Cyclistic Case Study Jul21

Hezar K

2022-11-29

This is an analysis for Cyclistic Case Study for Google Data Analytics Course. This is an analysis for July 2021.

STEP ONE: INSTALL REQUIRED PACKAGES AND IMPORT DATA

Install the required packages. **Tidyverse** package to import and wrangling the data and **ggplot2** package for visualization of the data. **Lubridate** package for date parsing and **anytime** package for the datetime conversion.

- install.packages("tidyverse")
- install.packages("ggplot2")
- install.packages("lubridate")
- install.packages("anytime")

```
library(tidyverse)
library(lubridate)
library(data.table)
library(ggplot2)
library(anytime)
```

Import data from local drive.

```
Jul21 <- read_csv("C:/Users/theby/Documents/202107-divvy-tripdata.csv")
```

STEP TWO: EXAMINE THE DATA

Examine the dataframe for an overview of the data. Review column names, **colnames()**, dimensions of the dataframe by row and column, **dim()**, the first, **head()**, and the last, **tail()**, six rows in the dataframe, the summary, **summary()**, statistics on the columns of the dataframe, and review the data type structure of columns, **str()**.

View(Jul21)

```
colnames(Jul21)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"        
##  [4] "ended_at"           "start_station_name" "start_station_id"  
##  [7] "end_station_name"   "end_station_id"     "start_lat"         
## [10] "start_lng"          "end_lat"            "end_lng"           
## [13] "member_casual"
```

```
nrow(Jul21)
```

```
## [1] 822410
```

```
dim(Jul21)
```

```
## [1] 822410     13
```

```
head(Jul21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>  
## 1 0A1B623926EF4… docked… 2021-07-02 14:44:36 2021-07-02 15:19:58 Michig… 13001  
## 2 B2D5583A5A5E7… classi… 2021-07-07 16:57:42 2021-07-07 17:16:09 Califo… 17660  
## 3 6F264597DDBF4… classi… 2021-07-25 11:30:55 2021-07-25 11:48:45 Wabash… SL-012 
## 4 379B58EAB20E8… classi… 2021-07-08 22:08:30 2021-07-08 22:23:32 Califo… 17660  
## 5 6615C1E4EB08E… electr… 2021-07-28 16:08:06 2021-07-28 16:27:09 Califo… 17660  
## 6 62DC2B32872F9… electr… 2021-07-29 17:09:08 2021-07-29 17:15:00 Califo… 17660  
## # … with 7 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names ¹rideable_type,
## #   ²start_station_name, ³start_station_id
```

```
tail(Jul21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 7B47CA3E874D2… electr… 2021-07-04 05:34:53 2021-07-04 05:36:46 <NA>    <NA>
## 2 1E660BF8DCDAA… electr… 2021-07-04 10:40:41 2021-07-04 11:30:13 <NA>    <NA>
## 3 A2448BDFD9B36… electr… 2021-07-04 12:47:41 2021-07-04 12:54:46 <NA>    <NA>
## 4 2D612BF853037… electr… 2021-07-03 21:41:58 2021-07-03 21:57:14 <NA>    <NA>
## 5 6D615D18B765C… electr… 2021-07-03 22:10:31 2021-07-03 22:11:39 <NA>    <NA>
## 6 0F31D311323F0… electr… 2021-07-04 07:03:50 2021-07-04 07:32:38 <NA>    <NA>
## # … with 7 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names ¹rideable_type,
## #   ²start_station_name, ³start_station_id
```

```
summary(Jul21)
```

```
##    ride_id          rideable_type        started_at                
##  Length:822410      Length:822410      Min.   :2021-07-01 00:00:22.00  
##  Class :character   Class :character   1st Qu.:2021-07-08 17:44:35.00  
##  Mode  :character   Mode  :character   Median :2021-07-17 13:58:37.00  
##                                        Mean   :2021-07-16 22:23:15.46  
##                                        3rd Qu.:2021-07-24 18:23:39.25  
##                                        Max.   :2021-07-31 23:59:58.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2021-07-01 00:04:51.00   Length:822410      Length:822410     
##  1st Qu.:2021-07-08 18:02:01.25   Class :character   Class :character  
##  Median :2021-07-17 14:28:04.50   Mode  :character   Mode  :character  
##  Mean   :2021-07-16 22:47:28.09                                        
##  3rd Qu.:2021-07-24 18:46:20.25                                        
##  Max.   :2021-08-12 17:45:41.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:822410      Length:822410      Min.   :41.65   Min.   :-87.84  
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66  
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64  
##                                        Mean   :41.90   Mean   :-87.65  
##                                        3rd Qu.:41.93   3rd Qu.:-87.63  
##                                        Max.   :42.07   Max.   :-87.52  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   :41.63   Min.   :-87.85   Length:822410     
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character  
##  Median :41.90   Median :-87.64   Mode  :character  
##  Mean   :41.90   Mean   :-87.65                     
##  3rd Qu.:41.93   3rd Qu.:-87.63                     
##  Max.   :42.15   Max.   :-87.49                     
##  NA's   :731     NA's   :731
```

```
str(Jul21)
```

```
## spc_tbl_ [822,410 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:822410] "0A1B623926EF4E16" "B2D5583A5A5E76EE" "6F264597DDBF427A" "379B58EAB20E8A
A5" ...
##  $ rideable_type     : chr [1:822410] "docked_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:822410], format: "2021-07-02 14:44:36" "2021-07-07 16:57:42" ...
##  $ ended_at          : POSIXct[1:822410], format: "2021-07-02 15:19:58" "2021-07-07 17:16:09" ...
##  $ start_station_name: chr [1:822410] "Michigan Ave & Washington St" "California Ave & Cortez St" "Wabash Ave
& 16th St" "California Ave & Cortez St" ...
##  $ start_station_id  : chr [1:822410] "13001" "17660" "SL-012" "17660" ...
##  $ end_station_name  : chr [1:822410] "Halsted St & North Branch St" "Wood St & Hubbard St" "Rush St & Hubbard
St" "Carpenter St & Huron St" ...
##  $ end_station_id    : chr [1:822410] "KA1504000117" "13432" "KA1503000044" "13196" ...
##  $ start_lat         : num [1:822410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:822410] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num [1:822410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:822410] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr [1:822410] "casual" "casual" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Create new columns as for *date*, *month*, *day*, *year*, *day_of_week*, and *ride_length* in seconds.

```
Jul21$date <- as.Date(Jul21$started_at)
Jul21$month <- format(as.Date(Jul21$date), "%m")
Jul21$month <- month.name[as.numeric(Jul21$month)]
Jul21$day <- format(as.Date(Jul21$date), "%d")
Jul21$year <- format(as.Date(Jul21$date), "%Y")
Jul21$day_of_week <- format(as.Date(Jul21$date), "%A")
Jul21$ride_length <- difftime(Jul21$ended_at,Jul21$started_at)
```

Convert *ride_length* column to numeric in order to run calculations on the data. First, check to see if the data type is numeric, and then convert if needed.

```
is.numeric(Jul21$ride_length)
```

```
## [1] FALSE
```

Recheck *ride_length* data type.

```
Jul21$ride_length <- as.numeric(as.character(Jul21$ride_length))
is.numeric(Jul21$ride_length)
```

```
## [1] TRUE
```

STEP THREE: CLEAN DATA

na.omit() will remove all NA from the dataframe.

```
Jul21 <- na.omit(Jul21)
```

Remove rows with the *ride_id* column character length is not 16. This will remove all the scientific ride ids that we noticed while examining the data.

```
Jul21 <- subset(Jul21, nchar(as.character(ride_id)) == 16)
```

Remove rows with the *ride_length* less than 60 seconds or 1 minute.

```
Jul21 <- subset (Jul21, ride_length > 59)
```

STEP FOUR: ANALYZE DATA

Analyze the dataframe by find the **mean**, **median**, **max** (maximum), and **min** (minimum) of *ride_length*.

```
mean(Jul21$ride_length)
```

```
## [1] 1470.582
```

```
median(Jul21$ride_length)
```

```
## [1] 818
```

```
max(Jul21$ride_length)
```

```
## [1] 2946429
```

```
min(Jul21$ride_length)
```

```
## [1] 60
```

Run a statistical summary of the *ride_length*.

```
summary(Jul21$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      60     475     818    1471    1453 2946429
```

Compare the members and casual users

```
aggregate(Jul21$ride_length ~ Jul21$member_casual, FUN = mean)
```

```
##   Jul21$member_casual Jul21$ride_length
## 1              casual          2018.103
## 2              member           840.641
```

```
aggregate(Jul21$ride_length ~ Jul21$member_casual, FUN = median)
```

```
##   Jul21$member_casual Jul21$ride_length
## 1              casual              1042
## 2              member               634
```

```
aggregate(Jul21$ride_length ~ Jul21$member_casual, FUN = max)
```

```
##   Jul21$member_casual Jul21$ride_length
## 1              casual           2946429
## 2              member             75757
```

```
aggregate(Jul21$ride_length ~ Jul21$member_casual, FUN = min)
```

```
##   Jul21$member_casual Jul21$ride_length
## 1              casual                60
## 2              member                60
```

Aggregate the average ride length by each day of the week for members and users.

```
aggregate(Jul21$ride_length ~ Jul21$member_casual + Jul21$day_of_week, FUN = mean)
```

```
##    Jul21$member_casual Jul21$day_of_week Jul21$ride_length
## 1               casual            Friday         1889.3437
## 2               member            Friday          812.2277
## 3               casual            Monday         2252.2579
## 4               member            Monday          829.5126
## 5               casual          Saturday         2146.0160
## 6               member          Saturday          939.8370
## 7               casual            Sunday         2255.1447
## 8               member            Sunday          956.6862
## 9               casual          Thursday         1897.4272
## 10              member          Thursday          794.2570
## 11              casual           Tuesday         1707.8050
## 12              member           Tuesday          789.3299
## 13              casual         Wednesday         1747.2477
## 14              member         Wednesday          796.7211
```

Sort the days of the week in order.

```
Jul21$day_of_week <- ordered(Jul21$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"))
```

Assign the aggregate the average ride length by each day of the week for members and users to x.

```
x <- aggregate(Jul21$ride_length ~ Jul21$member_casual + Jul21$day_of_week, FUN = mean)

head(x)
```

```
##   Jul21$member_casual Jul21$day_of_week Jul21$ride_length
## 1              casual            Sunday         2255.1447
## 2              member            Sunday          956.6862
## 3              casual            Monday         2252.2579
## 4              member            Monday          829.5126
## 5              casual           Tuesday         1707.8050
## 6              member           Tuesday          789.3299
```

Find the average ride length of member riders and casual riders per day and assign it to y.

```
y <- Jul21 %>%
  mutate(weekday = wday(started_at)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, weekday)

head(y)
```

```
## # A tibble: 6 × 4
##   member_casual weekday number_of_rides average_duration
##   <chr>           <int>           <int>            <dbl>
## 1 casual              1           58912            2255.
## 2 casual              2           39922            2252.
## 3 casual              3           36337            1708.
## 4 casual              4           37575            1747.
## 5 casual              5           46130            1897.
## 6 casual              6           58634            1889.
```

Analyze the dataframe to find the frequency of member riders, casual riders, classic bikes, docked bikes, and electric bikes.

```
table(Jul21$member_casual)
```

```
## 
## casual member 
## 365513 317690
```

```
table(Jul21$rideable_type)
```

```
## 
##  classic_bike   docked_bike electric_bike 
##        498616         57346        127241
```

```
table(Jul21$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##     91690     79008     80381     82755    100285    111725    137359
```

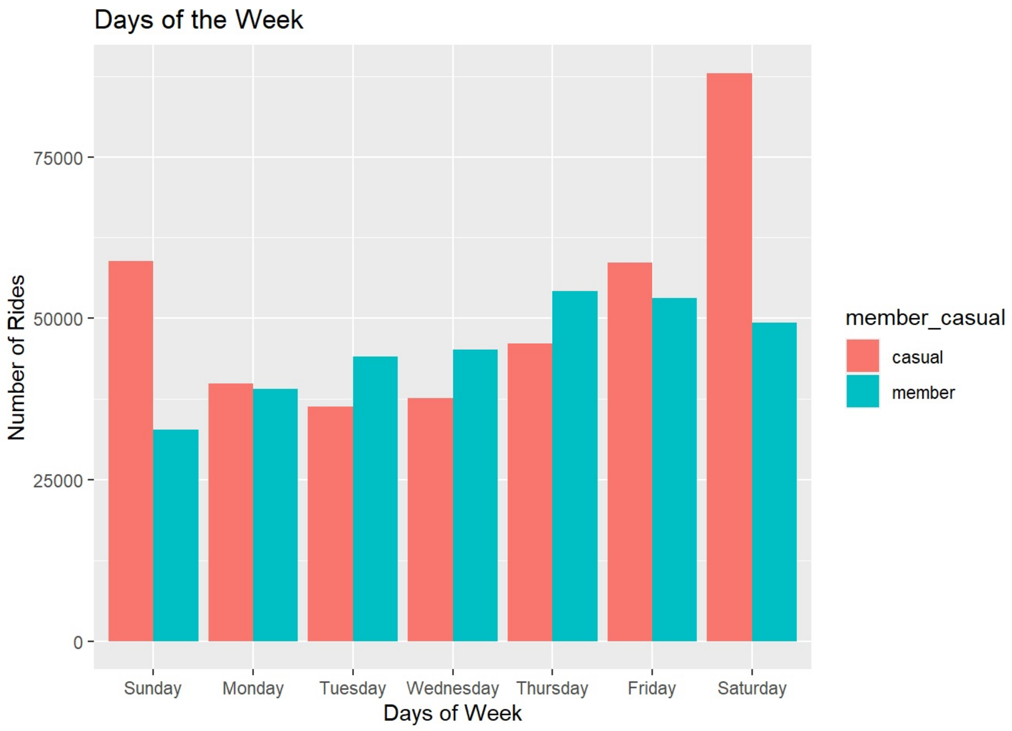
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

```
options(scipen=999)
```

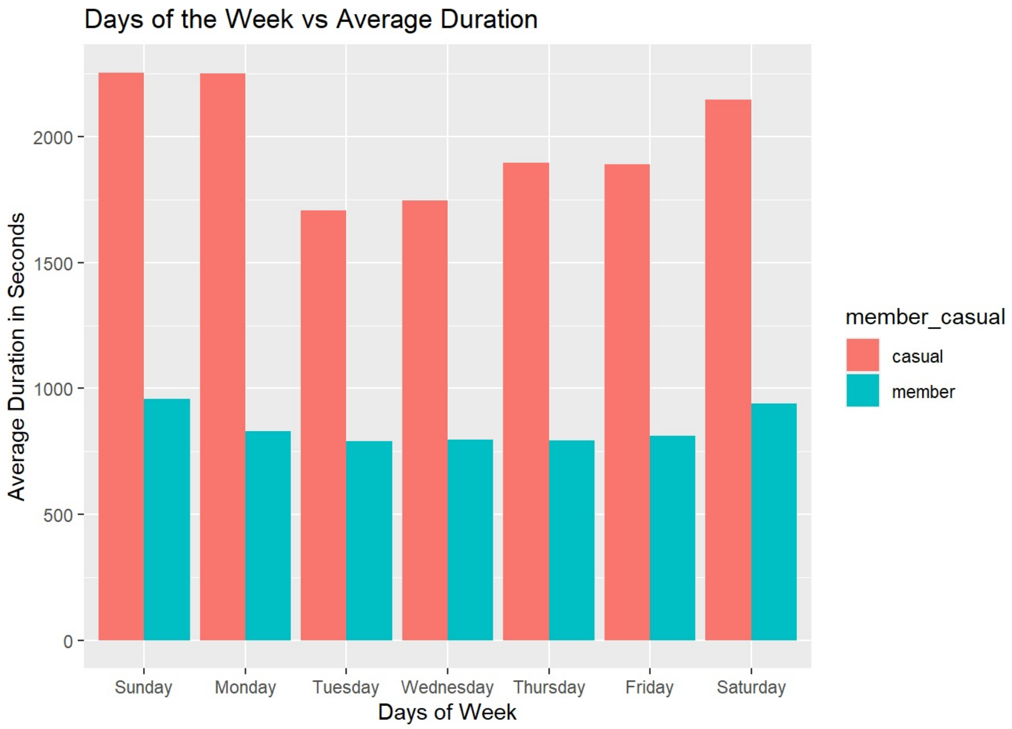
Plot the number of rides by user type during the week.

```
Jul21 %>%
  mutate(day_of_week) %>%
  group_by(member_casual,day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")+
labs(x = "Days of Week",
     y= "Number of Rides",
     title= "Days of the Week")
```



Plot the duration of the ride by user type during the week.

```
Jul21 %>%
  mutate(day_of_week) %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(x = "Days of Week",
       y= "Average Duration in Seconds",
       title= "Days of the Week vs Average Duration")
```



Create new dataframe for plots for weekday trends vs weekend trends.

```
mc<- as.data.frame(table(Jul21$day_of_week,Jul21$member_casual))
```

Rename columns

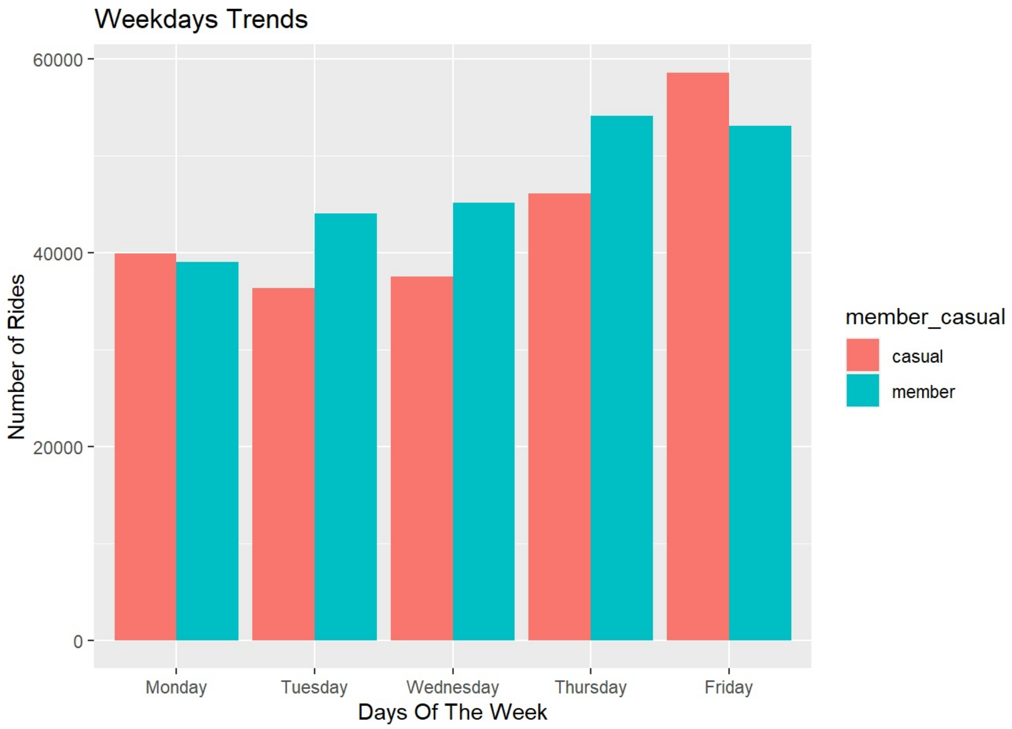
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)

head(mc)
```

```
##   day_of_week member_casual  Freq
## 1      Sunday        casual 58912
## 2      Monday        casual 39922
## 3     Tuesday        casual 36337
## 4   Wednesday        casual 37575
## 5    Thursday        casual 46130
## 6      Friday        casual 58634
```

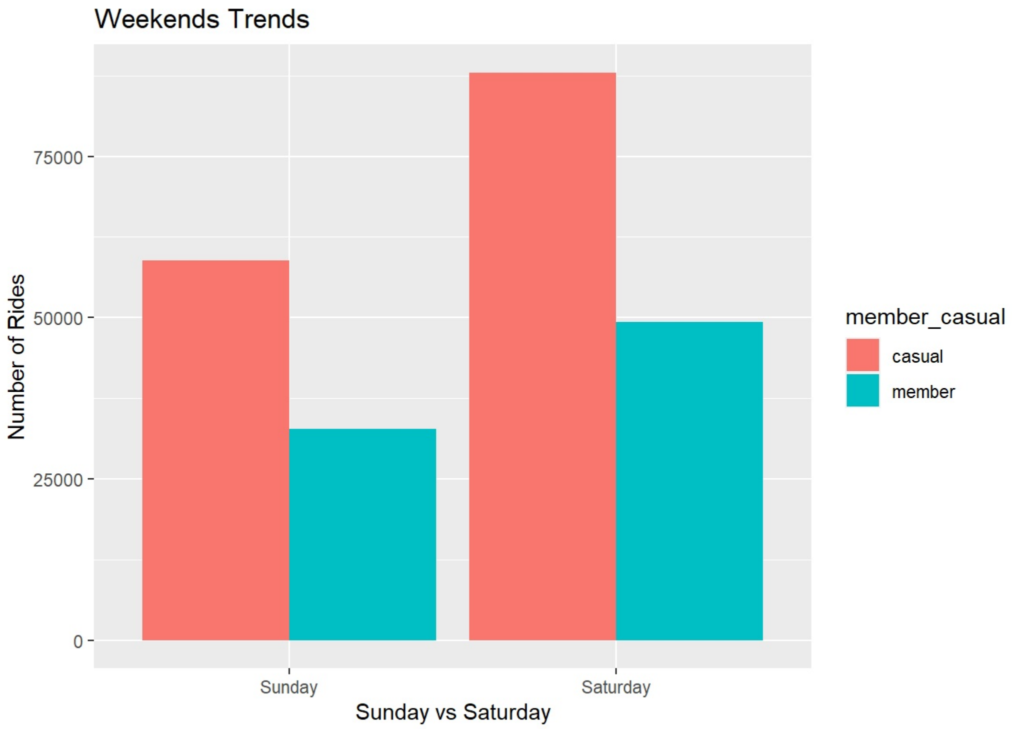
Weekday trends (Monday through Friday).

```
mc %>%
  filter(day_of_week == "Monday" |
           day_of_week == "Tuesday" |
           day_of_week == "Wednesday" |
           day_of_week == "Thursday" |
           day_of_week == "Friday") %>%
  ggplot(aes(x = day_of_week, y = Freq, fill = member_casual))+
  geom_bar(stat = "identity" , position = "dodge") +
  labs(title = "Weekdays Trends",
       x= "Days Of The Week",
       y = "Number of Rides")
```



Weekend trends (Sunday and Saturday).

```
mc %>%
  filter(day_of_week == "Sunday" |
           day_of_week == "Saturday") %>%
  ggplot(aes(x = day_of_week, y = Freq, fill = member_casual))+
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Weekends Trends",
       x= "Sunday vs Saturday",
       y = "Number of Rides")
```



Create dataframe for member and casual riders vs ride type

```
rt<- as.data.frame(table(Jul21$rideable_type,Jul21$member_casual))
```

Rename columns.

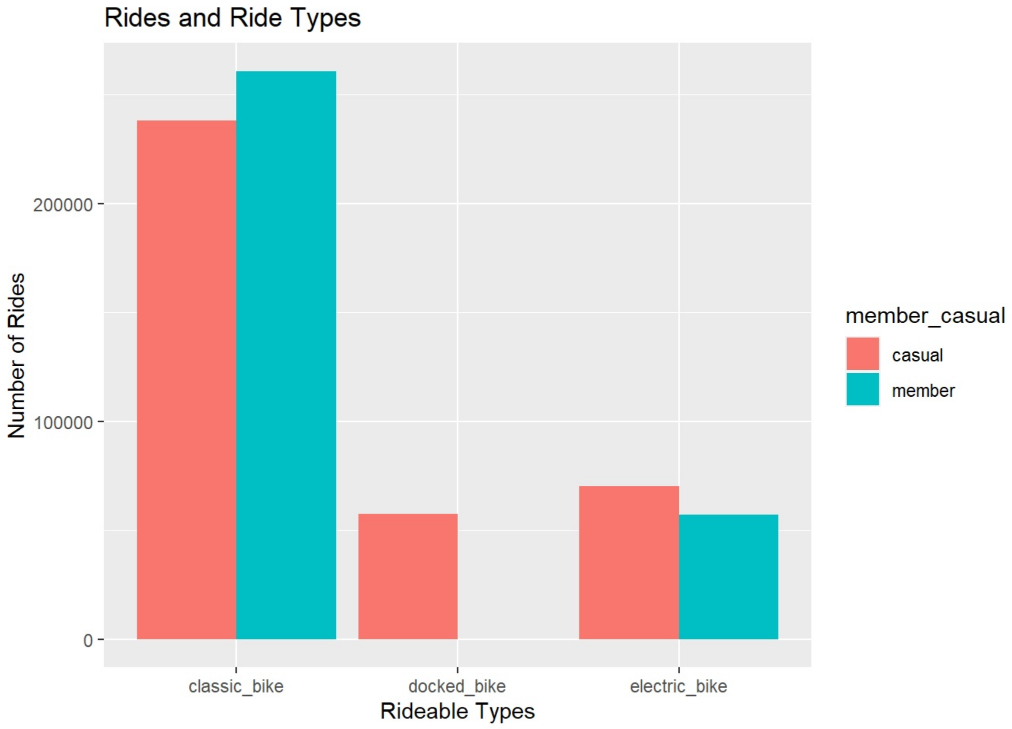
```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)

head(rt)
```

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual 237845
## 2   docked_bike        casual  57346
## 3 electric_bike        casual  70322
## 4  classic_bike        member 260771
## 5   docked_bike        member      0
## 6 electric_bike        member  56919
```

Plot for bike user vs bike type.

```
rt %>%
  filter(member_casual == "member" |
           member_casual == "casual") %>%
  ggplot(aes(x = rideable_type, y = Freq, fill = member_casual))+
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Rides and Ride Types",
       x= "Rideable Types",
       y = "Number of Rides")
```



STEP SIX: EXPORT ANALYZED DATA

Save the analyzed data as a new file. fwrite(Jul21, “Jul21.csv”)