Biker_02_2023.csv")
```

Combine to one data set named as *alldata*.

```
alldata <- rbind(march_22, april_22, may_22, june_22, july_22, august_22, september_22, october_2
2, november_22, december_22, january_23, february_23)
```

Get a overview of the dataset.

```
head(alldata)
```

```
##            ride_id rideable_type      started_at        ended_at
## 1 47EC0A7F82E65D52  classic_bike 3/21/2022 13:45 3/21/2022 13:51
## 2 8494861979B0F477 electric_bike  3/16/2022 9:37  3/16/2022 9:43
## 3 EFE527AF80B66109  classic_bike 3/23/2022 19:52 3/23/2022 19:54
## 4 9F446FD9DEE3F389  classic_bike  3/1/2022 19:12  3/1/2022 19:22
## 5 431128AD9AFFEDC0  classic_bike 3/21/2022 18:37 3/21/2022 19:19
## 6 9AA8A13AF7A85325  classic_bike  3/7/2022 17:10  3/7/2022 17:15
##                   start_station_name start_station_id
## 1             Wabash Ave & Wacker Pl     TA1307000131
## 2              Michigan Ave & Oak St            13042
## 3              Broadway & Berwyn Ave            13109
## 4             Wabash Ave & Wacker Pl     TA1307000131
## 5 DuSable Lake Shore Dr & North Blvd           LF-005
## 6          Bissell St & Armitage Ave            13059
##                       end_station_name end_station_id start_lat start_lng
## 1             Kingsbury St & Kinzie St   KA1503000043  41.88688 -87.62603
## 2 Orleans St & Chestnut St (NEXT Apts)            620  41.90100 -87.62375
## 3                 Broadway & Ridge Ave          15578  41.97835 -87.65975
## 4           Franklin St & Jackson Blvd   TA1305000025  41.88688 -87.62603
## 5             Loomis St & Jackson Blvd          13206  41.91172 -87.62680
## 6         Southport Ave & Clybourn Ave   TA1309000030  41.91802 -87.65218
##    end_lat   end_lng member_casual ride_time day_of_week
## 1 41.88918 -87.63851        member   0:06:17           2
## 2 41.89820 -87.63754        member   0:06:18           4
## 3 41.98404 -87.66027        member   0:02:46           4
## 4 41.87771 -87.63532        member   0:09:48           3
## 5 41.87794 -87.66201        member   0:42:10           2
## 6 41.92077 -87.66371        member   0:04:42           2
```

```
glimpse(alldata)
```

```
## Rows: 5,829,084
## Columns: 15
## $ ride_id            <chr> "47EC0A7F82E65D52", "8494861979B0F477", "EFE527AF80…
## $ rideable_type      <chr> "classic_bike", "electric_bike", "classic_bike", "c…
## $ started_at         <chr> "3/21/2022 13:45", "3/16/2022 9:37", "3/23/2022 19:…
## $ ended_at           <chr> "3/21/2022 13:51", "3/16/2022 9:43", "3/23/2022 19:…
## $ start_station_name <chr> "Wabash Ave & Wacker Pl", "Michigan Ave & Oak St", …
## $ start_station_id   <chr> "TA1307000131", "13042", "13109", "TA1307000131", "…
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Orleans St & Chestnut …
## $ end_station_id     <chr> "KA1503000043", "620", "15578", "TA1305000025", "13…
## $ start_lat          <dbl> 41.88688, 41.90100, 41.97835, 41.88688, 41.91172, 4…
## $ start_lng          <dbl> -87.62603, -87.62375, -87.65975, -87.62603, -87.626…
## $ end_lat            <dbl> 41.88918, 41.89820, 41.98404, 41.87771, 41.87794, 4…
## $ end_lng            <dbl> -87.63851, -87.63754, -87.66027, -87.63532, -87.662…
## $ member_casual      <chr> "member", "member", "member", "member", "member", "…
## $ ride_time          <chr> "0:06:17", "0:06:18", "0:02:46", "0:09:48", "0:42:1…
## $ day_of_week        <int> 2, 4, 4, 3, 2, 2, 5, 7, 5, 6, 1, 4, 2, 2, 4, 4, 4, …
```

```
summary(alldata)
```

```
##    ride_id          rideable_type       started_at          ended_at        
##  Length:5829084     Length:5829084     Length:5829084     Length:5829084    
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##  start_station_name start_station_id   end_station_name   end_station_id    
##  Length:5829084     Length:5829084     Length:5829084     Length:5829084    
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    start_lat       start_lng         end_lat         end_lng      
##  Min.   :41.64   Min.   :-87.84   Min.   : 0.00   Min.   :-88.14  
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66  
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64  
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65  
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63  
##  Max.   :42.07   Max.   :-87.52   Max.   :42.37   Max.   :  0.00  
##                                   NA's   :5938    NA's   :5938    
##  member_casual       ride_time          day_of_week   
##  Length:5829084     Length:5829084     Min.   :1.000  
##  Class :character   Class :character   1st Qu.:2.000  
##  Mode  :character   Mode  :character   Median :4.000  
##                                        Mean   :4.092  
##                                        3rd Qu.:6.000  
##                                        Max.   :7.000  
## 
```

Cleaning data

To avoid the future complications, let’s first make the missing and spacing values if any into **NA**

```
alldata[alldata=="" | alldata==" "] <- NA
```

Check NA values

Let’s check how many missing values are there in the dataset column-wise.

ride_id

```
table(is.na(alldata$ride_id))
```

```
##
##   FALSE
## 5829084
```

rideable_type

```
table(is.na(alldata$rideable_type))
```

```
##
##   FALSE
## 5829084
```

started_at

```
table(is.na(alldata$started_at))
```

```
##
##   FALSE
## 5829084
```

ended_at

```
table(is.na(alldata$ended_at))
```

```
##
##   FALSE
## 5829084
```

start_station_name

```
table(is.na(alldata$start_station_name))
```

```
##
##   FALSE    TRUE
## 4978666  850418
```

Now, we know that there are some missing value *(850418)* in this column so if needed we can add…

start_station_id

```
table(is.na(alldata$start_station_id))
```

```
##
##   FALSE    TRUE
## 4978534  850550
```

This column have *850550* missing values

end_station_name

```
table(is.na(alldata$end_station_name))
```

```
##
##   FALSE    TRUE
## 4920046  909038
```

This column have *909038* missing values

end_station_id

```
table(is.na(alldata$end_station_id))
```

```
##
##   FALSE    TRUE
## 4919905  909179
```

This column have *909179* missing values

start_lat

```
table(is.na(alldata$start_lat))
```

```
##
##   FALSE
## 5829084
```

start_lng

```
table(is.na(alldata$start_lng))
```

```
##
##   FALSE
## 5829084
```

end_lat

```
table(is.na(alldata$end_lat))
```

```
##
##   FALSE    TRUE
## 5823146    5938
```

This column have *5938* missing values

end_lng

```
table(is.na(alldata$end_lng))
```

```
##
##   FALSE    TRUE
## 5823146    5938
```

This column have *5938* missing values

member_casual

```
table(is.na(alldata$member_casual))
```

```
##
##   FALSE
## 5829084
```

ride_time

```
table(is.na(alldata$ride_time))
```

```
##
##   FALSE
## 5829084
```

day_of_week

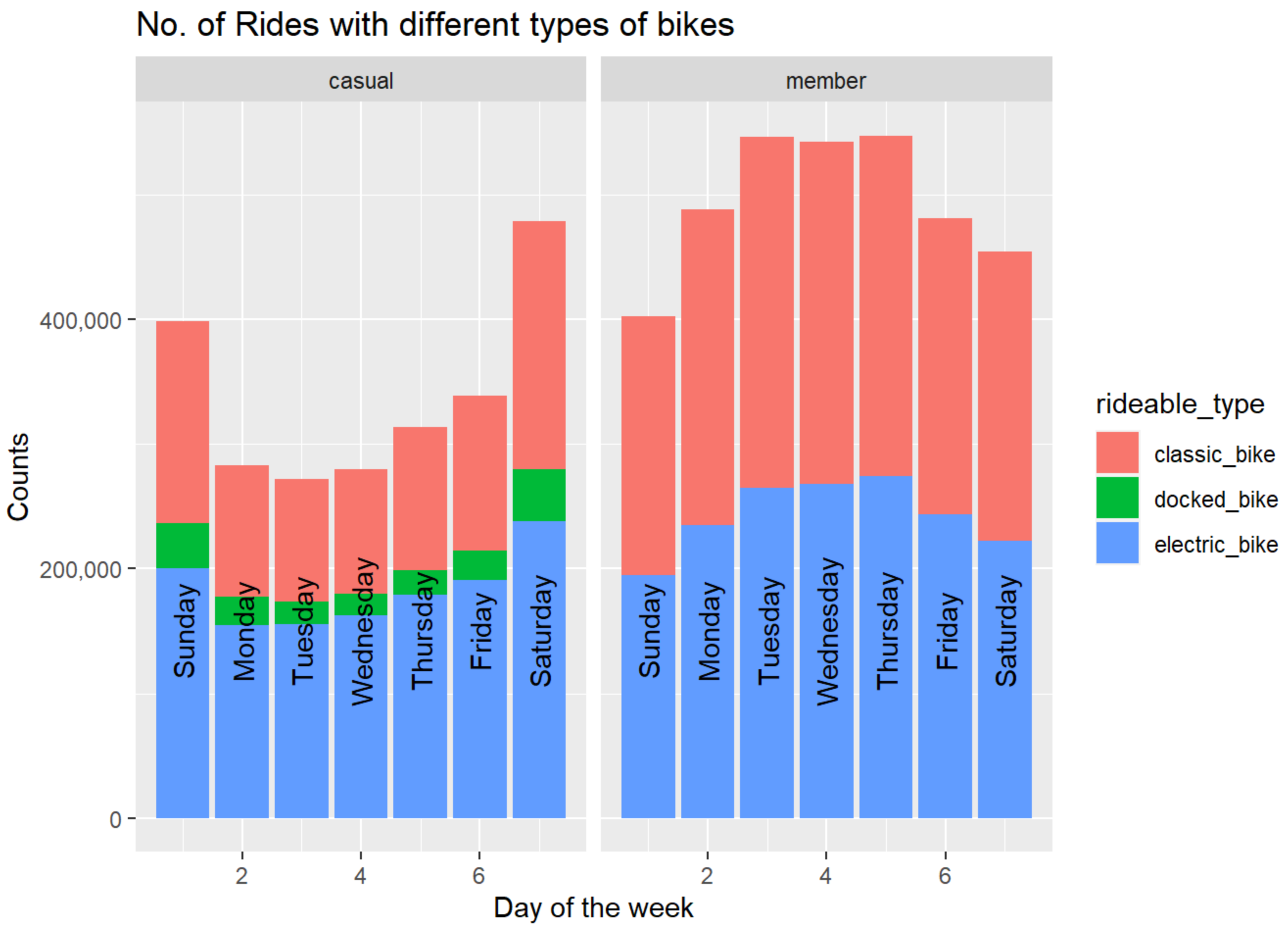
```
table(is.na(alldata$day_of_week))
```

```
##
##   FALSE
## 5829084
```

Analysis

Let’s plot the counts of bike rides on each day of the week with different types of riders

```
ggplot(data=alldata) +
  geom_bar(mapping=aes(x=day_of_week, fill=rideable_type)) +
  facet_wrap(~member_casual) +
  scale_y_continuous(labels = scales::comma) +
  annotate("text",x=1, y=150000, label="Sunday", angle=90)+
  annotate("text",x=2, y=150000, label="Monday", angle=90)+
  annotate("text",x=3, y=150000, label="Tuesday", angle=90)+
  annotate("text",x=4, y=150000, label="Wednesday", angle=90)+
  annotate("text",x=5, y=150000, label="Thursday", angle=90)+
  annotate("text",x=6, y=150000, label="Friday", angle=90)+
  annotate("text",x=7, y=150000, label="Saturday", angle=90) +
  labs(title="No. of Rides with different types of bikes", x="Day of the week", y="Counts")
```



```
ggsave("No. of Rides with different types of bikes.png")
```

```
## Saving 7 x 5 in image
```

Manipulating Data

Add some important column which will be useful for our analysis to the *alldata* dataset like *date*, *month*, *day*, *year*

```
date <- c(1:5829084)
month <- c(1:5829084)
day <- c(1:5829084)
year <- c(1:5829084)
df <- data.frame(date,month,day,year)
df$date[date!=""] <- NA
df$month[month!=""] <- NA
df$day[day!=""] <- NA
df$year[year!=""] <- NA
```

```
alldata <- cbind(alldata, df)
```

As *started_at* is in the “character” type so lets first convert it to “factor(integer)” form for further making it separate *DateTime* format to *Date*

```
alldata$started_at <- factor(alldata$started_at)
```

```
typeof(alldata$started_at)
```

```
## [1] "integer"
```

Now, convert “factor” to “DateTime” format

```
alldata$started_at <- mdy_hm(alldata$started_at)
```

and finally separate the Date out

```
alldata$date <- as.Date(alldata$started_at)
```

Now separate year, month and day

```
alldata$year <- year(alldata$started_at)
```

```
alldata$month <- month(alldata$started_at)
```

```
alldata$day <- day(alldata$started_at)
```

Plotting Data

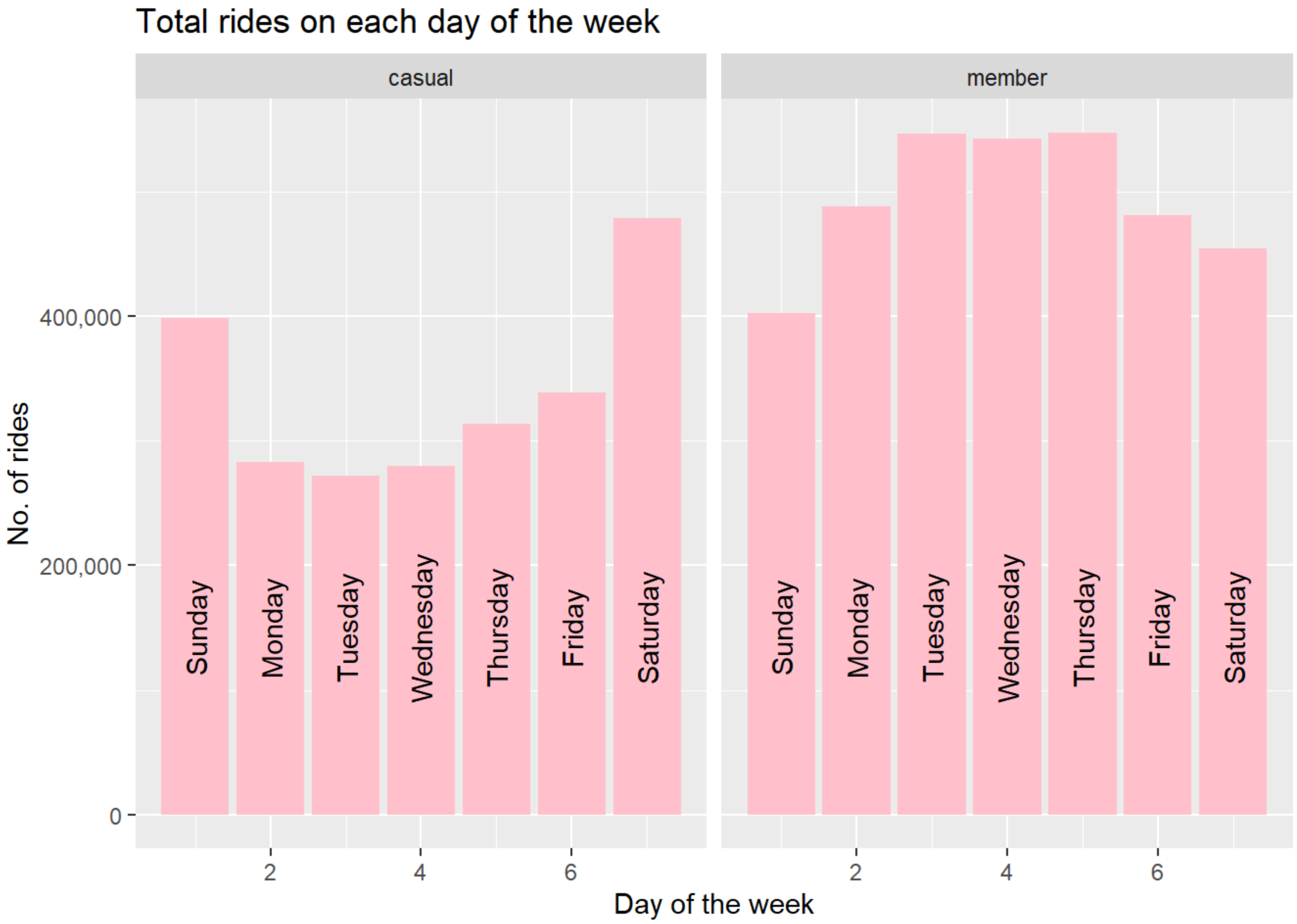
Let’s first separate out a dataset in which there will be data of membership type, day, and count of number of rides in a grouped form…

```
rides_data <- alldata %>%
  group_by(member_casual, day_of_week) %>%
  summarize(number_of_rides=n()) %>%
  select(member_casual, day_of_week, number_of_rides)
```

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

Plotting the data

```
ggplot(data=rides_data) +
  geom_col(mapping=aes(x=day_of_week, y=number_of_rides), fill="pink") +
  facet_wrap(~member_casual) +
  scale_y_continuous(labels = scales::comma) +
  annotate("text",x=1, y=150000, label="Sunday", angle=90)+
  annotate("text",x=2, y=150000, label="Monday", angle=90)+
  annotate("text",x=3, y=150000, label="Tuesday", angle=90)+
  annotate("text",x=4, y=150000, label="Wednesday", angle=90)+
  annotate("text",x=5, y=150000, label="Thursday", angle=90)+
  annotate("text",x=6, y=150000, label="Friday", angle=90)+
  annotate("text",x=7, y=150000, label="Saturday", angle=90)+
  labs(title="Total rides on each day of the week", x="Day of the week", y="No. of rides")
```



```
ggsave("Total rides on each day of the week.png")
```

```
## Saving 7 x 5 in image
```

Manipulating Data

Let’s find the total ride time to gain some insights
First convert *ride_time* from “character” to “integer” in *alldata* dataset

```
alldata$ride_time <- factor(alldata$ride_time)
```

Add one column *average_ride_time* into our newly created dataset

```
average_ride_time <- c(1:14)
average_ride_time <- data.frame(average_ride_time)
average_ride_time[!is.na(average_ride_time)] <- NA
rides_data <- cbind(rides_data, average_ride_time)
```

Manually enter the values using this code to find the mean of each row in our new dataset, **rides_data**

```
mean(times((alldata %>% filter(member_casual=="member", day_of_week==7))$ride_time))
```

```
## Warning in unpaste(times, sep = fmt$sep, fnames = fmt$periods, nfields = 3):
## wrong number of fields in entry(ies) 171738, 190853, 266039, 266043, 266044,
## 266045, 266366, 454733
```

```
## Warning in convert.times(times., fmt): 101 time-of-day entries out of range set
## to NA
```

```
## [1] 00:13:43
```

Fill the data found by the above code

```
rides_data$average_ride_time[rides_data$member_casual=="casual"&rides_data$day_of_week==1] <- "00:24:49"
rides_data$average_ride_time[rides_data$member_casual=="casual"&rides_data$day_of_week==2] <- "00:22:02"
rides_data$average_ride_time[rides_data$member_casual=="casual"&rides_data$day_of_week==3] <- "00:19:17"
rides_data$average_ride_time[rides_data$member_casual=="casual"&rides_data$day_of_week==4] <- "00:18:38"
rides_data$average_ride_time[rides_data$member_casual=="casual"&rides_data$day_of_week==5] <- "00:19:20"
rides_data$average_ride_time[rides_data$member_casual=="casual"&rides_data$day_of_week==6] <- "00:20:20"
rides_data$average_ride_time[rides_data$member_casual=="casual"&rides_data$day_of_week==7] <- "00:24:24"
```

```
rides_data$average_ride_time[rides_data$member_casual=="member"&rides_data$day_of_week==1] <- "00:13:36"
rides_data$average_ride_time[rides_data$member_casual=="member"&rides_data$day_of_week==2] <- "00:11:50"
rides_data$average_ride_time[rides_data$member_casual=="member"&rides_data$day_of_week==3] <- "00:11:37"
rides_data$average_ride_time[rides_data$member_casual=="member"&rides_data$day_of_week==4] <- "00:11:42"
rides_data$average_ride_time[rides_data$member_casual=="member"&rides_data$day_of_week==5] <- "00:11:52"
rides_data$average_ride_time[rides_data$member_casual=="member"&rides_data$day_of_week==6] <- "00:12:05"
rides_data$average_ride_time[rides_data$member_casual=="member"&rides_data$day_of_week==7] <- "00:13:43"
```

```
typeof(rides_data$average_ride_time)
```

```
## [1] "character"
```

Values filled is of “character” format, converting into “factor”

```
rides_data$average_ride_time <- factor(rides_data$average_ride_time)
```

```
summary(rides_data)
```

```
##  member_casual       day_of_week   number_of_rides  average_ride_time
##  Length:14          Min.   :1.00   Min.   :272322   00:11:37:1
##  Class :character   1st Qu.:2.25   1st Qu.:319933   00:11:42:1
##  Mode  :character   Median :4.00   Median :428844   00:11:50:1
##                     Mean   :4.00   Mean   :416363   00:11:52:1
##                     3rd Qu.:5.75   3rd Qu.:486771   00:12:05:1
##                     Max.   :7.00   Max.   :547400   00:13:36:1
##                                                     (Other) :8
```

Just verifying…

```
sum(rides_data$number_of_rides)
```

```
## [1] 5829084
```

Making data more readable by making some minor upgrades

```
rides_data$day_of_week[rides_data$day_of_week==1] <- "Sunday"
rides_data$day_of_week[rides_data$day_of_week==2] <- "Monday"
rides_data$day_of_week[rides_data$day_of_week==3] <- "Tuesday"
rides_data$day_of_week[rides_data$day_of_week==4] <- "Wednesday"
rides_data$day_of_week[rides_data$day_of_week==5] <- "Thursday"
rides_data$day_of_week[rides_data$day_of_week==6] <- "Friday"
rides_data$day_of_week[rides_data$day_of_week==7] <- "Saturday"
```

Export New Dataset

```
write.csv(rides_data, "Final Analysis.csv")
```

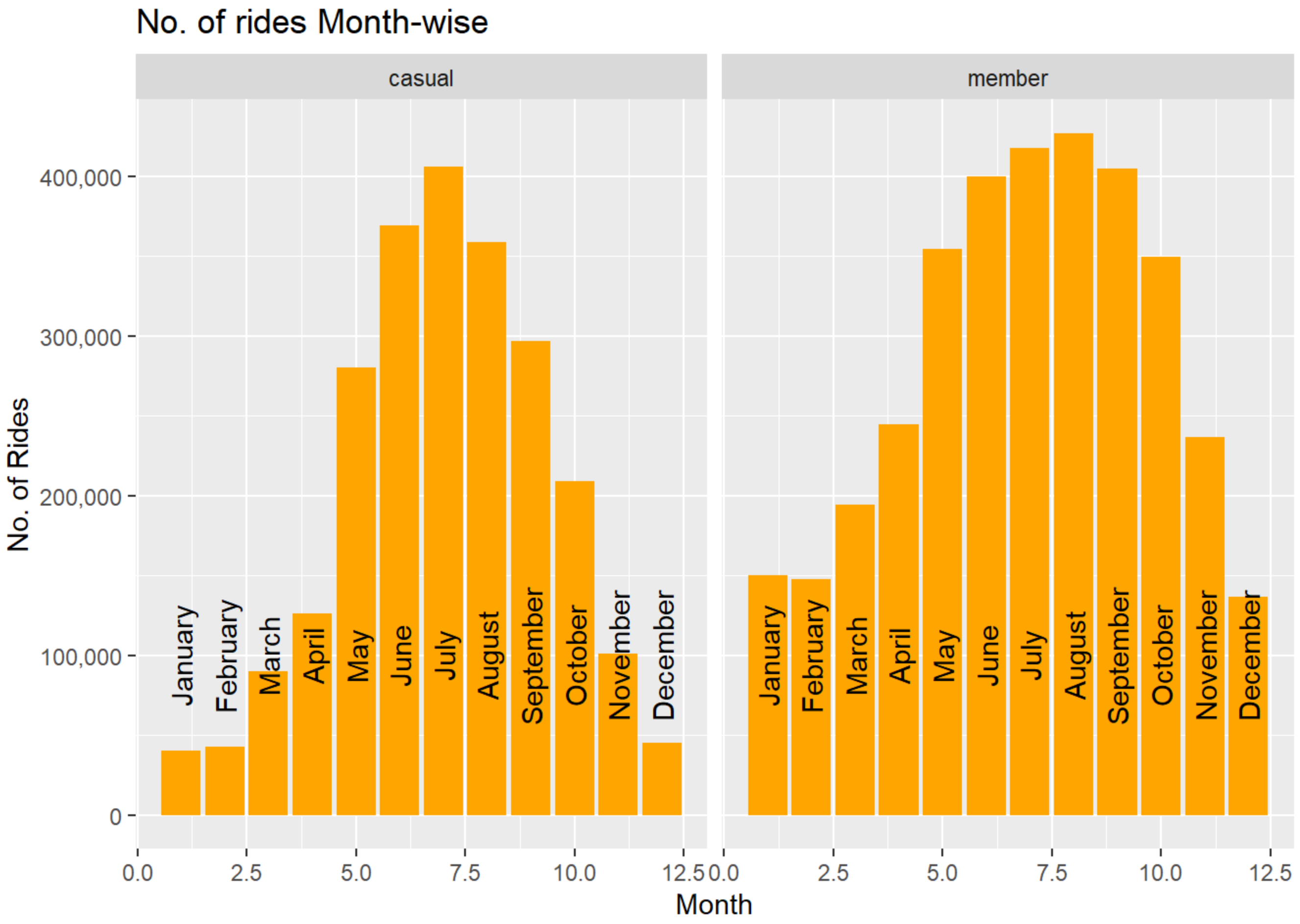
Create another dataset to analysis number of rides month-wise

```
rides_data_monthly <- alldata %>%
  group_by(member_casual, month) %>%
  summarize(number_of_rides=n()) %>%
  select(member_casual, month, number_of_rides)
```

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

Plot the data…

```
ggplot(data=rides_data_monthly) +
  geom_col(mapping=aes(x=month, y=number_of_rides), fill="orange")+
  facet_wrap(~member_casual)+
  scale_y_continuous(labels = scales::comma)+
  labs(title = "No. of rides Month-wise", x="Month", y="No. of Rides")+
  annotate("text",x=1, y=100000, label="January", angle=90)+
  annotate("text",x=2, y=100000, label="February", angle=90)+
  annotate("text",x=3, y=100000, label="March", angle=90)+
  annotate("text",x=4, y=100000, label="April", angle=90)+
  annotate("text",x=5, y=100000, label="May", angle=90)+
  annotate("text",x=6, y=100000, label="June", angle=90)+
  annotate("text",x=7, y=100000, label="July", angle=90)+
  annotate("text",x=8, y=100000, label="August", angle=90)+
  annotate("text",x=9, y=100000, label="September", angle=90)+
  annotate("text",x=10, y=100000, label="October", angle=90)+
  annotate("text",x=11, y=100000, label="November", angle=90)+
  annotate("text",x=12, y=100000, label="December", angle=90)
```



```
ggsave("No. of rides Month-wise.png")
```

```
## Saving 7 x 5 in image
```

Make data more readable and export it for further possible use

```
rides_data_monthly$month[rides_data_monthly$month==1] <- "January"
rides_data_monthly$month[rides_data_monthly$month==2] <- "February"
rides_data_monthly$month[rides_data_monthly$month==3] <- "March"
rides_data_monthly$month[rides_data_monthly$month==4] <- "April"
rides_data_monthly$month[rides_data_monthly$month==5] <- "May"
rides_data_monthly$month[rides_data_monthly$month==6] <- "June"
rides_data_monthly$month[rides_data_monthly$month==7] <- "July"
rides_data_monthly$month[rides_data_monthly$month==8] <- "August"
rides_data_monthly$month[rides_data_monthly$month==9] <- "September"
rides_data_monthly$month[rides_data_monthly$month==10] <- "October"
rides_data_monthly$month[rides_data_monthly$month==11] <- "November"
rides_data_monthly$month[rides_data_monthly$month==12] <- "December"
```

```
write.csv(rides_data_monthly, "Monthly Analysis.csv")
```

Now after applying formula in the Excel, update the dataset to add the hour of rides by every row in *rides_data* dataset

```
average_time_analysis <- read.csv("Average Time Analysis in Hour.csv")
```

```
head(average_time_analysis)
```

```
##   member_casual day_of_week number_of_rides average_ride_time  time total_time
## 1        casual      Sunday          398647           0:24:49 24.82    9894419
## 2        casual      Monday          283327           0:22:02 22.03    6241694
## 3        casual     Tuesday          272322           0:19:17 19.28    5250368
## 4        casual   Wednesday          279897           0:18:38 18.63    5214481
## 5        casual    Thursday          313642           0:19:20 19.33    6062700
## 6        casual      Friday          338806           0:20:20 20.33    6887926
##     in_hour
## 1 164906.98
## 2 104028.23
## 3  87506.14
## 4  86908.02
## 5 101045.00
## 6 114798.77
```

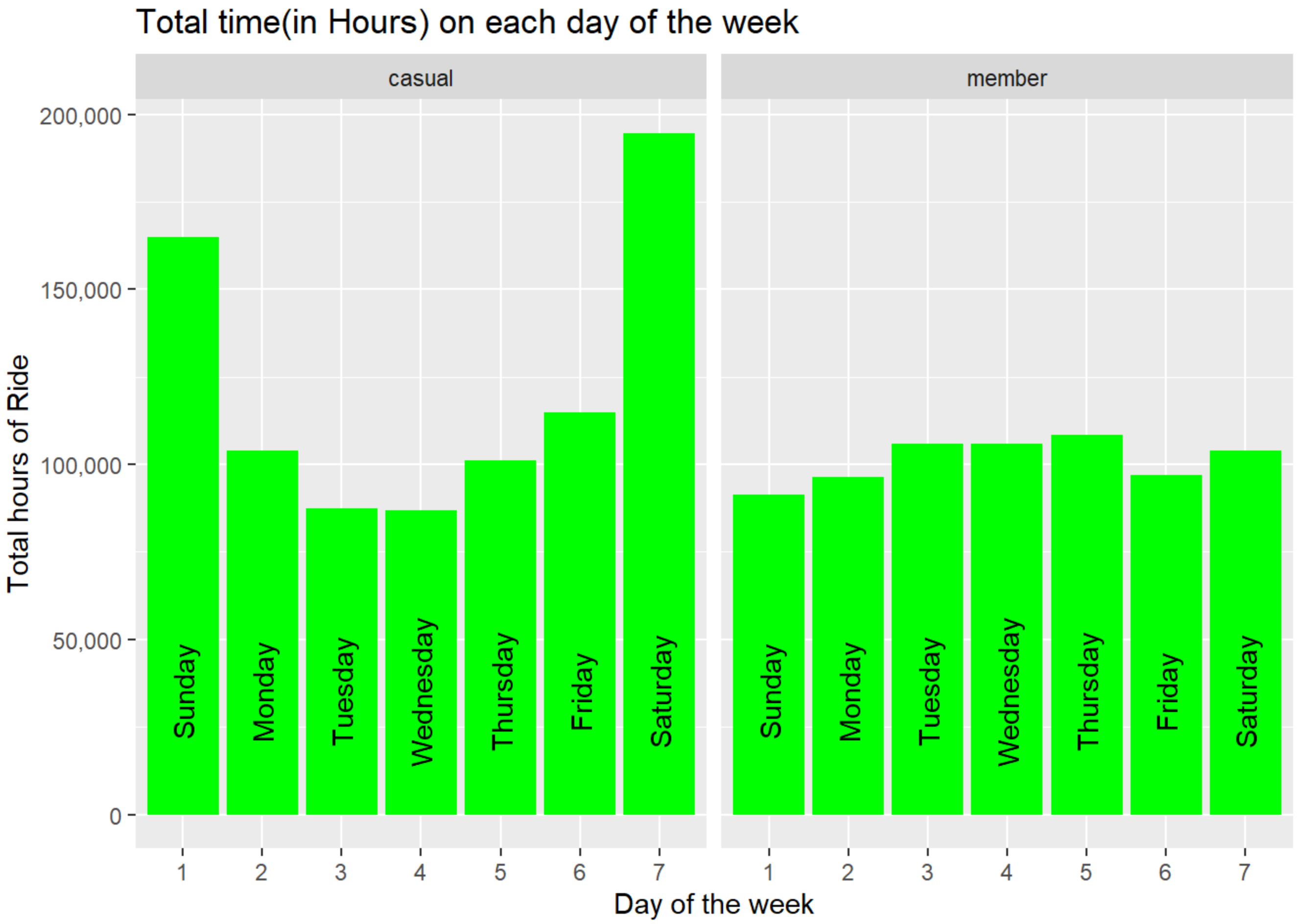
```
summary(average_time_analysis)
```

```
##  member_casual      day_of_week        number_of_rides  average_ride_time 
##  Length:14          Length:14          Min.   :272322   Length:14         
##  Class :character   Class :character   1st Qu.:319933   Class :character  
##  Mode  :character   Mode  :character   Median :428844   Mode  :character  
##                                        Mean   :416363                     
##                                        3rd Qu.:486771                     
##                                        Max.   :547400                     
##       time         total_time          in_hour      
##  Min.   :11.62   Min.   : 5214481   Min.   : 86908  
##  1st Qu.:11.92   1st Qu.: 5788576   1st Qu.: 96476  
##  Median :16.18   Median : 6240898   Median :104015  
##  Mean   :16.80   Mean   : 6695339   Mean   :111589  
##  3rd Qu.:20.08   3rd Qu.: 6461160   3rd Qu.:107686  
##  Max.   :24.82   Max.   :11674888   Max.   :194581  
```

```
average_time_analysis$day_of_week[average_time_analysis$day_of_week=="Sunday"] <- 1
average_time_analysis$day_of_week[average_time_analysis$day_of_week=="Monday"] <- 2
average_time_analysis$day_of_week[average_time_analysis$day_of_week=="Tuesday"] <- 3
average_time_analysis$day_of_week[average_time_analysis$day_of_week=="Wednesday"] <- 4
average_time_analysis$day_of_week[average_time_analysis$day_of_week=="Thursday"] <- 5
average_time_analysis$day_of_week[average_time_analysis$day_of_week=="Friday"] <- 6
average_time_analysis$day_of_week[average_time_analysis$day_of_week=="Saturday"] <- 7
```

Plotting data to analyze ride hours and days of the week relationship

```
ggplot(data=average_time_analysis) +
  geom_col(mapping=aes(x=day_of_week, y=in_hour), fill="green")+
  facet_wrap(~member_casual) +
  scale_y_continuous(labels = scales::comma) +
  annotate("text",x=1, y=35000, label="Sunday", angle=90)+
  annotate("text",x=2, y=35000, label="Monday", angle=90)+
  annotate("text",x=3, y=35000, label="Tuesday", angle=90)+
  annotate("text",x=4, y=35000, label="Wednesday", angle=90)+
  annotate("text",x=5, y=35000, label="Thursday", angle=90)+
  annotate("text",x=6, y=35000, label="Friday", angle=90)+
  annotate("text",x=7, y=35000, label="Saturday", angle=90)+
  labs(title="Total time(in Hours) on each day of the week", x="Day of the week", y="Total hours o
f Ride")
```



```
ggsave("Total time(in Hours) on each day of the week.png")
```

```
## Saving 7 x 5 in image
```

Based on this Analysis, we will make a presentation regarding our business goal

Thank you…