Cyclistic Case Study Jun21

Hezar K

2022-11-29

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

```
Jun21 <- read_csv("C:/Users/theby/Documents/202106-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(Jun21)

```
colnames(Jun21)
```

```
##  [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(Jun21)
```

```
## [1] 729595
```

```
dim(Jun21)
```

```
## [1] 729595     13
```

```
head(Jun21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>  
## 1 99FEC93BA843F… electr… 2021-06-13 14:31:28 2021-06-13 14:34:11 <NA>    <NA>   
## 2 06048DCFC8520… electr… 2021-06-04 11:18:02 2021-06-04 11:24:19 <NA>    <NA>   
## 3 9598066F68045… electr… 2021-06-04 09:49:35 2021-06-04 09:55:34 <NA>    <NA>   
## 4 B03C0FE48C412… electr… 2021-06-03 19:56:05 2021-06-03 20:21:55 <NA>    <NA>   
## 5 B9EEA89F8FEE7… electr… 2021-06-04 14:05:51 2021-06-04 14:09:59 <NA>    <NA>   
## 6 62B943CEAAA42… electr… 2021-06-03 19:32:01 2021-06-03 19:38:46 <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
```

```
tail(Jun21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 547E5403EE677… electr… 2021-06-12 15:31:50 2021-06-12 16:38:22 Wells … SL-011
## 2 CB282292CCFCE… electr… 2021-06-14 00:17:31 2021-06-14 00:56:46 Wells … SL-011
## 3 47BD346FAFB9B… classi… 2021-06-30 17:35:10 2021-06-30 17:43:20 Clark … 13303
## 4 52467C23D17C6… classi… 2021-06-13 19:24:30 2021-06-13 19:34:11 Indian… TA1307…
## 5 7DF6D74420D7D… electr… 2021-06-08 15:44:28 2021-06-08 16:15:01 Clark … 13303
## 6 0C01F8BA99E51… electr… 2021-06-03 16:18:38 2021-06-03 16:47:49 Clark … 13303
## # … 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(Jun21)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:729595      Length:729595      Min.   :2021-06-01 00:00:38.00  
##  Class :character   Class :character   1st Qu.:2021-06-08 16:03:57.00  
##  Mode  :character   Mode  :character   Median :2021-06-14 19:46:47.00  
##                                        Mean   :2021-06-15 09:48:47.76  
##                                        3rd Qu.:2021-06-21 19:10:47.00  
##                                        Max.   :2021-06-30 23:59:59.00  
##                                                                        
##     ended_at                    start_station_name start_station_id  
##  Min.   :2021-06-01 00:06:22.00   Length:729595      Length:729595     
##  1st Qu.:2021-06-08 16:23:54.00   Class :character   Class :character  
##  Median :2021-06-14 20:13:55.00   Mode  :character   Mode  :character  
##  Mean   :2021-06-15 10:14:52.60                                        
##  3rd Qu.:2021-06-21 19:31:59.00                                        
##  Max.   :2021-07-13 22:51:35.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:729595      Length:729595      Min.   :41.64   Min.   :-87.78  
##  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   :-87.64  
##                                        3rd Qu.:41.93   3rd Qu.:-87.63  
##                                        Max.   :42.07   Max.   :-87.52  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   :41.51   Min.   :-87.86   Length:729595     
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character  
##  Median :41.90   Median :-87.64   Mode  :character  
##  Mean   :41.90   Mean   :-87.64                     
##  3rd Qu.:41.93   3rd Qu.:-87.63                     
##  Max.   :42.08   Max.   :-87.49                     
##  NA's   :717     NA's   :717
```

```
str(Jun21)
```

```
## spc_tbl_ [729,595 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:729595] "99FEC93BA843FB20" "06048DCFC8520CAF" "9598066F68045DF2" "B03C0FE48C4122
14" ...
##  $ rideable_type     : chr [1:729595] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:729595], format: "2021-06-13 14:31:28" "2021-06-04 11:18:02" ...
##  $ ended_at          : POSIXct[1:729595], format: "2021-06-13 14:34:11" "2021-06-04 11:24:19" ...
##  $ start_station_name: chr [1:729595] NA NA NA NA ...
##  $ start_station_id  : chr [1:729595] NA NA NA NA ...
##  $ end_station_name  : chr [1:729595] NA NA NA NA ...
##  $ end_station_id    : chr [1:729595] NA NA NA NA ...
##  $ start_lat         : num [1:729595] 41.8 41.8 41.8 41.8 41.8 ...
##  $ start_lng         : num [1:729595] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:729595] 41.8 41.8 41.8 41.8 41.8 ...
##  $ end_lng           : num [1:729595] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:729595] "member" "member" "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.

```
Jun21$date <- as.Date(Jun21$started_at)
Jun21$month <- format(as.Date(Jun21$date), "%m")
Jun21$month <- month.name[as.numeric(Jun21$month)]
Jun21$day <- format(as.Date(Jun21$date), "%d")
Jun21$year <- format(as.Date(Jun21$date), "%Y")
Jun21$day_of_week <- format(as.Date(Jun21$date), "%A")
Jun21$ride_length <- difftime(Jun21$ended_at,Jun21$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(Jun21$ride_length)
```

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

Recheck *ride_length* data type.

```
Jun21$ride_length <- as.numeric(as.character(Jun21$ride_length))
is.numeric(Jun21$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
Jun21 <- na.omit(Jun21)
```

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.

```
Jun21 <- subset(Jun21, nchar(as.character(ride_id)) == 16)
```

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

```
Jun21 <- subset (Jun21, ride_length > 59)
```

STEP FOUR: ANALYZE DATA

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

```
mean(Jun21$ride_length)
```

```
## [1] 1600.688
```

```
median(Jun21$ride_length)
```

```
## [1] 835
```

```
max(Jun21$ride_length)
```

```
## [1] 3356649
```

```
min(Jun21$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(Jun21$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      60     481     835    1601    1488 3356649
```

Compare the members and casual users

```
aggregate(Jun21$ride_length ~ Jun21$member_casual, FUN = mean)
```

```
##   Jun21$member_casual Jun21$ride_length
## 1              casual         2337.1520
## 2              member          861.7032
```

```
aggregate(Jun21$ride_length ~ Jun21$member_casual, FUN = median)
```

```
##   Jun21$member_casual Jun21$ride_length
## 1              casual              1097
## 2              member               649
```

```
aggregate(Jun21$ride_length ~ Jun21$member_casual, FUN = max)
```

```
##   Jun21$member_casual Jun21$ride_length
## 1              casual           3356649
## 2              member             89738
```

```
aggregate(Jun21$ride_length ~ Jun21$member_casual, FUN = min)
```

```
##   Jun21$member_casual Jun21$ride_length
## 1              casual                60
## 2              member                60
```

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

```
aggregate(Jun21$ride_length ~ Jun21$member_casual + Jun21$day_of_week, FUN = mean)
```

```
##    Jun21$member_casual Jun21$day_of_week Jun21$ride_length
## 1               casual            Friday         2287.1129
## 2               member            Friday          849.2956
## 3               casual            Monday         1903.2484
## 4               member            Monday          798.7826
## 5               casual          Saturday         2615.5646
## 6               member          Saturday          944.1037
## 7               casual            Sunday         2608.5238
## 8               member            Sunday          982.1960
## 9               casual          Thursday         2204.6371
## 10              member          Thursday          827.1367
## 11              casual           Tuesday         2097.5352
## 12              member           Tuesday          833.6984
## 13              casual         Wednesday         2195.1779
## 14              member         Wednesday          825.7246
```

Sort the days of the week in order.

```
Jun21$day_of_week <- ordered(Jun21$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(Jun21$ride_length ~ Jun21$member_casual + Jun21$day_of_week, FUN = mean)

head(x)
```

```
##   Jun21$member_casual Jun21$day_of_week Jun21$ride_length
## 1              casual            Sunday         2608.5238
## 2              member            Sunday          982.1960
## 3              casual            Monday         1903.2484
## 4              member            Monday          798.7826
## 5              casual           Tuesday         2097.5352
## 6              member           Tuesday          833.6984
```

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

```
y <- Jun21 %>%
  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           58111            2609.
## 2 casual              2           27787            1903.
## 3 casual              3           38195            2098.
## 4 casual              4           38685            2195.
## 5 casual              5           32803            2205.
## 6 casual              6           42646            2287.
```

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

```
table(Jun21$member_casual)
```

```
## 
## casual member 
## 300769 299743
```

```
table(Jun21$rideable_type)
```

```
## 
##  classic_bike   docked_bike electric_bike 
##        427723         51377        121412
```

```
table(Jun21$day_of_week)
```

```
## 
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday 
##     95575     64635     89863     93418     74120     82251    100650
```

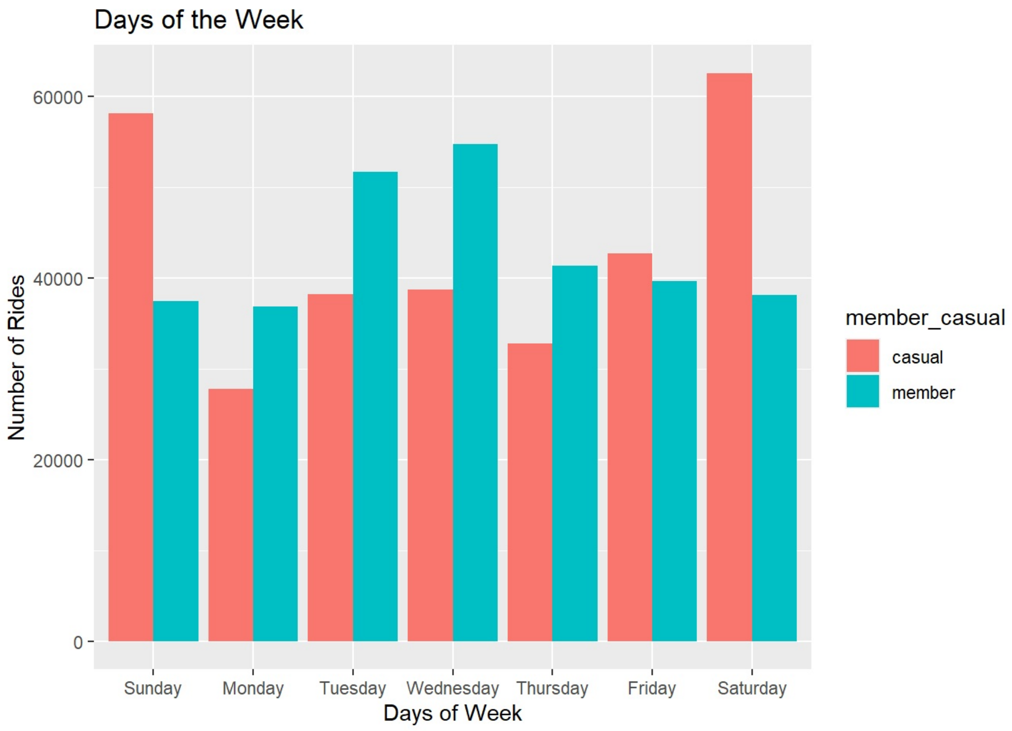
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

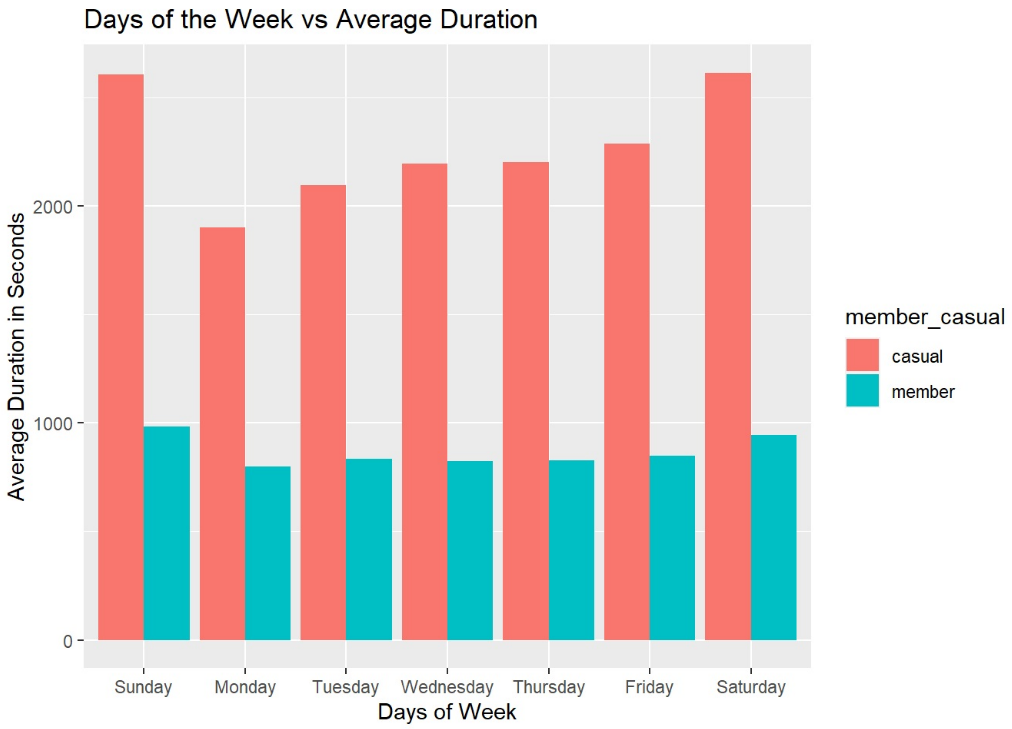
Plot the number of rides by user type during the week.

```
Jun21 %>%
  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.

```
Jun21 %>%
  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(Jun21$day_of_week,Jun21$member_casual))
```

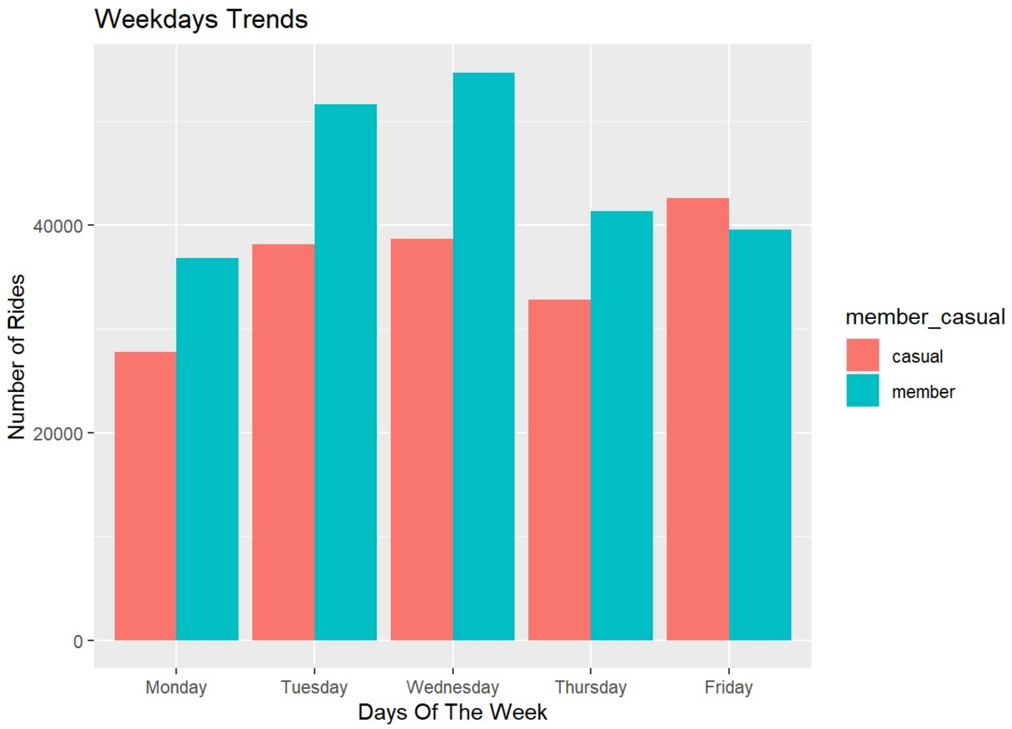
Rename columns

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

```
##   day_of_week member_casual  Freq
## 1      Sunday        casual 58111
## 2      Monday        casual 27787
## 3     Tuesday        casual 38195
## 4   Wednesday        casual 38685
## 5    Thursday        casual 32803
## 6      Friday        casual 42646
```

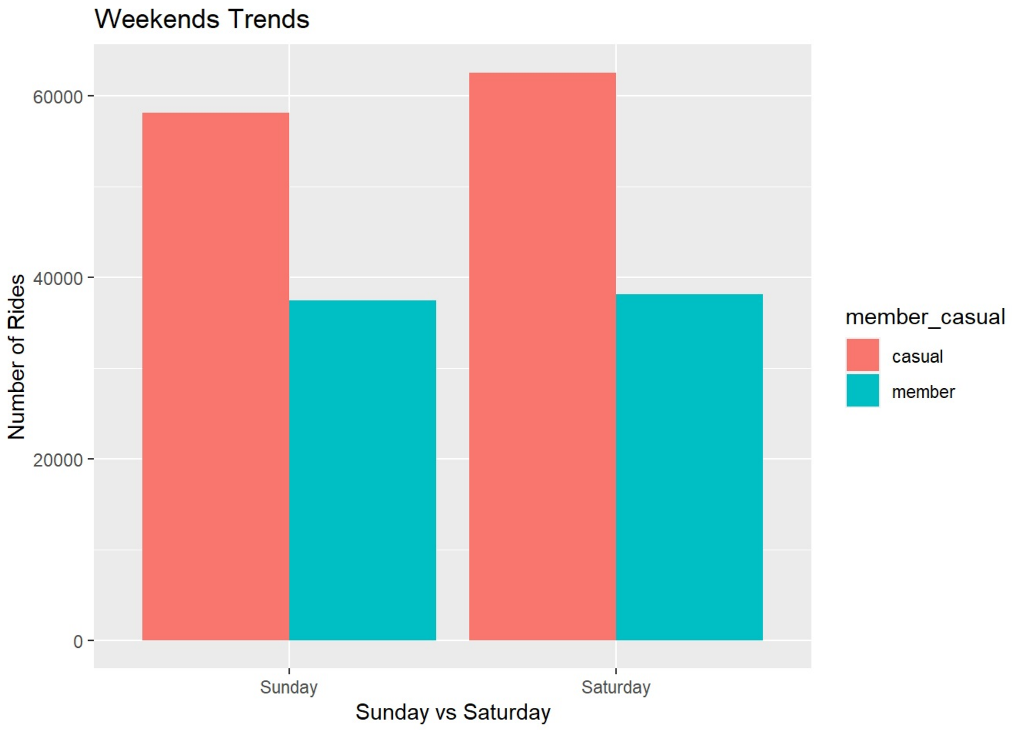
Weekday trends (Monday through Friday).

```
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(Jun21$rideable_type,Jun21$member_casual))
```

Rename columns.

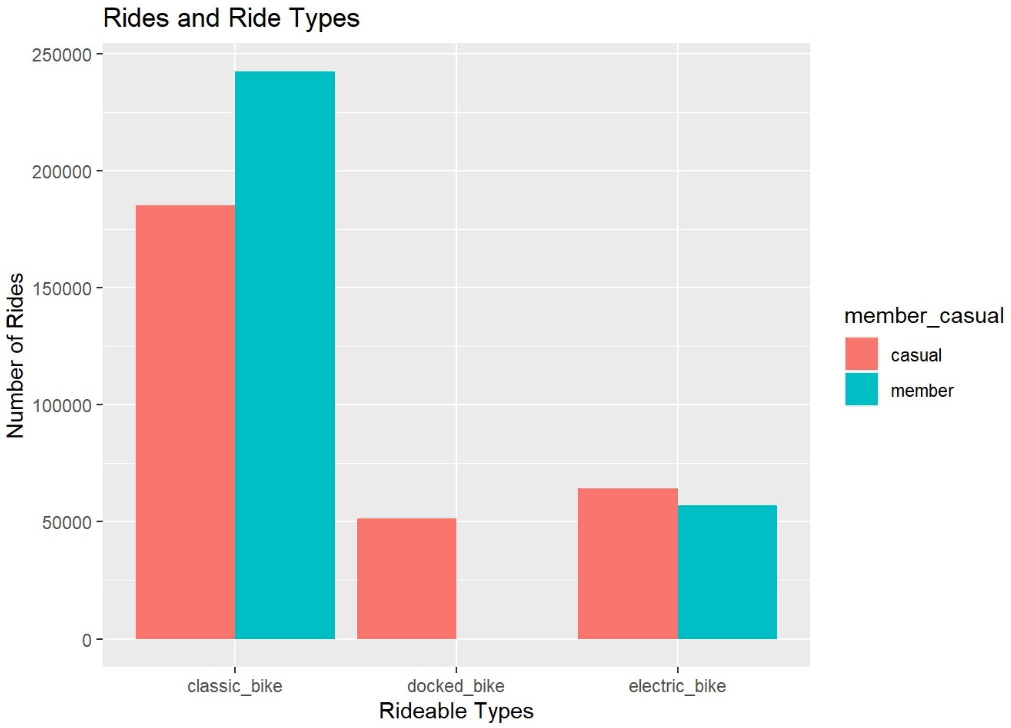
```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)

head(rt)
```

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual 185124
## 2   docked_bike        casual  51377
## 3 electric_bike        casual  64268
## 4  classic_bike        member 242599
## 5   docked_bike        member      0
## 6 electric_bike        member  57144
```

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(Jun21, “Jun21.csv”)