Cyclistic Case Study Q2_2021

Hezar K

2022-11-29

This is an analysis for Cyclistic Case Study for Google Data Analytics Course. This is an analysis for 2021's second quarter.

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.

```
Apr21 <- read_csv("202104-divvy-tripdata.csv")
May21 <- read_csv("202105-divvy-tripdata.csv")
Jun21 <- read_csv("202106-divvy-tripdata.csv")
```

STEP TWO: EXAMINE THE DATA

Examine the dataframe for an overview of the data. Review column names, **colnames()**. Then, we need to combine all data one dataframe. Then we examine dataframes to find dimensions, **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()**.

```
colnames(Apr21)
```

```
##  [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"
```

```
colnames(May21)
```

```
##  [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"
```

```
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"
```

Since all column names are the same. We can combine the data for each month into quarters.

```
q2_2021 <- bind_rows(Apr21, May21, Jun21)
```

```
View(q2_2021)
```

```
nrow(q2_2021)
```

```
## [1] 1598458
```

```
dim(q2_2021)
```

```
## [1] 1598458      13
```

```
head(q2_2021)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 6C992BD37A98A… classi… 2021-04-12 18:25:36 2021-04-12 18:56:55 State … TA1307…
## 2 1E0145613A209… docked… 2021-04-27 17:27:11 2021-04-27 18:31:29 Dorche… KA1503…
## 3 E498E15508A80… docked… 2021-04-03 12:42:45 2021-04-07 11:40:24 Loomis… 20121
## 4 1887262AD101C… classi… 2021-04-17 09:17:42 2021-04-17 09:42:48 Honore… TA1305…
## 5 C123548CAB2A3… docked… 2021-04-03 12:42:25 2021-04-03 14:13:42 Loomis… 20121
## 6 097E76F3651B1… classi… 2021-04-25 18:43:18 2021-04-25 18:43:59 Clinto… 15542
## # … 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(q2_2021)
```

```
## # 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(q2_2021)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:1598458     Length:1598458     Min.   :2021-04-01 00:03:18.00  
##  Class :character   Class :character   1st Qu.:2021-05-04 14:09:53.00  
##  Mode  :character   Mode  :character   Median :2021-05-29 06:21:16.50  
##                                        Mean   :2021-05-24 03:33:00.96  
##                                        3rd Qu.:2021-06-13 15:32:47.00  
##                                        Max.   :2021-06-30 23:59:59.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2021-04-01 00:14:29.00   Length:1598458     Length:1598458    
##  1st Qu.:2021-05-04 14:27:43.50   Class :character   Class :character  
##  Median :2021-05-29 06:58:11.50   Mode  :character   Mode  :character  
##  Mean   :2021-05-24 03:58:40.43                                        
##  3rd Qu.:2021-06-13 16:02:30.00                                        
##  Max.   :2021-07-13 22:51:35.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:1598458     Length:1598458     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:1598458    
##  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.   :42.15   Max.   :-87.49                     
##  NA's   :1436    NA's   :1436
```

```
str(q2_2021)
```

```
## spc_tbl_ [1,598,458 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1598458] "6C992BD37A98A63F" "1E0145613A209000" "E498E15508A80BAD" "1887262AD101C
604" ...
##  $ rideable_type     : chr [1:1598458] "classic_bike" "docked_bike" "docked_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:1598458], format: "2021-04-12 18:25:36" "2021-04-27 17:27:11" ...
##  $ ended_at          : POSIXct[1:1598458], format: "2021-04-12 18:56:55" "2021-04-27 18:31:29" ...
##  $ start_station_name: chr [1:1598458] "State St & Pearson St" "Dorchester Ave & 49th St" "Loomis Blvd & 84th
St" "Honore St & Division St" ...
##  $ start_station_id  : chr [1:1598458] "TA1307000061" "KA1503000069" "20121" "TA1305000034" ...
##  $ end_station_name  : chr [1:1598458] "Southport Ave & Waveland Ave" "Dorchester Ave & 49th St" "Loomis Blvd
& 84th St" "Southport Ave & Waveland Ave" ...
##  $ end_station_id    : chr [1:1598458] "13235" "KA1503000069" "20121" "13235" ...
##  $ start_lat         : num [1:1598458] 41.9 41.8 41.7 41.9 41.7 ...
##  $ start_lng         : num [1:1598458] -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:1598458] 41.9 41.8 41.7 41.9 41.7 ...
##  $ end_lng           : num [1:1598458] -87.7 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:1598458] "member" "casual" "casual" "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.

```
q2_2021$date <- as.Date(q2_2021$started_at)
q2_2021$month <- format(as.Date(q2_2021$date), "%m")
q2_2021$month <- month.name[as.numeric(q2_2021$month)]
q2_2021$day <- format(as.Date(q2_2021$date), "%d")
q2_2021$year <- format(as.Date(q2_2021$date), "%Y")
q2_2021$day_of_week <- format(as.Date(q2_2021$date), "%A")
q2_2021$ride_length <- difftime(q2_2021$ended_at,q2_2021$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(q2_2021$ride_length)
```

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

Recheck *ride_length* data type.

```
q2_2021$ride_length <- as.numeric(as.character(q2_2021$ride_length))
is.numeric(q2_2021$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
q2_2021 <- na.omit(q2_2021)
```

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.

```
q2_2021 <- subset(q2_2021, nchar(as.character(ride_id)) == 16)
```

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

```
q2_2021 <- subset (q2_2021, ride_length > 59)
```

STEP FOUR: ANALYZE DATA

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

```
mean(q2_2021$ride_length)
```

```
## [1] 1572.827
```

```
median(q2_2021$ride_length)
```

```
## [1] 830
```

```
max(q2_2021$ride_length)
```

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

```
min(q2_2021$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(q2_2021$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      60     472     830    1573    1508 3356649
```

Compare the members and casual users

```
aggregate(q2_2021$ride_length ~ q2_2021$member_casual, FUN = mean)
```

```
##   q2_2021$member_casual q2_2021$ride_length
## 1                casual           2357.0740
## 2                member            867.1988
```

```
aggregate(q2_2021$ride_length ~ q2_2021$member_casual, FUN = median)
```

```
##   q2_2021$member_casual q2_2021$ride_length
## 1                casual                1135
## 2                member                 645
```

```
aggregate(q2_2021$ride_length ~ q2_2021$member_casual, FUN = max)
```

```
##   q2_2021$member_casual q2_2021$ride_length
## 1                casual             3356649
## 2                member               89738
```

```
aggregate(q2_2021$ride_length ~ q2_2021$member_casual, FUN = min)
```

```
##   q2_2021$member_casual q2_2021$ride_length
## 1                casual                  60
## 2                member                  60
```

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

```
aggregate(q2_2021$ride_length ~ q2_2021$member_casual + q2_2021$day_of_week, FUN = mean)
```

```
##    q2_2021$member_casual q2_2021$day_of_week q2_2021$ride_length
## 1                 casual              Friday           2321.7558
## 2                 member              Friday            836.3391
## 3                 casual              Monday           2155.3968
## 4                 member              Monday            829.7196
## 5                 casual            Saturday           2471.2512
## 6                 member            Saturday            964.9821
## 7                 casual              Sunday           2728.1021
## 8                 member              Sunday           1000.3970
## 9                 casual            Thursday           2026.4018
## 10                member            Thursday            808.7309
## 11                casual             Tuesday           2147.0841
## 12                member             Tuesday            825.5787
## 13                casual           Wednesday           2154.9303
## 14                member           Wednesday            815.9145
```

Sort the days of the week in order.

```
q2_2021$day_of_week <- ordered(q2_2021$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursda
y", "Friday", "Saturday"))
```

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

```
x <- aggregate(q2_2021$ride_length ~ q2_2021$member_casual + q2_2021$day_of_week, FUN = mean)

head(x)
```

```
##   q2_2021$member_casual q2_2021$day_of_week q2_2021$ride_length
## 1                casual              Sunday           2728.1021
## 2                member              Sunday           1000.3970
## 3                casual              Monday           2155.3968
## 4                member              Monday            829.7196
## 5                casual             Tuesday           2147.0841
## 6                member             Tuesday            825.5787
```

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

```
y <- q2_2021 %>%
  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          134329            2728.
## 2 casual              2           70422            2155.
## 3 casual              3           70799            2147.
## 4 casual              4           67674            2155.
## 5 casual              5           61541            2026.
## 6 casual              6           86981            2322.
```

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

```
table(q2_2021$member_casual)
```

```
##
## casual member
## 634783 705509
```

```
table(q2_2021$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##        943278        118998        278016
```

```
table(q2_2021$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##    229175    166600    178678    177880    157230    187401    243328
```

```
table(q2_2021$month)
```

```
##
##  April   June    May
## 294623 600512 445157
```

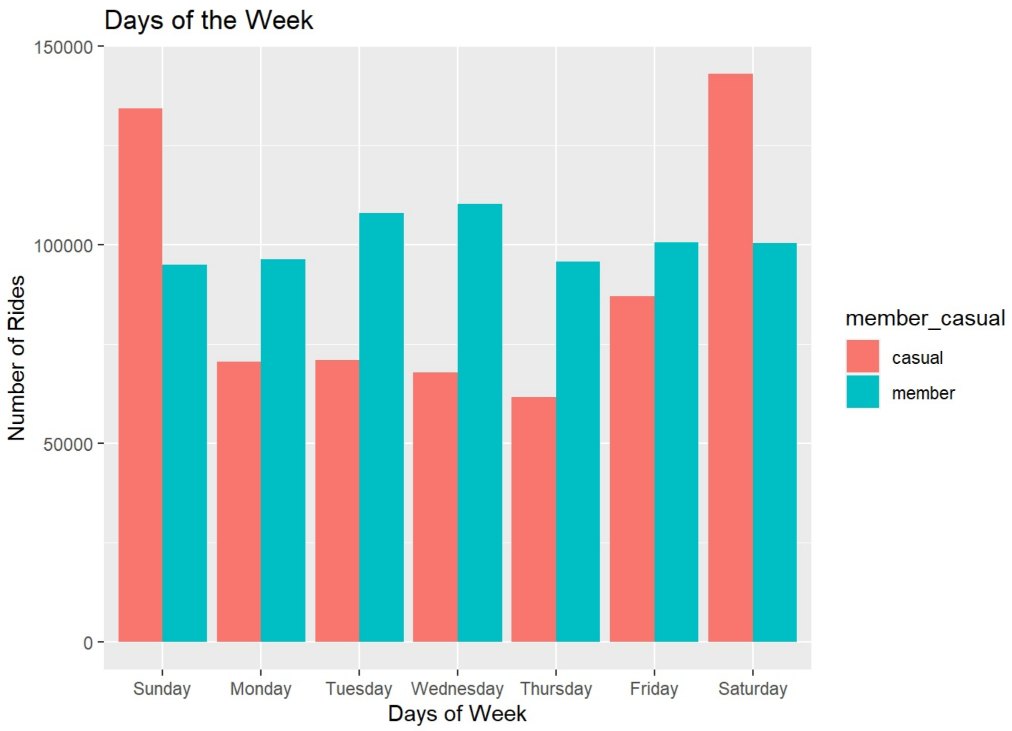
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

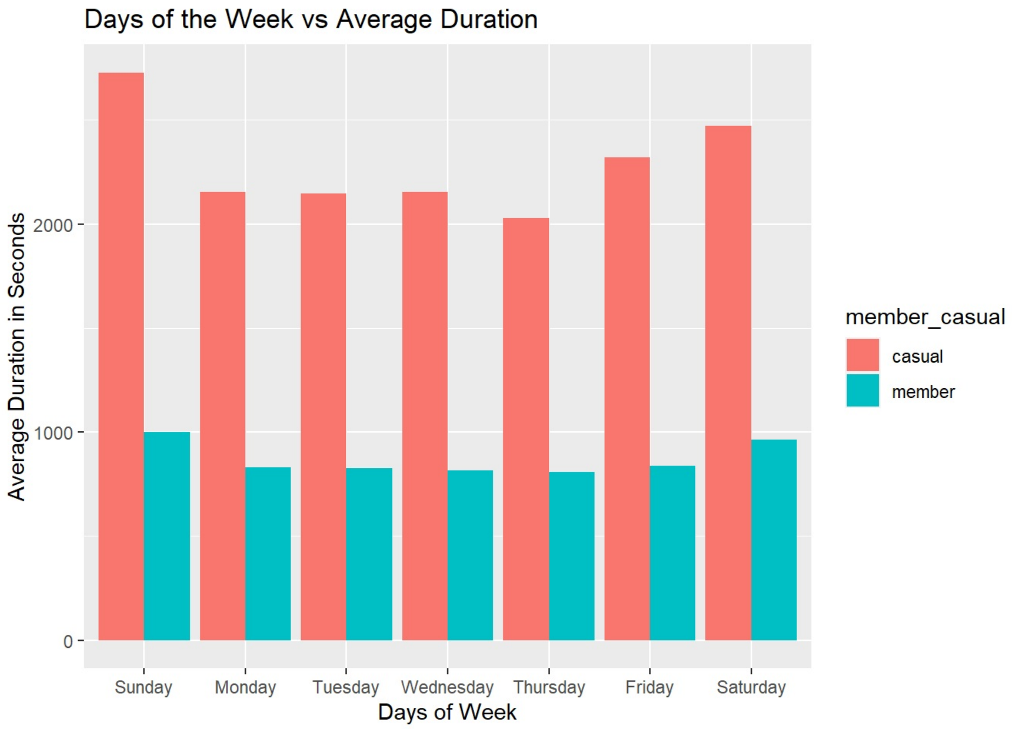
Plot the number of rides by user type during the week.

```
q2_2021 %>%
  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.

```
q2_2021 %>%
  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(q2_2021$day_of_week,q2_2021$member_casual))
```

Rename columns

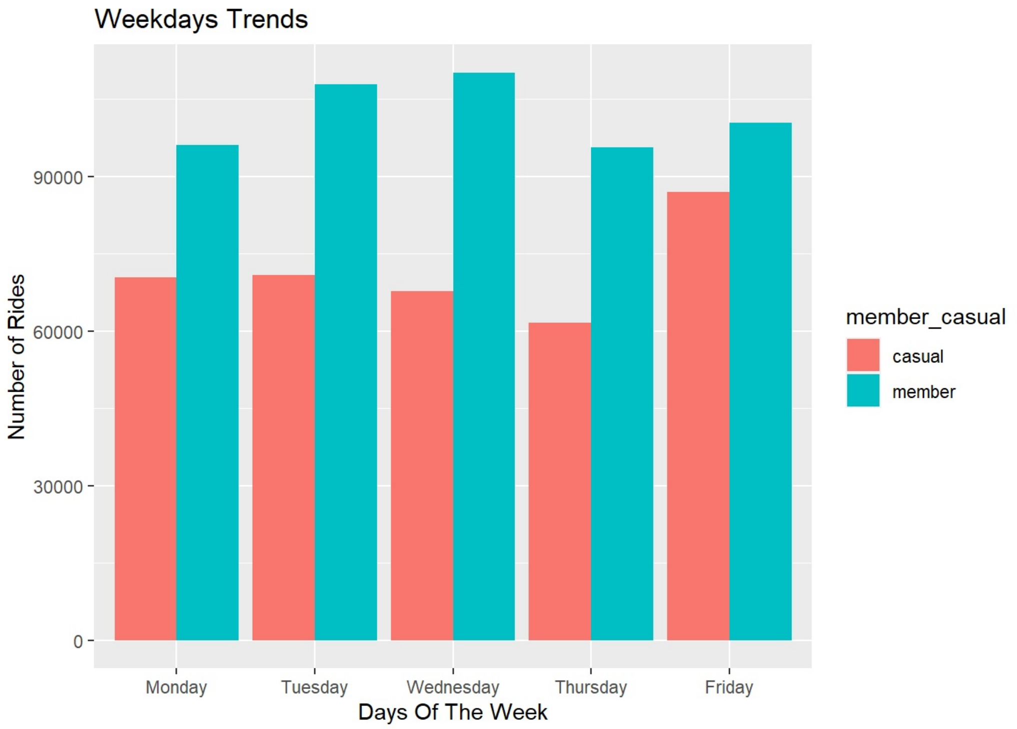
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)

head(mc)
```

```
##   day_of_week member_casual   Freq
## 1      Sunday        casual 134329
## 2      Monday        casual  70422
## 3     Tuesday        casual  70799
## 4   Wednesday        casual  67674
## 5    Thursday        casual  61541
## 6      Friday        casual  86981
```

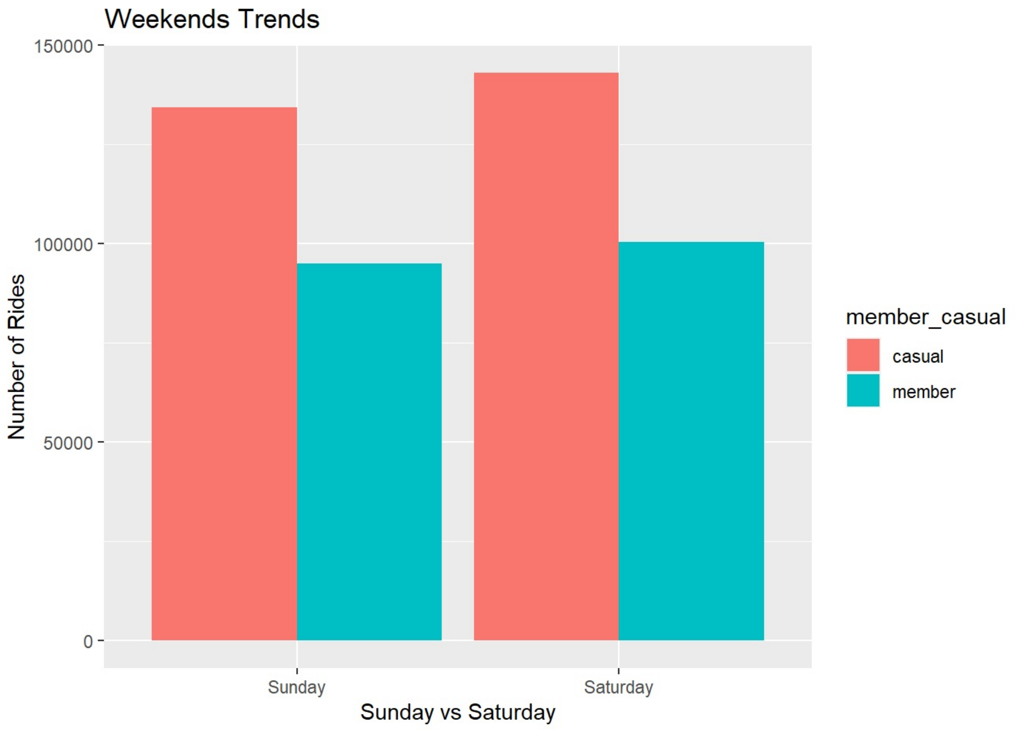
Weekday trends (Monday through Friday).

```
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(q2_2021$rideable_type,q2_2021$member_casual))
```

Rename columns.

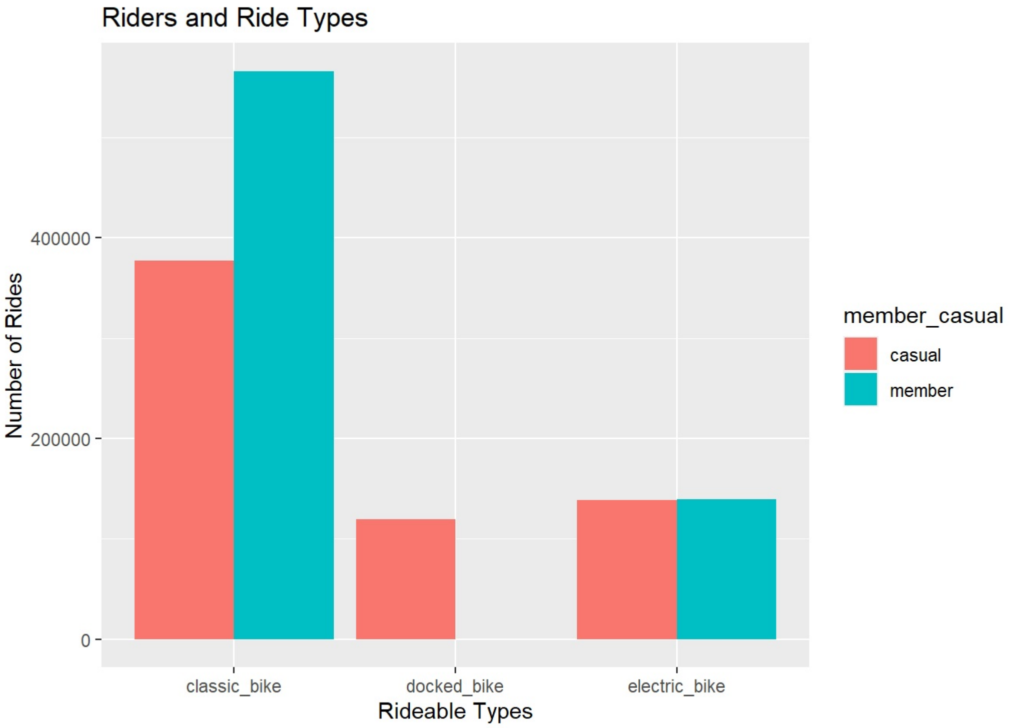
```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)

head(rt)
```

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual 377037
## 2   docked_bike        casual 118998
## 3 electric_bike        casual 138748
## 4  classic_bike        member 566241
## 5   docked_bike        member      0
## 6 electric_bike        member 139268
```

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 = "Riders and Ride Types",
       x= "Rideable Types",
       y = "Number of Rides")
```



Create vector of month names for Q2 2021

```
q2_months <- c("April", "May", "June")
```

Subset month.name to include only Q2 2021 months

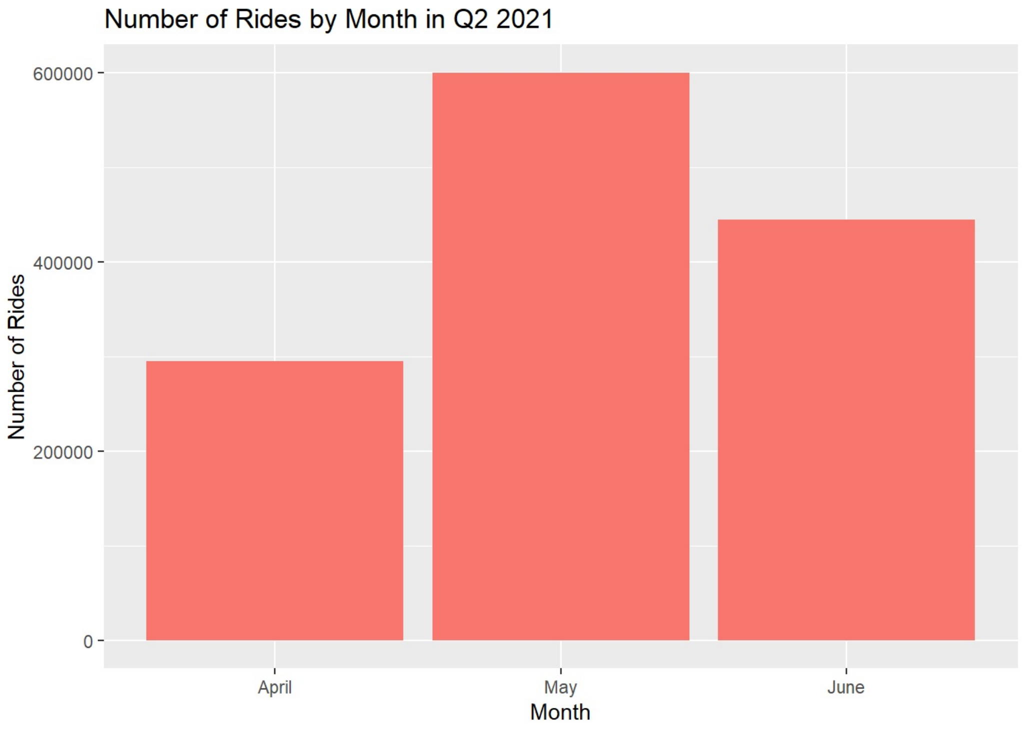
```
q2_month_names <- month.name[match(q2_months, month.name)]
```

Create trips_by_month dataframe with only Q2 2021 months

```
trips_by_month <- data.frame(month = q2_month_names, count = table(q2_2021$month))
```

Set the levels of the month variable in the trips_by_month dataframe

```
trips_by_month$month <- factor(trips_by_month$month, levels = c("April", "May", "June"))
ggplot(trips_by_month, aes(x = month, y = count.Freq)) +
geom_bar(stat = "identity", fill = "#F8766D") +
  labs(x = "Month", y = "Number of Rides", title = "Number of Rides by Month in Q2 2021")
```



STEP SIX: EXPORT ANALYZED DATA

Save the analyzed data as a new file. fwrite(q2_2021, “q2_2021.csv”)