Analysis of Divvy Bike Sharing Data

Thanzin Naing

2022-08-09

This is a report on my project for the Google Analytics Course. This is an analysis of bike sharing data that has been collected every single month by a bike sharing company. My analysis includes one years worth of data from June 2021 to May 2022. First we must set up our R packages.

```
library(tidyverse)
```

Helps wrangle data

```
library(lubridate)
```

Helps wrangle date attributes

```
library(ggplot2)
```

Helps visualize data

```
library(scales)
```

Changes unit format

Step 1: Collect Data

Load Dataset and convert to dataframes

```
setwd("C:/Users/An-94/Desktop/google_analytics/proj_data/Divvy_Trips_2021to2022") #sets your working di

may_2022 <- read_csv("202205-divvy-tripdata-edit.csv")
april_2022 <- read_csv("202204-divvy-tripdata-edit.csv")
march_2022 <- read_csv("202203-divvy-tripdata-edit.csv")
feb_2022 <- read_csv("202202-divvy-tripdata-edit.csv")
jan_2022 <- read_csv("202201-divvy-tripdata-edit.csv")
dec_2021 <- read_csv("202112-divvy-tripdata-edit.csv")
nov_2021 <- read_csv("202111-divvy-tripdata-copy.csv")
oct_2021 <- read_csv("202110-divvy-tripdata-edit.csv")
sep_2021 <- read_csv("202109-divvy-tripdata-edit.csv")
aug_2021 <- read_csv("202108-divvy-tripdata-edit.csv")
july_2021 <- read_csv("202107-divvy-tripdata-edit.csv")
june_2021 <- read_csv("202106-divvy-tripdata-edit.csv")
```

Step 2: Wrangle the data and Combine the data frames into one

Before I combine data frames I must make sure that all of their fields are the same

First I will check the fields

```
colnames(june_2021)
```

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

```
colnames(july_2021)
```

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

```
colnames(aug_2021)
```

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

```
colnames(sep_2021)
```

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

```
colnames(oct_2021)
```

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

```
colnames(nov_2021)
```

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

```
colnames(dec_2021)
```

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

```
colnames(jan_2022)
```

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

```
colnames(feb_2022)
```

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

```
colnames(march_2022)
```

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

```
colnames(april_2022)
```

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

```
colnames(may_2022)
```

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

Now that I know all fields are the same. I will also rename columns to make them easier to understand

```
june_2021 <- rename(june_2021
                    ,trip_id = ride_id
                    ,biketype  = rideable_type
                    ,start_time  = started_at
                    ,end_time = ended_at
                    ,from_station_name = start_station_name
                    ,from_station_id  = start_station_id
                    ,to_station_name  = end_station_name
                    ,to_station_id  = end_station_id
                    ,usertype = member_casual)

july_2021 <- rename(july_2021
                    ,trip_id = ride_id
                    ,biketype  = rideable_type
                    ,start_time  = started_at
                    ,end_time = ended_at
                    ,from_station_name = start_station_name
                    ,from_station_id  = start_station_id
                    ,to_station_name  = end_station_name
                    ,to_station_id  = end_station_id
                    ,usertype = member_casual)
aug_2021 <- rename(aug_2021
                   ,trip_id = ride_id
                   ,biketype  = rideable_type
                   ,start_time  = started_at
                   ,end_time = ended_at
                   ,from_station_name = start_station_name
                   ,from_station_id  = start_station_id
                   ,to_station_name  = end_station_name
                   ,to_station_id  = end_station_id
                   ,usertype = member_casual)

sep_2021 <- rename(sep_2021
                   ,trip_id = ride_id
                   ,biketype  = rideable_type
                   ,start_time  = started_at
                   ,end_time = ended_at
                   ,from_station_name = start_station_name
                   ,from_station_id  = start_station_id
                   ,to_station_name  = end_station_name
                   ,to_station_id  = end_station_id
                   ,usertype = member_casual)

oct_2021 <- rename(oct_2021
                   ,trip_id = ride_id
                   ,biketype  = rideable_type
                   ,start_time  = started_at
                   ,end_time = ended_at
                   ,from_station_name = start_station_name
                   ,from_station_id  = start_station_id
                   ,to_station_name  = end_station_name
                   ,to_station_id  = end_station_id
                   ,usertype = member_casual)

nov_2021 <- rename(nov_2021
                   ,trip_id = ride_id
                   ,biketype  = rideable_type
                   ,start_time  = started_at
                   ,end_time = ended_at
                   ,from_station_name = start_station_name
                   ,from_station_id  = start_station_id
                   ,to_station_name  = end_station_name
                   ,to_station_id  = end_station_id
                   ,usertype = member_casual)

dec_2021 <- rename(dec_2021
                   ,trip_id = ride_id
                   ,biketype  = rideable_type
                   ,start_time  = started_at
                   ,end_time = ended_at
                   ,from_station_name = start_station_name
                   ,from_station_id  = start_station_id
                   ,to_station_name  = end_station_name
                   ,to_station_id  = end_station_id
                   ,usertype = member_casual)


jan_2022 <- rename(jan_2022
                   ,trip_id = ride_id
                   ,biketype  = rideable_type
                   ,start_time  = started_at
                   ,end_time = ended_at
                   ,from_station_name = start_station_name
                   ,from_station_id  = start_station_id
                   ,to_station_name  = end_station_name
                   ,to_station_id  = end_station_id
                   ,usertype = member_casual)
```

```
feb_2022 <- rename(feb_2022
                   ,trip_id = ride_id
                   ,biketype  = rideable_type
                   ,start_time  = started_at
                   ,end_time = ended_at
                   ,from_station_name = start_station_name
                   ,from_station_id  = start_station_id
                   ,to_station_name  = end_station_name
                   ,to_station_id  = end_station_id
                   ,usertype = member_casual)


march_2022 <- rename(march_2022
                   ,trip_id = ride_id
                   ,biketype  = rideable_type
                   ,start_time  = started_at
                   ,end_time = ended_at
                   ,from_station_name = start_station_name
                   ,from_station_id  = start_station_id
                   ,to_station_name  = end_station_name
                   ,to_station_id  = end_station_id
                   ,usertype = member_casual)

april_2022 <- rename(april_2022
                     ,trip_id = ride_id
                     ,biketype  = rideable_type
                     ,start_time  = started_at
                     ,end_time = ended_at
                     ,from_station_name = start_station_name
                     ,from_station_id  = start_station_id
                     ,to_station_name  = end_station_name
                     ,to_station_id  = end_station_id
                     ,usertype = member_casual)


may_2022 <- rename(may_2022
                     ,trip_id = ride_id
                     ,biketype  = rideable_type
                     ,start_time  = started_at
                     ,end_time = ended_at
                     ,from_station_name = start_station_name
                     ,from_station_id  = start_station_id
                     ,to_station_name  = end_station_name
                     ,to_station_id  = end_station_id
                     ,usertype = member_casual)
```

I will then check the data types for each column and see if there are any dissimilarities. Because dissimilarities in data type will also prevent a merge.

```
str(june_2021)
```

```
## spec_tbl_df [721,787 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ trip_id          : chr [1:721787] "45A37F50CBEA6B1B" "8BDDD73BCD395A3C" "6C16A5E7E6A957EA" "EA7283
##  $ biketype         : chr [1:721787] "docked_bike" "docked_bike" "docked_bike" "classic_bike" ...
##  $ start_time       : chr [1:721787] "6/5/2021 7:26" "6/5/2021 7:27" "6/5/2021 7:27" "6/1/2021 17:42
##  $ end_time         : chr [1:721787] "6/6/2021 7:24" "6/6/2021 7:24" "6/6/2021 7:24" "6/2/2021 17:38
##  $ ride_length      : 'hms' num [1:721787] 23:58:00 23:57:00 23:57:00 23:56:00 ...
##   ..- attr(*, "units")= chr "secs"
##  $ day_of_the_week  : num [1:721787] 7 7 7 3 5 7 6 1 7 1 ...
##  $ from_station_name: chr [1:721787] "Yates Blvd & 75th St" "Yates Blvd & 75th St" "Yates Blvd & 75th
##  $ from_station_id  : chr [1:721787] "KA1503000024" "KA1503000024" "KA1503000024" "13008" ...
##  $ to_station_name  : chr [1:721787] "Yates Blvd & 75th St" "Yates Blvd & 75th St" "Yates Blvd & 75th
##  $ to_station_id    : chr [1:721787] "KA1503000024" "KA1503000024" "KA1503000024" "TA1305000005" ...
##  $ start_lat        : num [1:721787] 41.8 41.8 41.8 41.9 41.9 ...
##  $ start_lng        : num [1:721787] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat          : num [1:721787] 41.8 41.8 41.8 41.9 41.9 ...
##  $ end_lng          : num [1:721787] -87.6 -87.6 -87.6 -87.6 -87.7 ...
##  $ usertype         : chr [1:721787] "casual" "casual" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   ride_length = col_time(format = ""),
##   ..   day_of_the_week = col_double(),
##   ..   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>
```

```
str(july_2021)
str(aug_2021)
str(sep_2021)
str(oct_2021)
str(nov_2021)
str(dec_2021)
str(jan_2022)
str(feb_2022)
str(march_2022)
str(april_2022)
str(may_2022)
```

I've decided to only show the output for the June 2021 data because showing every month would take up too much space. To summarize, all the data frames have the same data type for each column.

Lets stack each month's data frame into a single data frame

```
all_trips <- bind_rows(june_2021,
                       july_2021,
                       aug_2021,
                       sep_2021,
                       oct_2021,
                       nov_2021,
                       dec_2021,
                       jan_2022,
                       feb_2022,
                       march_2022,
                       april_2022,
                       may_2022
                       )
```

Second, let us remove fields in the data frame that is not relevant to my analysis like coordinates(latitude, longitude)

```
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng))
```

STEP 3: Clean and Add data(data preparation)

Lets inspect the new table that has been created

```
colnames(all_trips)  #List of column names
```

```
##  [1] "trip_id"           "biketype"          "start_time"
##  [4] "end_time"          "ride_length"       "day_of_the_week"
##  [7] "from_station_name" "from_station_id"   "to_station_name"
## [10] "to_station_id"     "usertype"
```

```
dim(all_trips)  #Dimensions of the data frame?
```

```
## [1] 5802042      11
```

```
head(all_trips)  #See the first 6 rows of data frame
```

```
## # A tibble: 6 x 11
##   trip_id          biketype     start_time  end_time ride_length day_of_the_week
##   <chr>            <chr>        <chr>       <chr>    <time>                <dbl>
## 1 45A37F50CBEA6B1B docked_bike  6/5/2021 7~ 6/6/202~ 23:58                     7
## 2 8BDDD73BCD395A3C docked_bike  6/5/2021 7~ 6/6/202~ 23:57                     7
## 3 6C16A5E7E6A957EA docked_bike  6/5/2021 7~ 6/6/202~ 23:57                     7
## 4 EA728377BBF5C2A5 classic_bike 6/1/2021 1~ 6/2/202~ 23:56                     3
## 5 1EBE31B591E555EA classic_bike 6/24/2021 ~ 6/25/20~ 23:56                     5
## 6 CB32841B69D1778B classic_bike 6/26/2021 ~ 6/27/20~ 23:54                     7
## # ... with 5 more variables: from_station_name <chr>, from_station_id <chr>,
## #   to_station_name <chr>, to_station_id <chr>, usertype <chr>
```

```
tail(all_trips) #See the last 6 rows of data frame
```

```
## # A tibble: 6 x 11
##   trip_id          biketype      start_time end_time ride_length day_of_the_week
##   <chr>            <chr>         <chr>      <chr>    <time>                <dbl>
## 1 B8156ADA319B384E electric_bike 5/31/2022~ 5/31/20~ 01’00"                    3
## 2 F0249C4DA829A7FC classic_bike  5/31/2022~ 5/31/20~ 01’00"                    3
## 3 522078935568EE07 classic_bike  5/31/2022~ 5/31/20~ 01’00"                    3
## 4 6FBACB7E74D46A1A electric_bike 5/31/2022~ 5/31/20~ 01’00"                    3
## 5 1BBDB13D9BCB0192 electric_bike 5/31/2022~ 5/31/20~ 01’00"                    3
## 6 A904966008DE7AF1 electric_bike 5/31/2022~ 6/1/202~ 01’00"                    3
## # ... with 5 more variables: from_station_name <chr>, from_station_id <chr>,
## #   to_station_name <chr>, to_station_id <chr>, usertype <chr>
```

```
str(all_trips)  #See list of columns and data types (numeric, character, etc)
```

```
## tibble [5,802,042 x 11] (S3: tbl_df/tbl/data.frame)
##  $ trip_id          : chr [1:5802042] "45A37F50CBEA6B1B" "8BDDD73BCD395A3C" "6C16A5E7E6A957EA" "EA72
##  $ biketype         : chr [1:5802042] "docked_bike" "docked_bike" "docked_bike" "classic_bike" ...
##  $ start_time       : chr [1:5802042] "6/5/2021 7:26" "6/5/2021 7:27" "6/5/2021 7:27" "6/1/2021 17:4
##  $ end_time         : chr [1:5802042] "6/6/2021 7:24" "6/6/2021 7:24" "6/6/2021 7:24" "6/2/2021 17:3
##  $ ride_length      : ’hms’ num [1:5802042] 23:58:00 23:57:00 23:57:00 23:56:00 ...
##   ..- attr(*, "units")= chr "secs"
##  $ day_of_the_week  : num [1:5802042] 7 7 7 3 5 7 6 1 7 1 ...
##  $ from_station_name: chr [1:5802042] "Yates Blvd & 75th St" "Yates Blvd & 75th St" "Yates Blvd & 75t
##  $ from_station_id  : chr [1:5802042] "KA1503000024" "KA1503000024" "KA1503000024" "13008" ...
##  $ to_station_name  : chr [1:5802042] "Yates Blvd & 75th St" "Yates Blvd & 75th St" "Yates Blvd & 75t
##  $ to_station_id    : chr [1:5802042] "KA1503000024" "KA1503000024" "KA1503000024" "TA1305000005" ..
##  $ usertype         : chr [1:5802042] "casual" "casual" "casual" "casual" ...
```

```
summary(all_trips)  #Statistical summary of data.
```

```
##    trip_id            biketype           start_time          end_time
##  Length:5802042     Length:5802042     Length:5802042     Length:5802042
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  ride_length       day_of_the_week from_station_name  from_station_id
##  Length:5802042    Min.   :1.00    Length:5802042     Length:5802042
##  Class1:hms        1st Qu.:2.00    Class :character   Class :character
##  Class2:difftime   Median :4.00    Mode  :character   Mode  :character
##  Mode  :numeric    Mean   :4.08
##                    3rd Qu.:6.00
##                    Max.   :7.00
##  to_station_name    to_station_id        usertype
##  Length:5802042     Length:5802042     Length:5802042
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

Here are a list of things we can add to the data frame and what I believe should be removed

- Add some additional columns of data - such as day, month, year - that provide additional opportunities to aggregate the data.
- Add a "ride_length" calculation to all_trips (in seconds) so I can aggregate data
- There are some rides where trip duration shows up as negative, including several hundred rides where Divvy took bikes out of circulation for Quality Control reasons. Delete these rides as they would only make the analysis less correct

Create the date field from the start time which is a datetime object The default format is yyyy-mm-dd and this is how it is displayed

```
all_trips$date <- as.Date(all_trips$start_time, "%m/%d/%Y")
```

```
all_trips$month <- format(as.Date(all_trips$date), "%m") #create separate month column(numeric)
all_trips$day <- format(as.Date(all_trips$date), "%d") #create day column
all_trips$year <- format(as.Date(all_trips$date), "%Y") #create separate year column
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A") #create day of the week column(character
```

I make sure that start time and end time are in datetime format

```
all_trips <-  all_trips %>%
  mutate(start_time = mdy_hms(start_time), end_time = mdy_hms(end_time))
```

I find the difference between start and stop times

```
all_trips$ride_length <- difftime(all_trips$end_time,all_trips$start_time)
```

Since both start and stop times are datetime then the difference between the two will also be datetime. However, I want to convert it to a numeric so I can run calculations.

```
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
```

Create another version of the data frame that eliminate negative ride lengths

```
all_trips_all <- all_trips[(all_trips$ride_length>0),]
```

STEP 4: CONDUCT ANALYSIS AND MAKE RECOMMENDATIONS

The mean average

```
mean(all_trips_all$ride_length)
```

```
## [1] 588.3055
```

The midpoint number in the ascending array of ride lengths; median average

```r
median(all_trips_all$ride_length)
```

```
## [1] 11
```

Total number of seconds of all rides

```r
sum(all_trips_all$ride_length)
```

```
## [1] 3411302619
```

General descriptive statistics

```r
summary(all_trips_all$ride_length) #general descriptive statistics
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
##       1.0       6.0      11.0     588.3      21.0 2073395.0
```

Some insights from the summary statistics:

- Notice that the mean ride length is a lot higher than the median ride length.
- 50% of this dataset contains values below 11 seconds but the average value is 588 seconds
- This suggests that some rides are a lot longer than 11 seconds

Lets compare the differences in ride duration between members and casual users

```r
aggregate(all_trips_all$ride_length ~ all_trips_all$usertype, FUN = mean)
```

```
##   all_trips_all$usertype all_trips_all$ride_length
## 1                 casual                  980.0365
## 2                 member                  284.3094
```

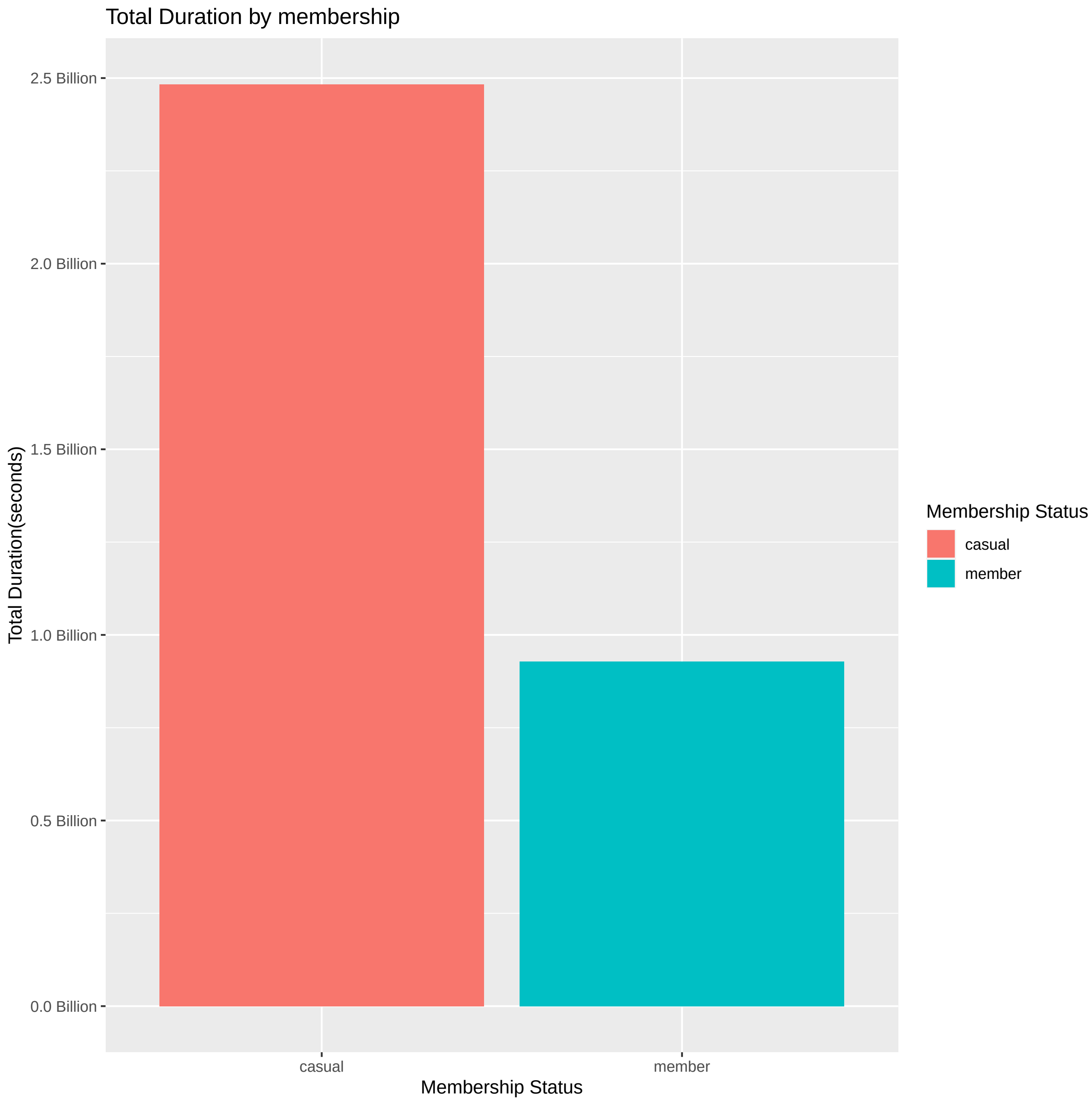
```r
aggregate(all_trips_all$ride_length ~ all_trips_all$usertype, FUN = median)
```

```
##   all_trips_all$usertype all_trips_all$ride_length
## 1                 casual                        15
## 2                 member                         9
```

Casuals on average had a higher ride duration

Which group has the highest ride time in total? First let us take a look at the total ride duration for casuals and members

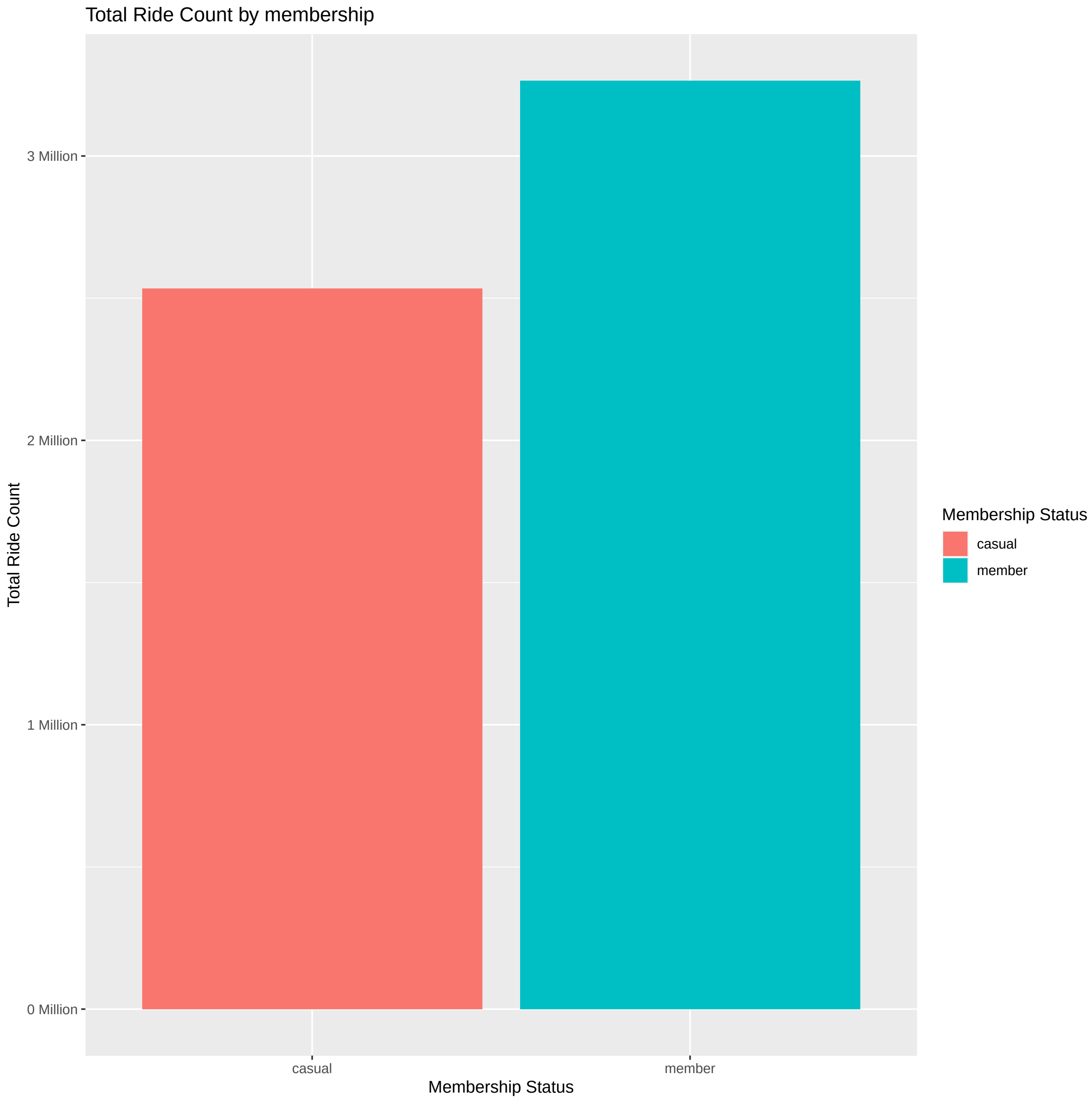
```r
all_trips_all %>%
  group_by(usertype) %>%
  summarise(total_ride_duration = sum(ride_length)) %>%
  ggplot(aes(x = usertype, y = total_ride_duration, fill = usertype)) +
  geom_bar(stat="identity") +
  labs(title="Total Duration by membership", y="Total Duration(seconds)", x="Membership Status") +
  guides(fill = guide_legend(title = 'Membership Status')) +
  scale_y_continuous(labels = unit_format(unit = "Billion", scale = 1e-09))
```



We see from this graph that casuals have the longest ride times in total and by a much greater amount

Which group uses the bikes the most often?

```
all_trips_all %>%
  group_by(usertype) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = usertype, y = number_of_rides, fill = usertype)) +
    geom_bar(stat="identity") +
    labs(title="Total Ride Count by membership", y="Total Ride Count", x="Membership Status") +
    guides(fill = guide_legend(title = 'Membership Status')) +
    scale_y_continuous(labels = unit_format(unit = "Million", scale = 1e-06))
```



Those with memberships use the bikes more often

On what days are the rides lengths longest and shortest for both members and casuals?

Here is a table

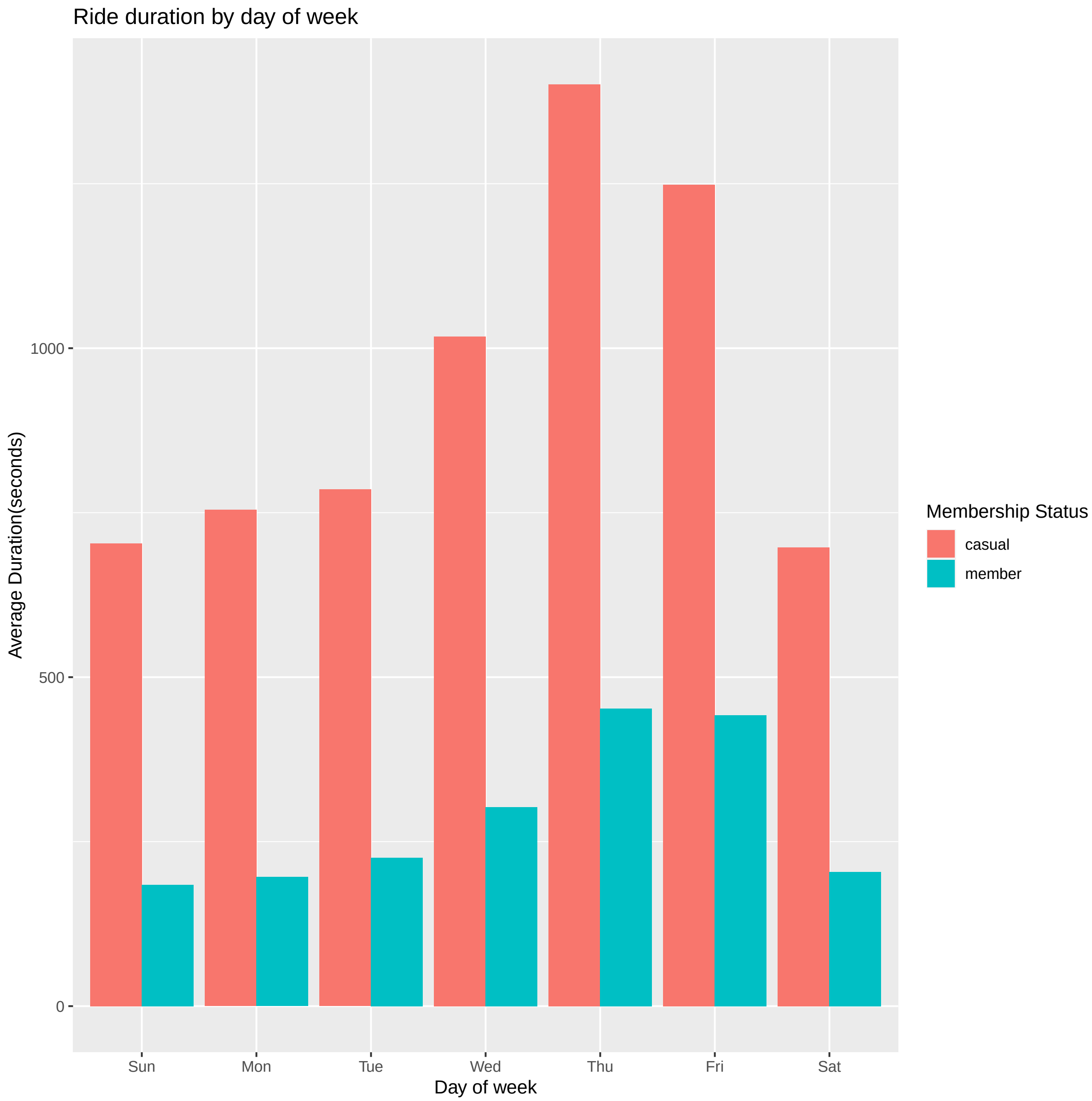
```
all_trips_all$day_of_week<- ordered(all_trips_all$day_of_week, levels=c("Sunday", "Monday", "Tuesday",

aggregate(all_trips_all$ride_length ~ all_trips_all$usertype + all_trips_all$day_of_week, FUN = mean)
```

```
##    all_trips_all$usertype all_trips_all$day_of_week all_trips_all$ride_length
## 1                  casual                    Sunday                  722.0695
## 2                  member                    Sunday                  228.3564
## 3                  casual                    Monday                  698.8537
## 4                  member                    Monday                  181.5481
## 5                    casual                    Tuesday                 745.3802
## 6                    member                    Tuesday                 192.6214
## 7                    casual                  Wednesday                 741.6891
## 8                    member                  Wednesday                 208.1308
## 9                    casual                   Thursday                 983.9302
## 10                   member                   Thursday                 264.8823
## 11                   casual                     Friday                1390.0873
## 12                   member                     Friday                 428.9706
## 13                   casual                   Saturday                1333.1202
## 14                   member                   Saturday                 512.2564
```

Here is a chart

```
all_trips_all %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(usertype, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(usertype, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = usertype)) +
  geom_col(position = "dodge") +
  labs(title="Ride duration by day of week", y="Average Duration(seconds)", x="Day of week") +
  guides(fill = guide_legend(title = 'Membership Status'))
```



The graph shows that average ride duration for casuals exceed that of members on all days.

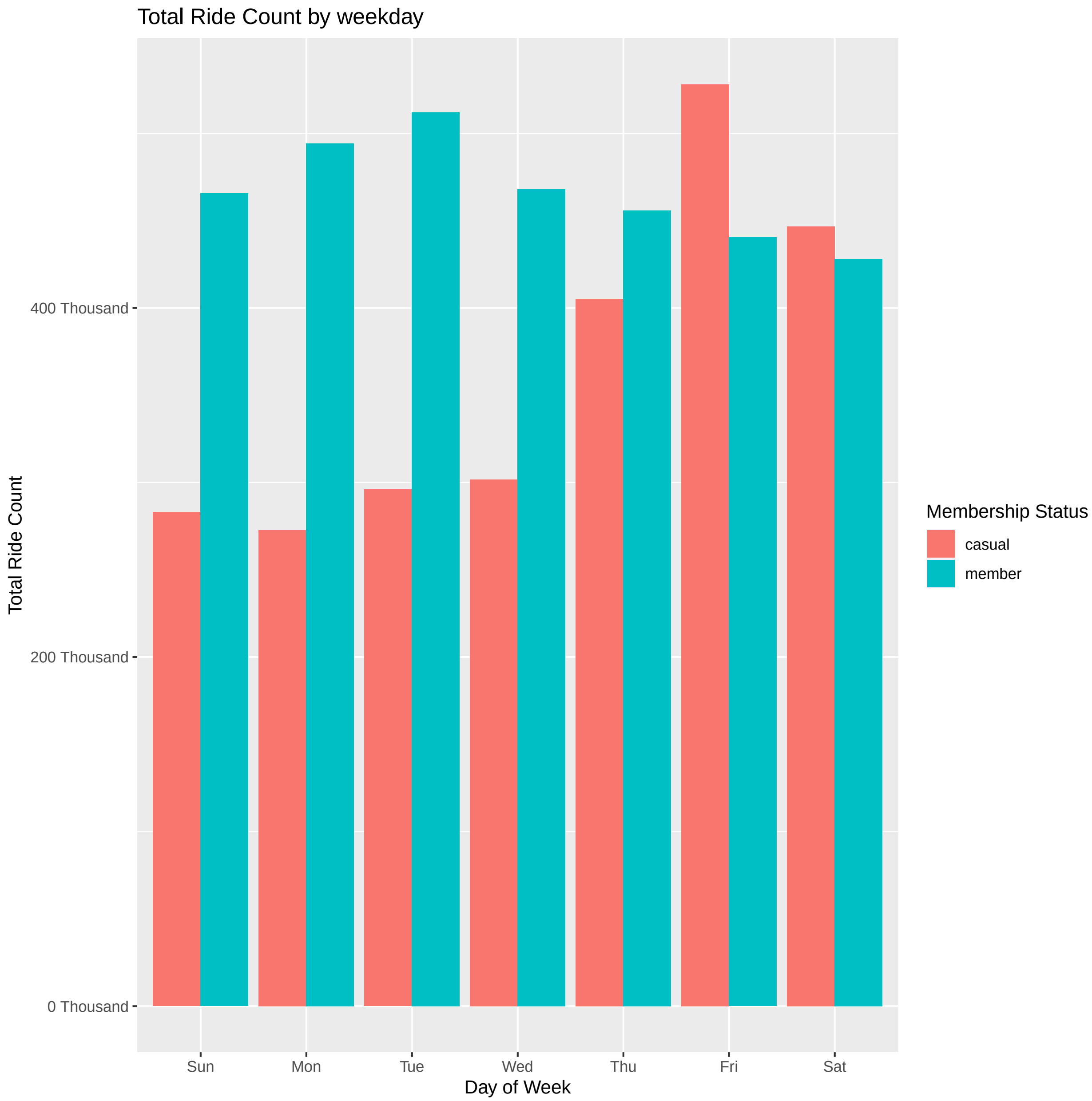
Now lets take a look at the number of rides per day for both groups.

```
all_trips_all %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(usertype, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n())  %>%                    #calculates the number of rides
  arrange(usertype, weekday)
```

```
## # A tibble: 14 x 3
## # Groups:   usertype [2]
##    usertype weekday number_of_rides
##    <chr>    <ord>             <int>
##  1 casual   Sun              283081
##  2 casual   Mon              272761
##  3 casual   Tue              296050
##  4 casual   Wed              301681
##  5 casual   Thu              405212
##  6 casual   Fri              528133
##  7 casual   Sat              446731
##  8 member   Sun              465667
##  9 member   Mon              494332
## 10 member   Tue              512058
## 11 member   Wed              468164
## 12 member   Thu              455944
## 13 member   Fri              440530
## 14 member   Sat              428178
```

Here is the visualization of this table.

```
all_trips_all %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(usertype, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(usertype, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = usertype)) +
    geom_col(position = "dodge") +
    labs(title="Total Ride Count by weekday", y="Total Ride Count", x="Day of Week") +
    guides(fill = guide_legend(title = 'Membership Status')) +
    scale_y_continuous(labels = unit_format(unit = "Thousand", scale = 1e-03))
```



We can see that members ride the bikes the most times on all days except Friday and Saturday. Members rode more consistently while casuals had peak ride volumes on Thursday, Friday, and Saturday.

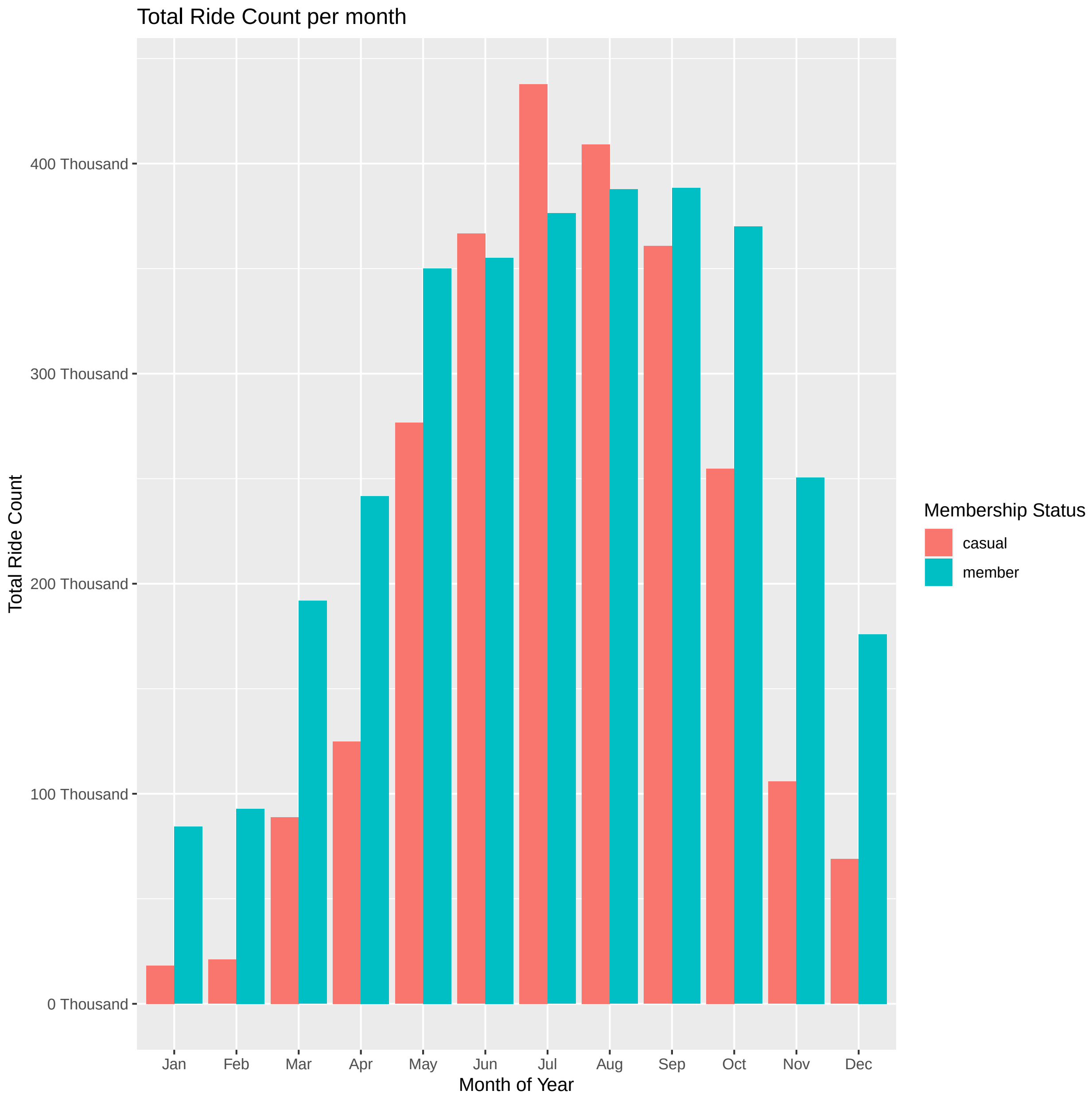
Let us analyze number of rides by month and ride type.

```
all_trips_all %>%
  mutate(month_of_year = month(start_time, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(usertype, month_of_year) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                            #calculates the number of rides and average
            ,average_duration = mean(ride_length)) %>%      # calculates the average duration
  arrange(usertype, month_of_year)
```

```
## # A tibble: 24 x 4
## # Groups:   usertype [2]
##    usertype month_of_year number_of_rides average_duration
##    <chr>    <ord>                   <int>            <dbl>
##  1 casual   Jan                     18190             587.
##  2 casual   Feb                     21045             549.
##  3 casual   Mar                     88876            1040.
##  4 casual   Apr                    124946            1157.
##  5 casual   May                    276637             900.
##  6 casual   Jun                    366644            1378.
##  7 casual   Jul                    437816            1105.
##  8 casual   Aug                    409100             930.
##  9 casual   Sep                    360767             871.
## 10 casual   Oct                    254856             758.
## # ... with 14 more rows
```

Here is the visualization of the table.

```
all_trips_all %>%
  mutate(month_of_year = month(start_time, label = TRUE)) %>%
  group_by(usertype, month_of_year) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(usertype, month_of_year)  %>%
  ggplot(aes(x = month_of_year, y = number_of_rides, fill = usertype)) +
  geom_col(position = "dodge") +
  labs(title="Total Ride Count per month", y="Total Ride Count", x="Month of Year") +
  guides(fill = guide_legend(title = 'Membership Status')) +
  scale_y_continuous(labels = unit_format(unit = "Thousand", scale = 1e-03))
```



We see that the quantity of rides peaks around June, July, and August for casuals. There is great potential to make a sales pitch to casuals during the summer months.

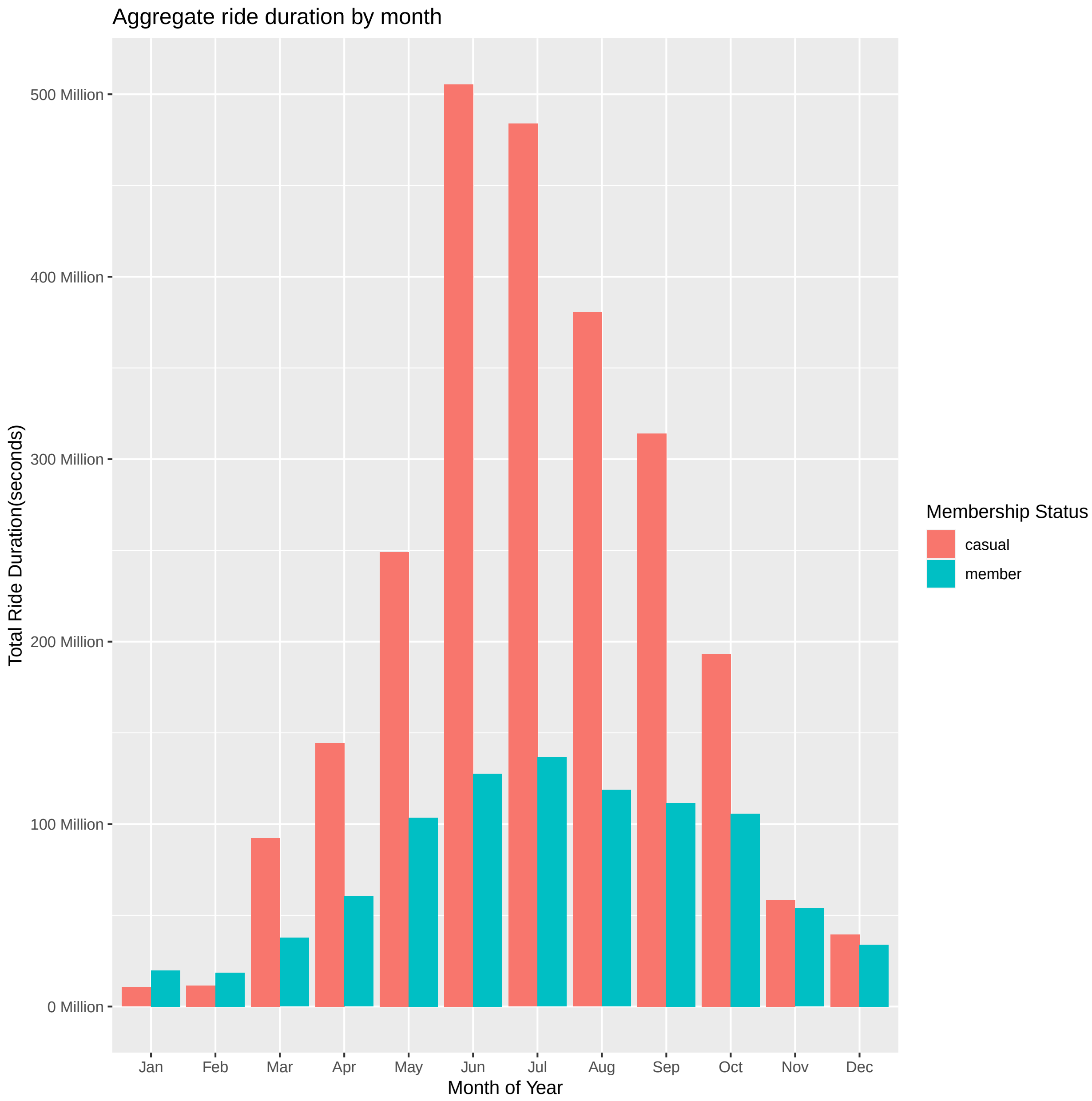
Let us analyze total ride duration for each month to see the variation in ride activity between months.

```
all_trips_all %>%
  mutate(month_of_year = month(start_time, label = TRUE)) %>%
  group_by(usertype, month_of_year) %>%
  summarise(sum_duration = sum(ride_length)) %>%
  arrange(usertype, month_of_year)
```

```
## # A tibble: 24 x 3
## # Groups:   usertype [2]
##    usertype month_of_year sum_duration
##    <chr>    <ord>                <dbl>
##  1 casual   Jan               10686566
##  2 casual   Feb               11553529
##  3 casual   Mar               92392737
##  4 casual   Apr              144512253
##  5 casual   May              249024923
##  6 casual   Jun              505417365
##  7 casual   Jul              483844994
##  8 casual   Aug              380427901
##  9 casual   Sep              314095897
## 10 casual   Oct              193292341
## # ... with 14 more rows
```

Here is the visualization

```
all_trips_all %>%
  mutate(month_of_year = month(start_time, label = TRUE)) %>%
  group_by(usertype, month_of_year) %>%
  summarise(sum_duration = sum(ride_length)) %>%
  arrange(usertype, month_of_year)  %>%
  ggplot(aes(x = month_of_year, y = sum_duration, fill = usertype)) +
  geom_col(position = "dodge") +
  labs(title="Aggregate ride duration by month", y="Total Ride Duration(seconds)", x="Month of Year") +
  guides(fill = guide_legend(title = 'Membership Status')) +
  scale_y_continuous(labels = unit_format(unit = "Million", scale = 1e-06))
```



Insight:

- For casuals, average ride duration is much higher for most months but especially true for June, July, August, and September
- From previous graphs, we learned that they are higher on all days of the week as well
- This graph reinforces the notion that the summer months provides the most amount of opportunities to convert casuals but the opportunity to convince casuals to buy memberships exists in all months and days of the week

Further Exploration:

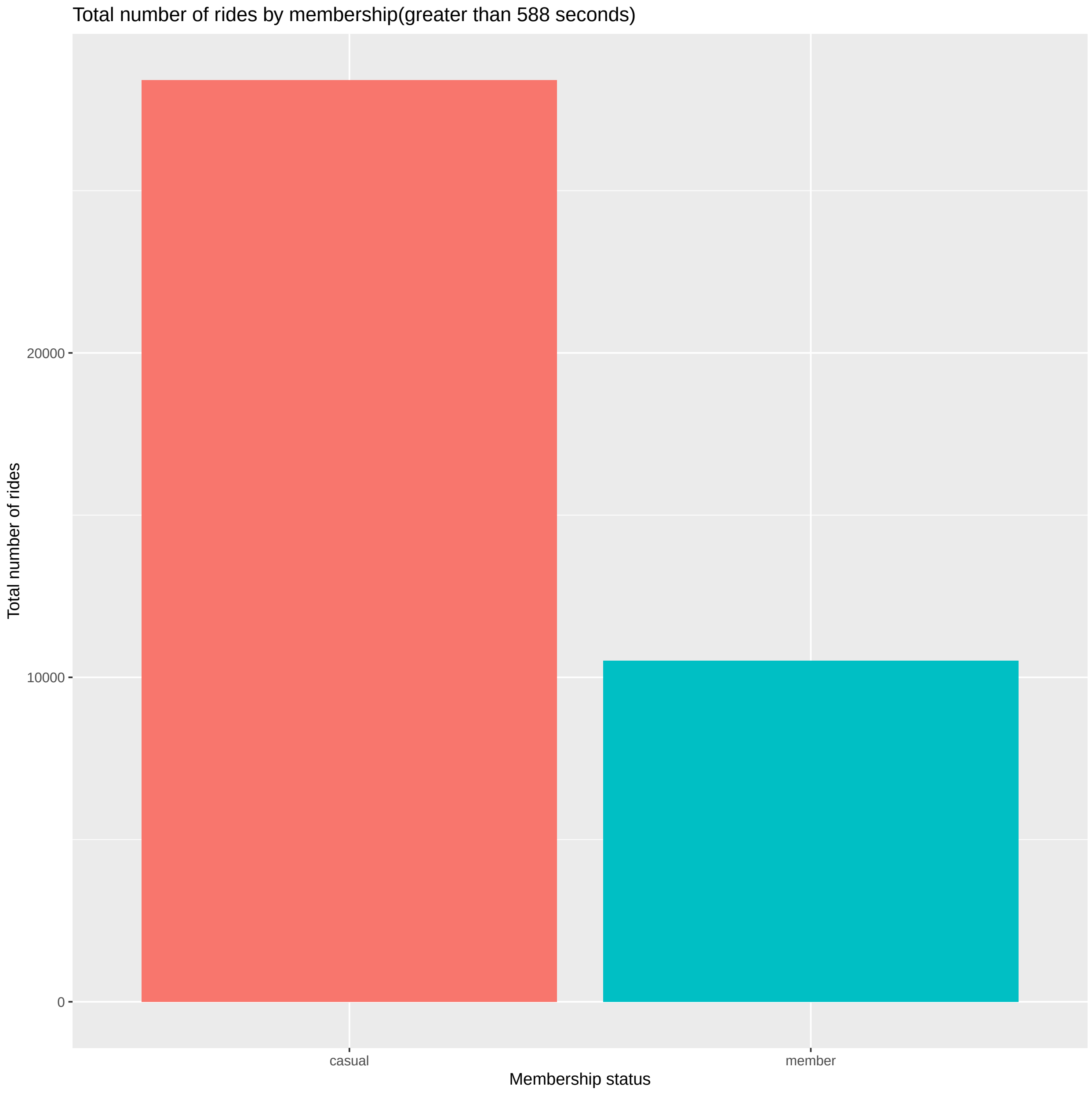
- We know from summary statistics that the mean ride duration is much higher than the median
- This suggest that there many rides with duration on extreme opposite ends of the spectrum

- One question that can be asked is whether those individuals who would be most affected by per minute pricing options would benefit from a membership
- This leaves the question of why casuals have a much larger total ride duration
- Perhaps casuals are much more likely to have high ride duration?

```
all_trips_test2 <- all_trips[(all_trips$ride_length>=588),] #the mean average 588 seconds
```

This creates a subset of the main data frame where ride lengths are greater than the mean.

```
all_trips_test2 %>%
  group_by(usertype) %>%
  summarize(number_of_rides = n()) %>%
  ggplot(aes(x = usertype, y = number_of_rides, fill = usertype)) +
    geom_bar(stat="identity") +
    labs(title="Total number of rides by membership(greater than 588 seconds)", y="Total number of rides
    theme(legend.position = "none")
```



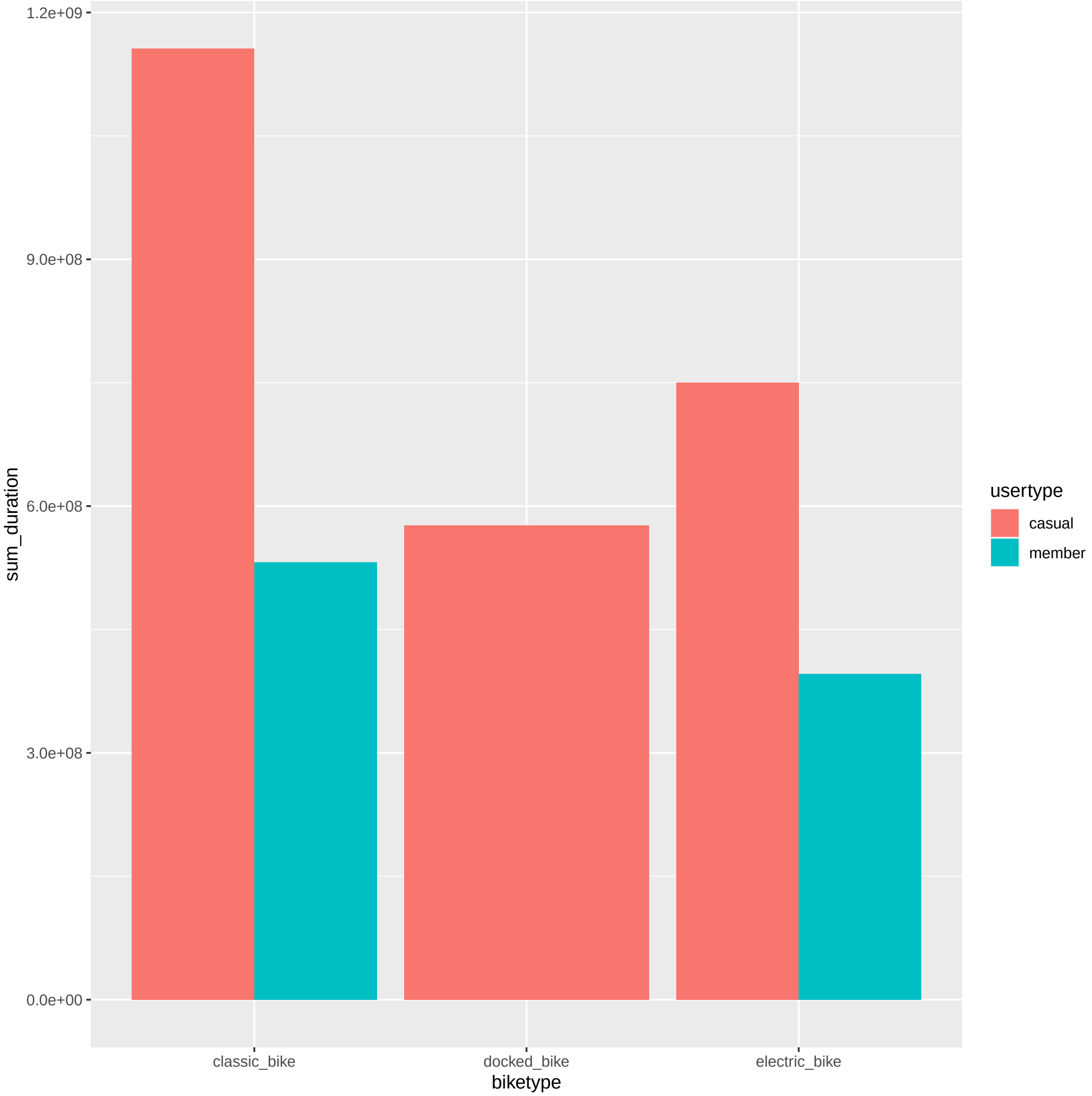
This visualization shows the difference between the number of rides for casuals and members for rides lasting longer than the mean.

Insight:

- Casuals much more frequently ride for more than 20 minutes(average) compared to members
- This is important because those who pay by the minute are missing out on the benefits of a membership

Another question one may have is how the statistics differ between different type of bikes.

```
all_trips_all %>%
  group_by(usertype, biketype) %>%
  summarise(sum_duration = sum(ride_length)) %>%
  arrange(usertype, biketype)  %>%
  ggplot(aes(x = biketype, y = sum_duration, fill = usertype)) +
  geom_col(position = "dodge")
```



Thoughts:

- We can ignore docked bikes since they are not actual rides
- This graph shows that classic bikes are much more popular than electric bikes. However, the difference in popularity is greater for the casuals than for the members.
- Perhaps electric bikes are more expensive on a per minute basis and casuals hesitate to pay for it.
- Perhaps members are less hesitant to ride it due to the fact that they have unlimited access to all bikes

Summary of findings:

1. The total amount of time that casuals spend riding bikes exceed that of members; this presents an opportunity to gain plenty of potential new members.
2. Members ride bikes more often than casuals; I believe this is because members can take full advantage of unlimited rides and choose to do so.
3. The data shows that number of rides are higher for members on the weekdays. This is likely due to members using the bikes to travel to a fro from work and home since people are most likely working on the weekdays instead of riding bikes for recreation.
4. The average amount of time spent riding bikes differ by day of the week; more specifically Thursdays, Fridays, and Saturdays are the busiest for casuals. The average amount of time spent riding bikes also differ by month of the year as well; specifically bikes get used the most in total in the Summer months.
5. My analysis shows that classic bikes are more popular than electric bikes for both groups; however, for members, the classic and electric bikes are closer in popularity compared to casuals.

How a company may utilize this information:

1. The company should engage in a marketing campaign to incentivize casual riders to buy a membership as the return on investment is likely to be high due to the fact that the total aggregate riding time for non-members exceed that of members by a wide margin.
2. The company should market unlimited rides as one of the benefits of obtaining a membership.
3. Since transportation to work may be one of the reasons why someone might purchase a membership, it may be wise to set up bike stations in both the business district of the city and the residential areas. In the marketing campaign, the company can tout how convenient the bike locations are; non members might start using the bikes to travel to work which might naturally lead them to purchase memberships.
4. Bike usage is the greatest during the Summer months. This is the prime opportunity to convert casuals to members. It may be beneficial to have a Summer sale to further incentivize non members to buy a membership during that critical time.
5. Since members are less hesitant to use electric bikes compared to casuals, the company can market unlimited rides with electric bikes as part of a membership to appeal to casuals who might want to ride electric bikes.