Analysis of Cyclistic Bike Shares

Faisal Mahmood

2022-07-19

Introduction

In this assignment, I chose to analyze the data of a fictional bikeshare company called Cyclistic, following each step of the analysis process: **Ask**, **Prepare**, **Process**, **Analyze**, **Share**, and **Act**.

My assignment was to download 12 months worth of data, follow all steps of the data analysis process, and figure out how to increase the number of paying members for the bikeshare company. The data contained more than 5 million observations, with each observation representing an individual bike ride. Therefore, I chose to use R for my analysis, since R enabled me to analyze a large dataset, clean the data, perform calculations, and visualize the data.

The Google Data Analytics course did provide an R script, which gave me guidance as I completed my analysis. However, I took many further steps as I scrutinized the data, cleaned it up, and looked for details that may have not been previously apparent.

Ask

The business task for this project is to determine how annual members of the Cyclistic bikeshare program use Cyclistic bikes differently from casual riders, for the purpose of figuring out how to encourage casual riders to sign up as annual members.

Casual riders are customers who use single-ride or full-day passes, while annual members pay an annual fee for access to Cyclistic bikes.

The stakeholders of this assignment are defined as the people who have invested time, interest, and resources into the project. The stakeholders include

- Lily Moreno – my manager and the director of marketing. She is responsible for advertising the bikeshare program, so my analysis will influence the decisions she makes in her marketing.
- Cyclistic Executive Team – The executive team will be making the key decision on whether or not to approve the marketing program, which will be determined by my analysis.

Since the financial analysts have concluded that annual members are more profitable than casual riders, the objective is to figure out how to convert casual riders into full members.

Prepare

The data for this project is located in this AWS bucket, organized by month, and for this assignment I used data from June 2021 through May 2022. The data is public, and it is licensed to be used for the purpose of this analysis, as long as no personally identifiable information is included. The data is organized by each individual bike trip, and each dataset is organized by month. The datasets are large, with a total of more than 5.8 million bike trips listed throughout the period of June 2021 through May 2022.

For each individual bike trip, the following attributes are listed:

- Ride ID code
- Bike type (classic bike, electric bike, docked bike)
- Starting and ending times and dates
- Starting and ending station and end station names and station IDs (for most observations)
- Starting and ending latitude and longitude (for most observations)
- Membership category (whether the rider is a casual biker or paying member)

Process

I used RStudio and the programming language R for my analysis. The reason for this decision is the size of the dataset. With 5.8 million observations divided into 12 monthly CSV documents, I concluded that a spreadsheet program would not be a practical tool. I could have chosen SQL for the data transformation and Tableau for the visualization, but R is also effective for large datasets as well as data visualization.

I downloaded each of the 12 monthly datasets into the following filepath by using the `setwd()` function.

```
setwd("~/Documents/Google Data Analytics/Capstone/Cyclistic-Monthly-Datasets/Raw-output/CSV-files")
```

Afterwards, I loaded the following packages:

```
library(tidyverse)
library(lubridate)
options(dplyr.summarise.inform = FALSE)
```

The `tidyverse` library contains a set of packages that are very useful for data wrangling, data cleaning, data manipulation, and visualization (via ggplot). The `lubridate` package is useful for manipulating data that involves dates and times.

I used the `list.files()` command to list each of the monthly datasets within the filepath, and stored it in the variable `bike_files`. Then I ran the `bike_files` variable to display the names of the CSV files that I used for the analysis. Each of the files can be found as ZIP files in the AWS bucket that was linked in the **Prepare** section. I ran a `for` loop to apply the `read_csv()` function for each of the 12 files, and also to name them from `df1` to `df12`.

```
bike_files <- list.files(pattern='csv')
bike_files
```

```
##  [1] "202106-divvy-tripdata.csv" "202107-divvy-tripdata.csv"
##  [3] "202108-divvy-tripdata.csv" "202109-divvy-tripdata.csv"
##  [5] "202110-divvy-tripdata.csv" "202111-divvy-tripdata.csv"
##  [7] "202112-divvy-tripdata.csv" "202201-divvy-tripdata.csv"
##  [9] "202202-divvy-tripdata.csv" "202203-divvy-tripdata.csv"
## [11] "202204-divvy-tripdata.csv" "202205-divvy-tripdata.csv"
```

```
for(i in 1:length(bike_files)) {
  assign(paste0("df", i),
  read_csv(bike_files[i]))
}
```

Before merging the dataframes, I created a `df_colnames` dataframe that consisted of the column names of each of the 12 newly created dataframes, with each attribute of the dataframe representing the column names for one of the 12 dataframes. I wanted to verify that each of the 12 dataframes has the same column names in the same order.

```
df_colnames <- data.frame(c(colnames(df1)), c(colnames(df2)),
                          c(colnames(df3)), c(colnames(df4)),
                          c(colnames(df5)), c(colnames(df6)),
                          c(colnames(df7)), c(colnames(df8)),
                          c(colnames(df9)), c(colnames(df10)),
                          c(colnames(df11)), c(colnames(df12)))
```

I chose to name each column in the `df_colnames` dataframe with the dataframe name that corresponded to the column. So column names of the dataframe `df1` would be listed under the column titled `df1` in `df_colnames`.

```
dataframe_number <- paste("df", c(1:length(bike_files)), sep="")
colnames(df_colnames) <- c(dataframe_number)
```

The following list consists of the column names for `df1`.

```
df_colnames$df1
```

```
##  [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"
```

I observed that column names for all the dataframes are identical, but I also wanted to verify it for the purpose of this analysis. I used the `sapply()` function to apply the `identical()` function to all the columns in the `df_colnames` dataframe. This would enable me to verify that the remaining column names, corresponding to the remaining dataframes, are identical to the column names in `df1`. The function returned as `TRUE`, so this confirms that all the dataframes for each month have the same attributes, or columns.

```
all(sapply(list(df_colnames$df2, df_colnames$df3,
                df_colnames$df4, df_colnames$df5,
                df_colnames$df6, df_colnames$df7,
                df_colnames$df8, df_colnames$df9,
                df_colnames$df10, df_colnames$df11,
                df_colnames$df12), FUN = identical,
                df_colnames$df1))
```

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

Since the columns are identical in all 12 dataframes, I confirmed that I was ready to merge the dataframes. I used the `bind_rows()` function to merge all 12 monthly dataframes as `tripdata`.

```
tripdata <- bind_rows(df1, df2, df3, df4, df5, df6,
                      df7, df8, df9, df10, df11, df12)
```

Analyze

Once the dataframes were combined into `tripdata`, I did a quick, basic data exploration to ensure that the data is clean. I ran the `colnames()`, `nrow()`, `dim()` (dimensions), `head()`, `tail()`, `str()` (structure),

`summary()`, and `colSums(is.na())` commands to quickly glance at the dataframe and check for any glaring errors. The `colSums(is.na())` command counts the number of missing values in each column. Most columns did not have any missing values, and those that did have missing values were not essential for the purpose of this analysis. So most of the data looked relatively clean, but still not ideal.

```
nrow(tripdata)
dim(tripdata)
head(tripdata)
tail(tripdata)
str(tripdata)
summary(tripdata)
colSums(is.na(tripdata))
```

In addition, I also produced tables that counted the unique values for the `rideable_type` and `member_casual` columns. I did this to familiarize myself with all the categories that were listed, and the number of observations for each category. They all appeared normal.

```
table(tripdata$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##       3217737        274447       2368592
```

```
table(tripdata$member_casual)
```

```
##
##  casual  member
## 2559857 3300919
```

After a quick glance using the `head()` function on the dataframe above, I noticed that the observations (rows) were out of chronological order. So I rearranged the start times using the `arrange()` function, as seen below.

```
tripdata <- tripdata %>% arrange(ymd_hms(tripdata$started_at))
```

Afterwards, I created a column for the length of each ride, by subtracting the start time column from the end time column. The results were given in seconds, and then I converted the data to numeric type for the sake of further analysis.

```
tripdata <- tripdata %>% mutate(ride_length = ended_at - started_at)
tripdata$ride_length <- as.numeric(tripdata$ride_length)
```

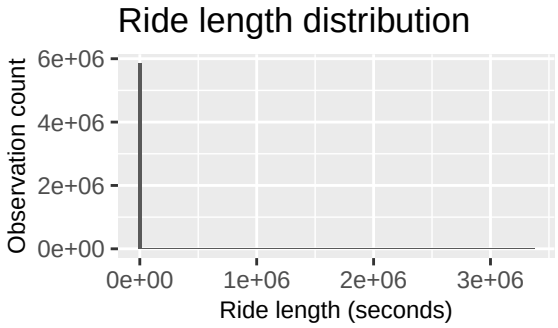
I did another quick glance using the `colnames()` and `summary()` functions, only to find that the `ride_length` column had some negative values, as well as some extremely high positive outliers. The highest `ride_length` value was more than 2700 times the mean value, and more than 4900 times the median value. So the outliers immediately looked suspicious to me.

```
summary(tripdata$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -3482     382     680    1241    1236 3356649
```

To visualize this issue, I attempted to plot a histogram of the ride length distribution, only to find one very large bar on the far left side of the chart, with virtually no visible data throughout the remainder of the chart.

```
ggplot(tripdata, aes(x=ride_length)) +
  geom_histogram(bins=100) + labs(title = "Ride length distribution") +
  xlab("Ride length (seconds)") + ylab("Observation count") +
  theme(axis.title=element_text(size=9))
```



To verify that multiple observations with extreme outliers (which are not visible on the above plot) exist, I created another dataframe `max_duration`, in which I arranged the `ride_length` column in descending order, and then viewed the top 10 values. All were greater than 2.4 million seconds, and the top three observations exceeded 3 million seconds.

```
max_duration <- tripdata %>% arrange(desc(ride_length)) %>%
  select(started_at, ended_at, ride_length)
head(max_duration, 10)
```

```
## # A tibble: 10 x 3
##    started_at          ended_at            ride_length
##    <dttm>              <dttm>                    <dbl>
##  1 2021-06-05 02:27:26 2021-07-13 22:51:35     3356649
##  2 2021-06-04 22:03:33 2021-07-13 14:15:14     3341501
##  3 2021-06-05 23:33:51 2021-07-12 13:55:14     3162083
##  4 2021-07-08 19:29:49 2021-08-11 21:56:58     2946429
##  5 2021-06-05 21:47:40 2021-07-08 13:18:31     2820651
##  6 2021-07-08 15:13:08 2021-08-06 13:18:39     2498731
##  7 2021-08-01 18:53:10 2021-08-30 16:42:20     2497750
##  8 2021-07-10 15:59:21 2021-08-07 22:43:57     2443476
##  9 2021-10-02 18:35:36 2021-10-31 01:00:37     2442301
## 10 2021-07-03 18:39:43 2021-07-31 19:00:58     2420475
```

I then created a similar dataframe `min_duration`, and viewed the 10 lowest ride_length values. All were negative. Since no bike ride can have an end time that occurs before the start time, I used the `subset()` function to eliminate the negative ride_length values from the dataframe, and stored the cleaned dataframe in `tripdata_v2`.

```
tripdata_v2 <- subset(tripdata, ride_length >=0)
```

Next, I created columns for each month, day, year, and day of the week, so that further analysis could be more easily done with any of the desired variables.

```
#Add columns for date, month, day, year
tripdata_v2$date <- as.Date(tripdata_v2$started_at)
tripdata_v2$month <- format(as.Date(tripdata_v2$date), "%m")
tripdata_v2$day <- format(as.Date(tripdata_v2$date), "%d")
tripdata_v2$year <- format(as.Date(tripdata_v2$date), "%Y")
tripdata_v2$day_of_week <- tripdata_v2$started_at %>% wday(label=TRUE)
```

Afterwards, I used the `aggregate()` function to calculate the mean, median, maximum, and minimum ride length by membership category. So for both members and casual riders, each of those statistics were calculated using the function.

```
#Average ride length (seconds) for paying members and casual riders
aggregate(tripdata_v2$ride_length ~ tripdata_v2$member_casual,
          FUN = mean)
```

```
##   tripdata_v2$member_casual tripdata_v2$ride_length
## 1                    casual               1832.9602
## 2                    member                782.6623
```

```
#Median ride length (seconds) for paying members and casual riders
aggregate(tripdata_v2$ride_length ~ tripdata_v2$member_casual, FUN = median)
```

```
##   tripdata_v2$member_casual tripdata_v2$ride_length
## 1                    casual                     916
## 2                    member                     547
```

```
#Maximum ride length (seconds) for paying members and casual riders
aggregate(tripdata_v2$ride_length ~ tripdata_v2$member_casual, FUN = max)
```

```
##   tripdata_v2$member_casual tripdata_v2$ride_length
## 1                    casual                 3356649
## 2                    member                   93594
```

```
#Minimum ride length (seconds) for paying members and casual riders
aggregate(tripdata_v2$ride_length ~ tripdata_v2$member_casual, FUN = min)
```

```
##   tripdata_v2$member_casual tripdata_v2$ride_length
## 1                    casual                       0
## 2                    member                       0
```

I was surprised to see that the data showed casual riders with much higher average and median ride lengths. If this data is correct, then enrolling those casual riders as paying members could be a major opportunity.

I also used the aggregate function to calculate the mean ride lengths and number of rides for both paying members and casual riders by the day of the week.

```
aggregate(tripdata_v2$ride_length ~ tripdata_v2$member_casual + tripdata_v2$day_of_week, FUN = mean)
```

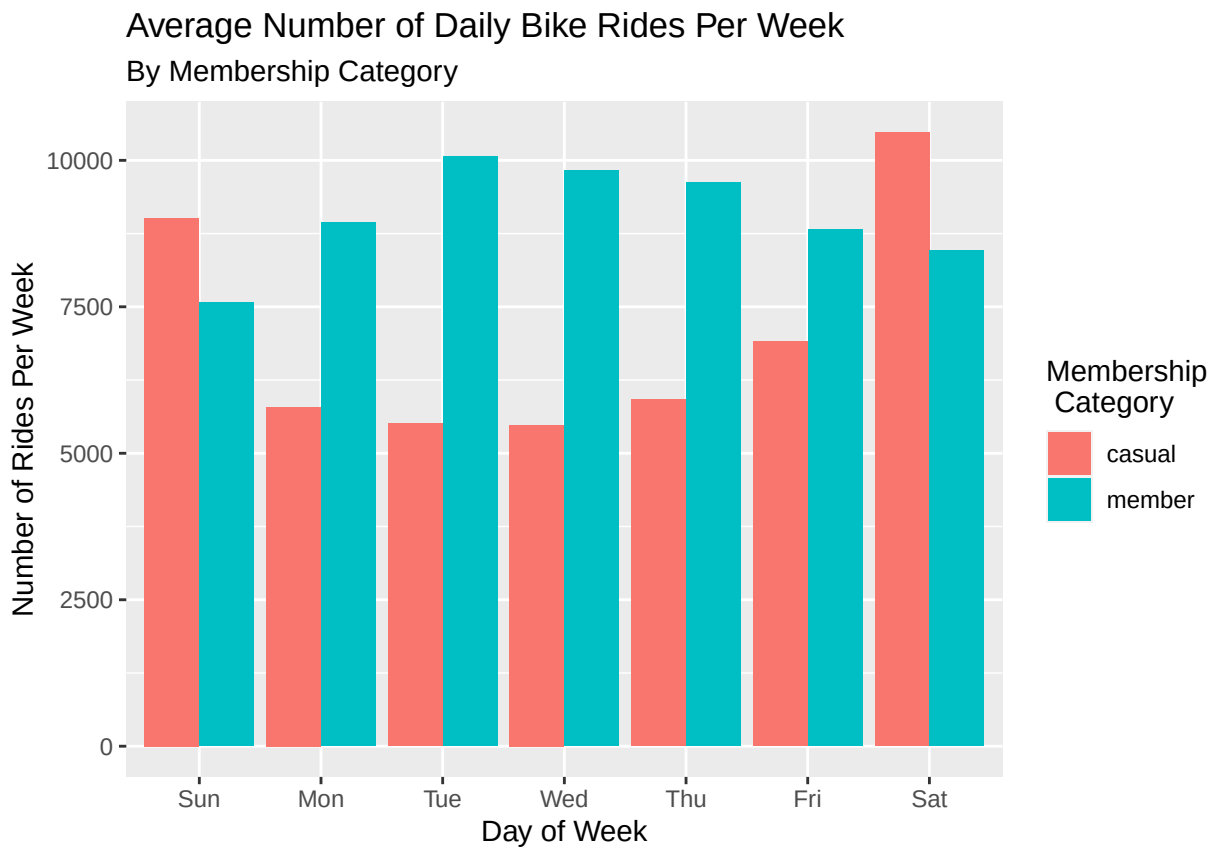
```
##    tripdata_v2$member_casual tripdata_v2$day_of_week tripdata_v2$ride_length
## 1                     casual                     Sun               2120.6685
## 2                     member                     Sun                887.6333
## 3                     casual                     Mon               1831.6798
## 4                     member                     Mon                758.7213
## 5                     casual                     Tue               1574.5248
## 6                     member                     Tue                736.7175
## 7                     casual                     Wed               1599.3807
## 8                     member                     Wed                738.5552
## 9                     casual                     Thu               1662.4429
## 10                    member                     Thu                746.5689
## 11                    casual                     Fri               1726.7647
## 12                    member                     Fri                766.9394
## 13                    casual                     Sat               2010.3801
## 14                    member                     Sat                877.4190
```

However, this data would be much easier to communicate if it is visualized.

Share

I visualized the data showing the number of rides by day of the week for casual riders and paying members, and found that paying members have a greater number of rides on weekdays, while casual riders have a greater number of rides on the weekends. Instead of showing the raw total number of bike rides, I divided that number by 52.1429 (the number of weeks in a year) to show the number of rides on a weekly basis. This way, the numbers are much simpler to comprehend.

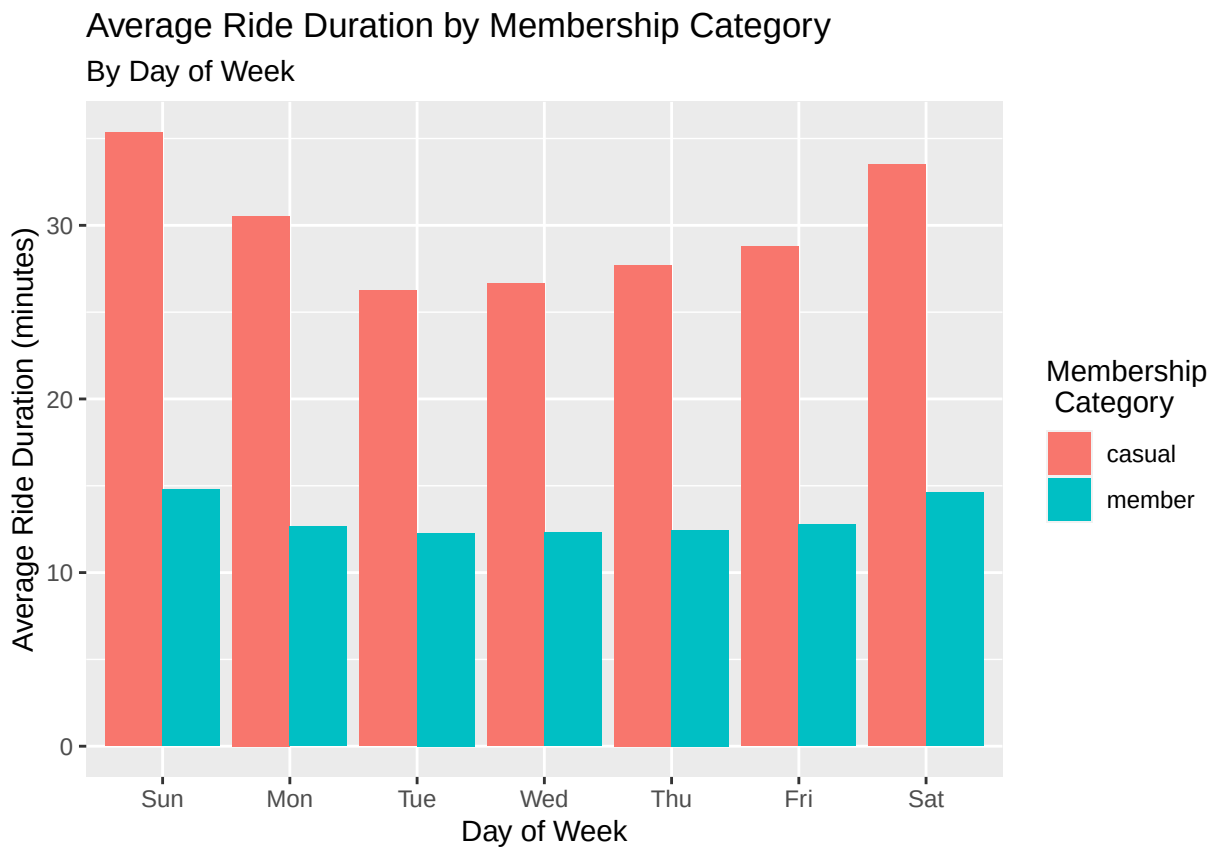
```
tripdata_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = number_of_rides/52.1429, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Average Number of Daily Bike Rides Per Week",
       subtitle = "By Membership Category",
       x = "Day of Week",
       y = "Number of Rides Per Week",
       fill = "Membership \n Category")
```



My next visualization showed the average ride duration for both membership categories by day of the week. I chose to display the ride duration in minutes rather than seconds, so I divided the duration variable by 60.

Casual riders consistently had much higher average ride length than paying members, with both categories showing longer trips on the weekends compared to weekdays.

```
tripdata_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = average_duration/60, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Average Ride Duration by Membership Category",
       subtitle = "By Day of Week",
       x = "Day of Week",
       y = "Average Ride Duration (minutes)",
       fill = "Membership \n Category")
```



I am skeptical of the above data, mainly because of the extremely high outliers for ride duration that were found. Such outliers are likely to skew the average duration significantly higher than the median times, so I also chose to visualize the median ride duration to get a sense of the typical duration, without any skewed data significantly interfering with the results.

```
tripdata_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,median_duration = median(ride_length)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = median_duration/60, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Median Ride Duration by Membership Category",
       subtitle = "By Day of Week",
       x = "Day of Week",
       y = "Median Ride Duration (minutes)",
       fill = "Membership \n Category")
```

To show the vast disparity between the mean and median ride lengths for both categories, I calculated the mean and median ride length for each membership category, and then calculated the mean/median ratios.

```
#Mean/Median for both members and casual riders
casual_mean_duration <- tripdata_v2 %>%
  filter(member_casual == "casual") %>%
  pull(ride_length) %>% mean() #be sure to use pull() function
casual_median_duration <- tripdata_v2 %>%
  filter(member_casual == "casual") %>%
  pull(ride_length) %>% median()

member_mean_duration <- tripdata_v2 %>%
  filter(member_casual == "member") %>%
  pull(ride_length) %>% mean()
member_median_duration <- tripdata_v2 %>%
  filter(member_casual == "member") %>%
  pull(ride_length) %>% median()

print(paste("Ratio of mean duration to median duration for casual riders is",
casual_mean_duration/casual_median_duration))
```

```
## [1] "Ratio of mean duration to median duration for casual riders is 2.00104828904864"
```

```
print(paste("Ratio of mean duration to median duration for paying members is",
member_mean_duration/member_median_duration))
```

```
## [1] "Ratio of mean duration to median duration for paying members is 1.43082691041431"
```

I found that for casual riders, the mean duration is 2.00 times higher than the median duration. For paying members, the mean duration is greater by a factor of 1.43.

I used the `quantile()` function to calculate the duration for both casual riders and paying members at the following percentiles: 95% 99% 99.5% 99.9% 99.99% 99.999%

```
tripdata_v2 %>%
  filter(member_casual == "casual") %>%
  pull(ride_length) %>%
  quantile(c(0.95, 0.99, 0.995, 0.999, 0.9999, 0.99999))
```

```
##        95%        99%      99.5%      99.9%     99.99%    99.999%
##    4717.00   10026.00   14980.07   89994.00  697040.95 1954260.84
```

```
print(paste("The 99.999th percentile for casual rider trip duration is", tripdata_v2 %>%
  filter(member_casual == "casual") %>%
  pull(ride_length) %>%
  quantile(0.99999)/3600/24, "days"))
```

```
## [1] "The 99.999th percentile for casual rider trip duration is 22.6187597256982 days"
```

I found that at the 99.999th percentile, casual riders had a duration of 1.95 million seconds, or 22.6 days.

```
tripdata_v2 %>%
  filter(member_casual == "member") %>%
  pull(ride_length) %>%
  quantile(c(0.95, 0.99, 0.995, 0.999, 0.9999, 0.99999))
```

```
##      95%      99%    99.5%    99.9%   99.99%  99.999%
##  2014.00  3173.00  4219.00 12431.32 89992.00 89996.00
```

```
print(paste("The 99.999th percentile for paying members' trip duration is", tripdata_v2 %>%
  filter(member_casual == "member") %>%
  pull(ride_length) %>%
  quantile(0.99999)/3600, "hours"))
```

```
## [1] "The 99.999th percentile for paying members' trip duration is 24.9988888888889 hours"
```

Meanwhile, at that same percentile for paying members, the duration was just under 90000 seconds, or 25 hours.

```
tripdata_v2 %>%
  filter(member_casual == "casual") %>%
  pull(ride_length) %>%
  max()
```

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

```
print(paste("The maximum value found for casual rider trip duration is", tripdata_v2 %>%
  filter(member_casual == "casual") %>%
  pull(ride_length) %>%
  max()/3600/24, "days"))
```

```
## [1] "The maximum value found for casual rider trip duration is 38.8501041666667 days"
```

```
tripdata_v2 %>%
  filter(member_casual == "member") %>%
  pull(ride_length) %>%
  max()
```

```
## [1] 93594
```

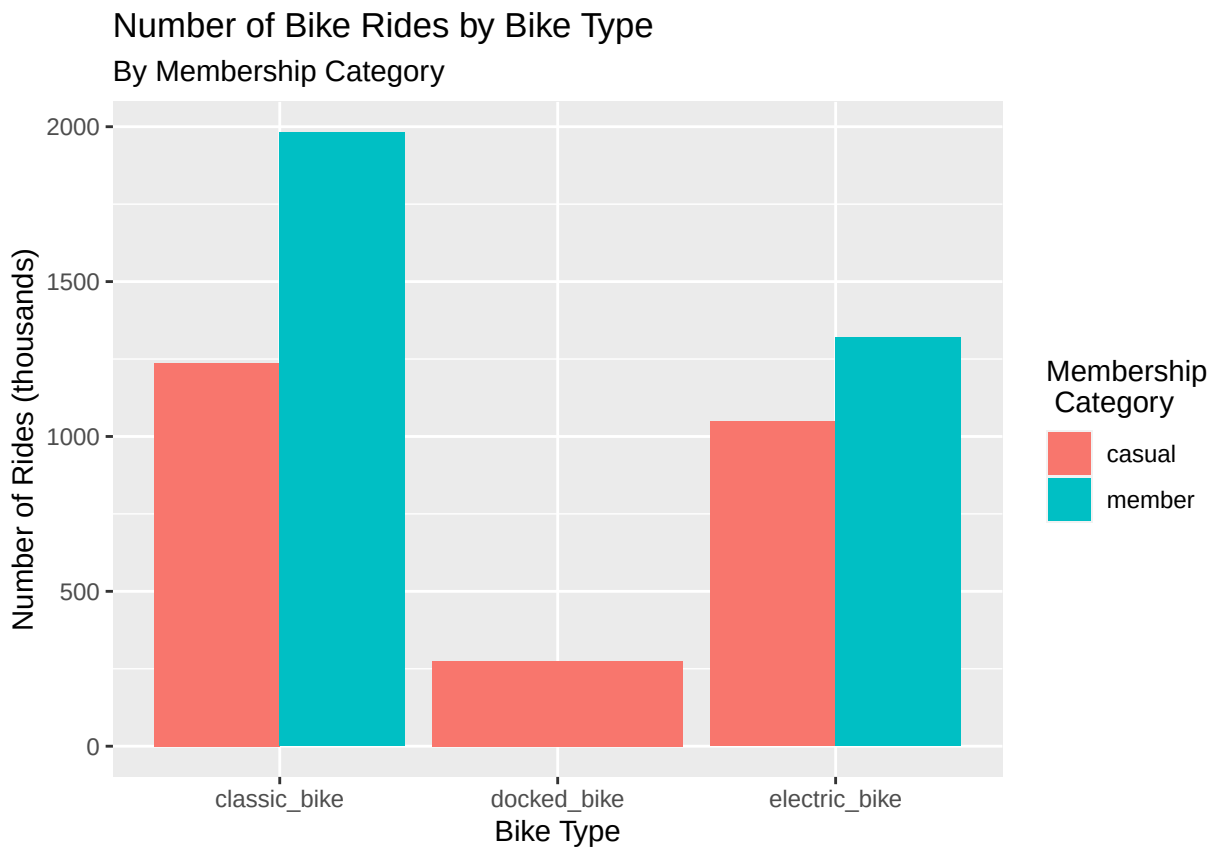
```
print(paste("The maximum value found for paying member trip duration is", tripdata_v2 %>%
    filter(member_casual == "member") %>%
    pull(ride_length) %>%
    max()/3600, "hours"))
```

```
## [1] "The maximum value found for paying member trip duration is 25.9983333333333 hours"
```

The highest recorded duration (calculated using the `max()` function) for a casual rider was 3356649 seconds, or nearly 39 days. For a paying member, that number was 93594 seconds, or nearly 26 hours.

To further investigate the outliers, I chose to break down the data by bike type. My first step was to determine how many bike rides of each type were observed in the data, broken down by membership category.

```
#Group by Bike Type
tripdata_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, rideable_type)  %>%
  ggplot(aes(x = rideable_type,
             y = number_of_rides/1000,
             fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Number of Bike Rides by Bike Type",
       subtitle = "By Membership Category",
       x = "Bike Type",
       y = "Number of Rides (thousands)",
       fill = "Membership \n Category")
```



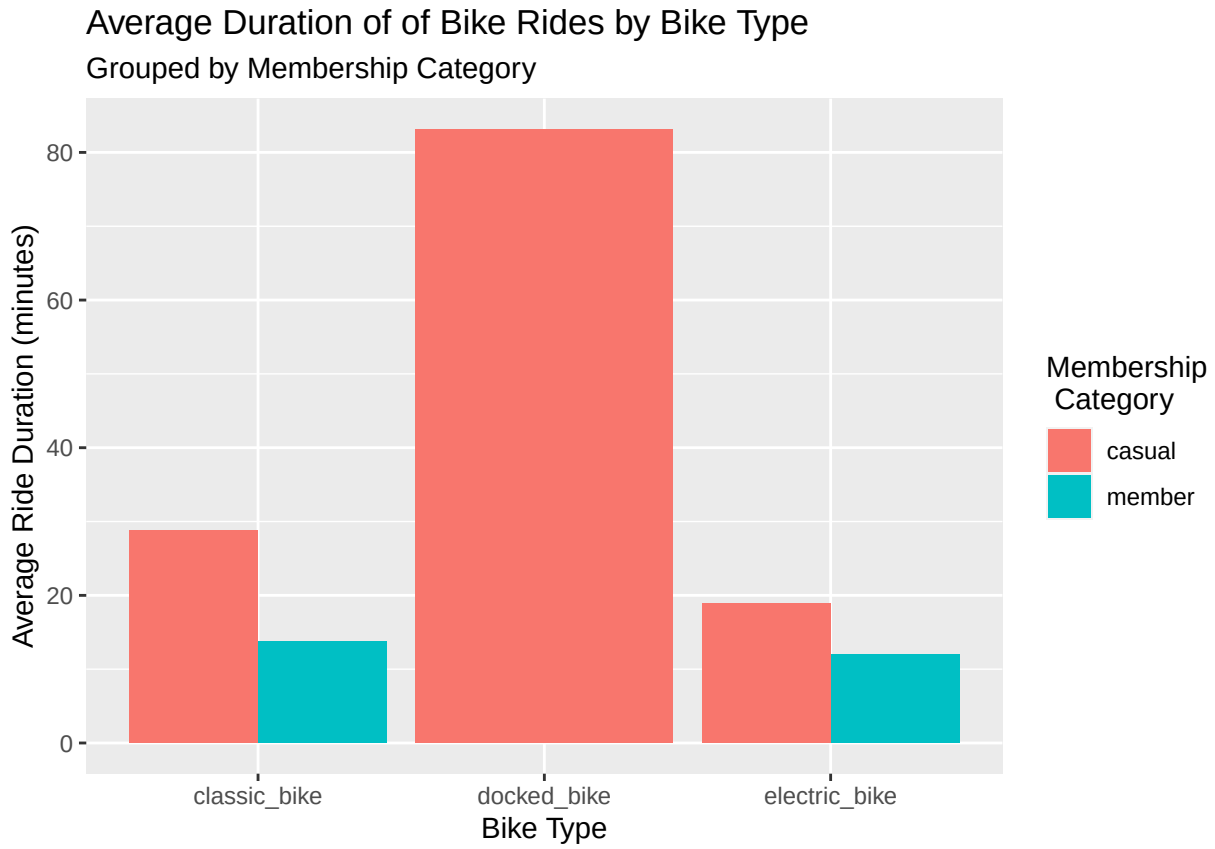
Paying members had more bike rides on the classic bike and electric bike than casual bikers. However, I was surprised to see that all rides with docked bikes were from casual riders. Afterwards, I chose to visualize the average duration of the rides for each bike type, again grouped by membership category.

```
tripdata_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n(),
            average_duration_min = mean(ride_length/60)) %>%
  arrange(member_casual, rideable_type)
```

```
## # A tibble: 5 x 4
## # Groups:   member_casual [2]
##   member_casual rideable_type number_of_rides average_duration_min
##   <chr>         <chr>                  <int>                <dbl>
## 1 casual        classic_bike         1236507                 28.7
## 2 casual        docked_bike           274442                 83.1
## 3 casual        electric_bike        1048847                 19.0
## 4 member        classic_bike         1981150                 13.7
## 5 member        electric_bike        1319691                 12.0
```

```
tripdata_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, rideable_type)  %>%
```

```
  ggplot(aes(x = rideable_type, y = average_duration/60,
             fill = member_casual)) + geom_col(position = "dodge") +
  labs(title = "Average Duration of of Bike Rides by Bike Type",
       subtitle = "Grouped by Membership Category",
       x = "Bike Type",
       y = "Average Ride Duration (minutes)",
       fill = "Membership \n Category")
```



Among paying members, there was a very minor difference between the average ride duration on classic bikes and electric bikes. However, the casual riders had a substantially longer average duration on the classic bikes. Furthermore, the average duration of docked bike rides (only from casual riders) was far longer compared to any other bike type.

Since there were extreme outliers in the ride duration data for casual bikers, it was worth exploring the bike types for which the outliers occurred.

```
print(paste("The maximum value found for a classic bike trip duration is",tripdata_v2 %>%
  filter(rideable_type == "classic_bike") %>%
  pull(ride_length) %>%
  max()/3600, "hours."))
```

```
## [1] "The maximum value found for a classic bike trip duration is 25.9988888888889 hours."
```

```
print(paste("The maximum value found for an electric bike trip duration is",tripdata_v2 %>%
  filter(rideable_type == "electric_bike") %>%
```

```
  pull(ride_length) %>%
  max()/3600, "hours."))
```

```
## [1] "The maximum value found for an electric bike trip duration is 8.12111111111111 hours."
```

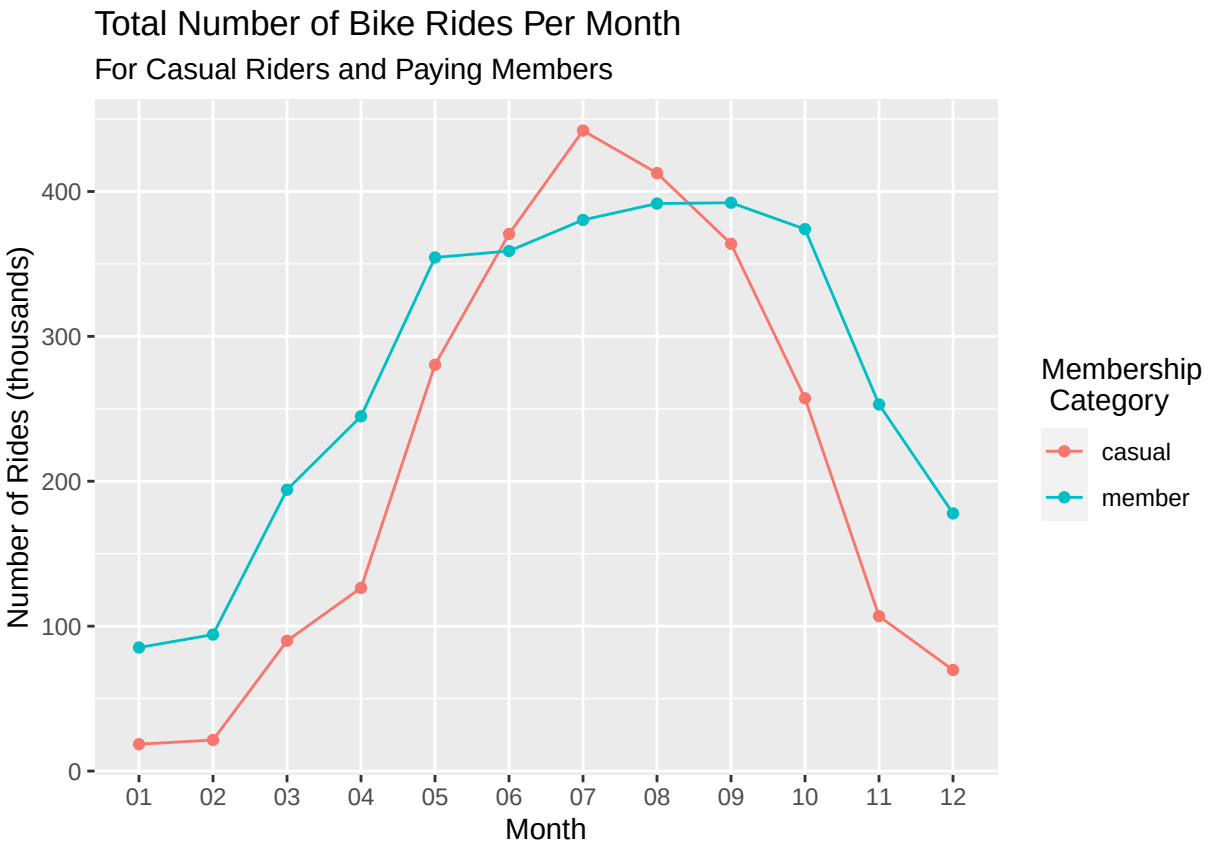
```
print(paste("The maximum value found for a docked bike trip duration is",tripdata_v2 %>%
  filter(rideable_type == "docked_bike") %>%
  pull(ride_length) %>%
  max()/3600/24, "days."))
```

```
## [1] "The maximum value found for a docked bike trip duration is 38.8501041666667 days."
```

Based on the maximum values of ride duration for each bike type, the largest outliers by far were found with docked bike trips. The classic and electric bikes also contained significant outliers that were well above the average ride lengths. However, none of them were even close to the extreme ride lengths found with the longer docked bike trips.

Finally, I was also interested in the seasonal distribution of the bike ride data, so I decided to plot the number of rides for each membership category by each month of the year.

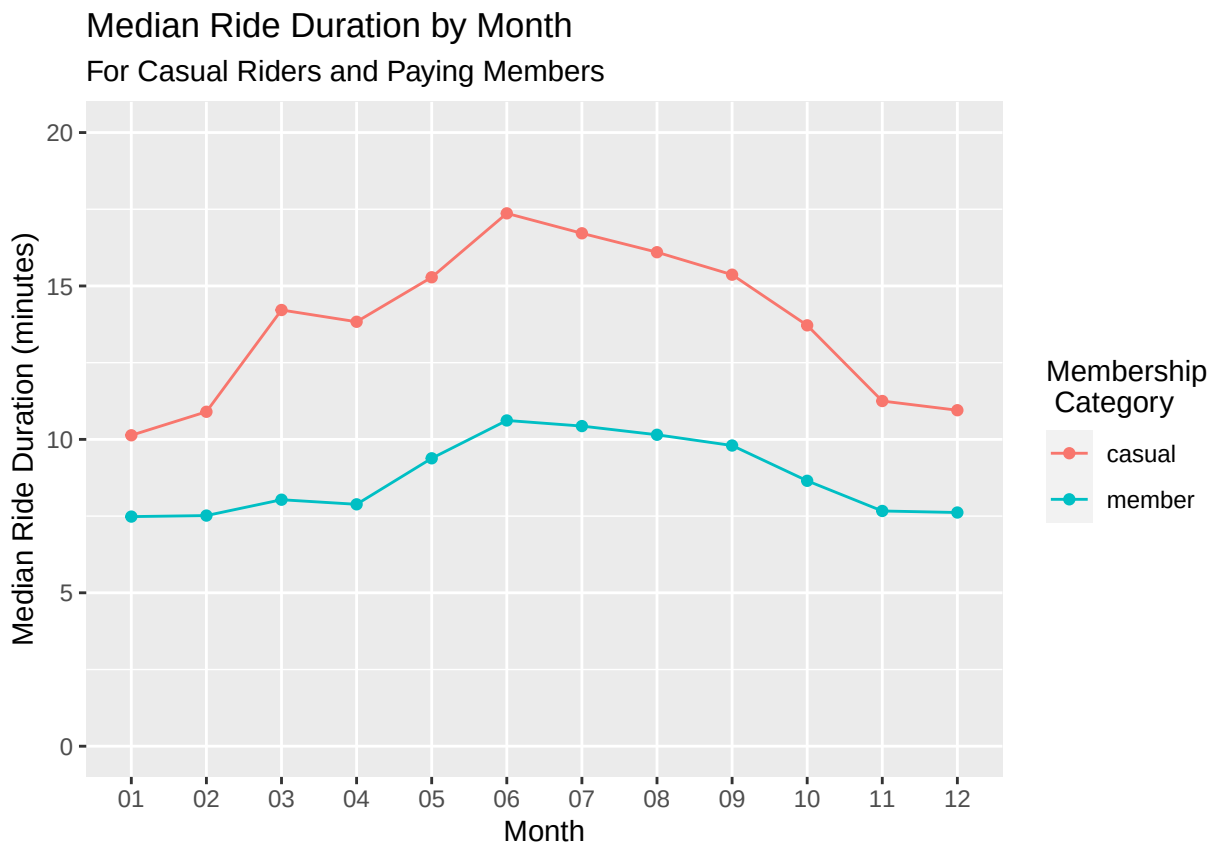
```
tripdata_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = number_of_rides/1000, group = member_casual)) +
  geom_line(aes(color=member_casual)) + geom_point(aes(color=member_casual)) +
  labs(title = "Total Number of Bike Rides Per Month",
       subtitle = "For Casual Riders and Paying Members",
       x = "Month",
       y = "Number of Rides (thousands)") +
  guides(col= guide_legend(title= "Membership \n Category"))
```



I was not surprised that the number of bike rides would have significant seasonal differences in a city like Chicago, which is known for its harsh winters. Casual riders had a greater seasonal difference in the quantity of bike rides than paying members. I found that outside of the summer months, paying members had a greater number of bike rides than casual riders. However, from June through August, the number of rides by casual bikers exceeded the number of rides by paying members.

I also chose to plot the median ride lengths by month for both categories.

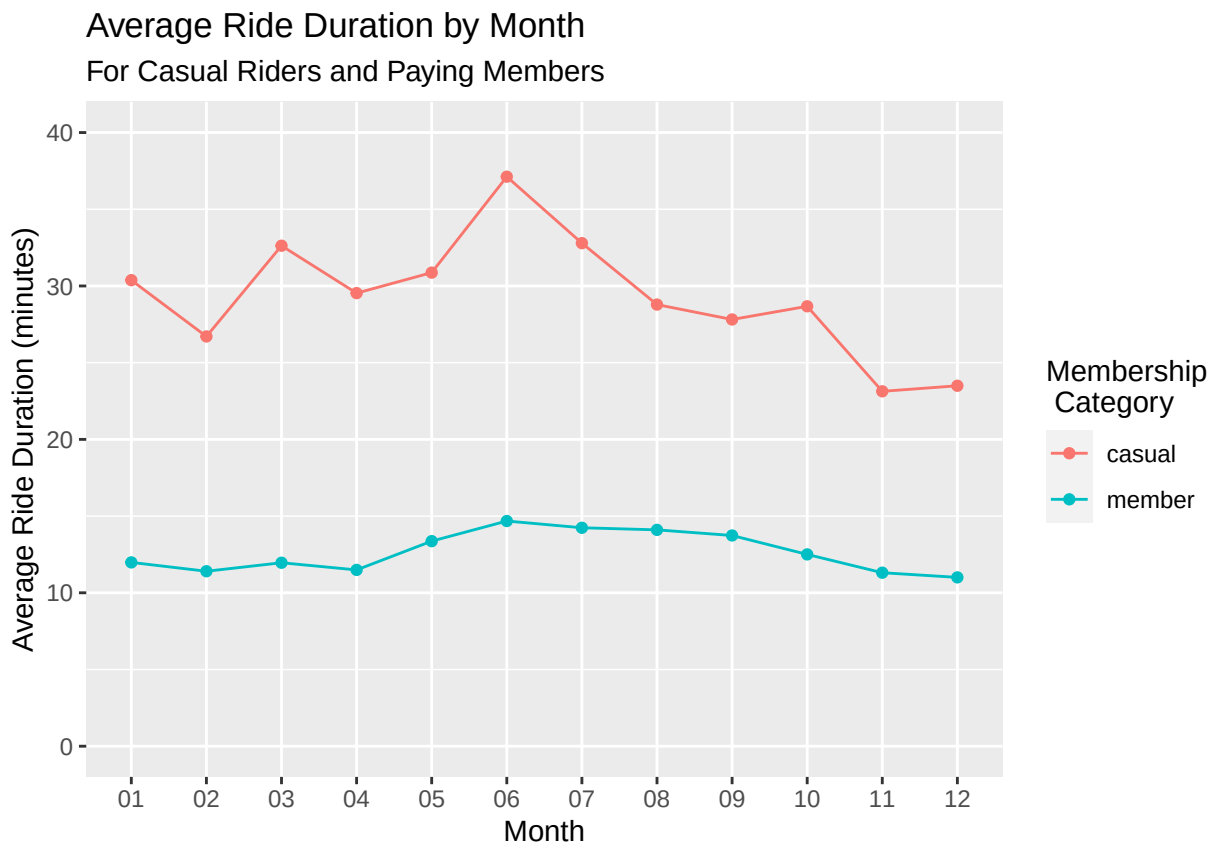
```
tripdata_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(),
            median_duration = median(ride_length)) %>%
  arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = median_duration/60, group = member_casual)) +
  geom_line(aes(color=member_casual)) + geom_point(aes(color=member_casual)) +
  ylim(0,20) + labs(title = "Median Ride Duration by Month",
                    subtitle = "For Casual Riders and Paying Members",
                    x = "Month",
                    y = "Median Ride Duration (minutes)") +
  guides(col= guide_legend(title= "Membership \n Category"))
```



The data showed that regardless of the month, casual riders consistently have higher median ride times than paying members. Not surprisingly, median ride times were higher in the warmer months compared to the colder months.

To confirm that casual riders also had a higher mean ride time throughout the year, I also plotted the mean ride duration by month.

```
tripdata_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = average_duration/60, group = member_casual)) +
  geom_line(aes(color=member_casual)) + geom_point(aes(color=member_casual)) +
  ylim(0,40) + labs(title = "Average Ride Duration by Month",
                    subtitle = "For Casual Riders and Paying Members",
                    x = "Month",
                    y = "Average Ride Duration (minutes)") +
  guides(col= guide_legend(title= "Membership \n Category"))
```



Act

Based on my analysis, I concluded that casual riders present a significant opportunity to expand paid membership for Cyclistic. Casual riders showed greater variability than paying members when it came to the number of rides based on the time of the week or the time of the year. More specifically, casual riders had more rides than members during the weekends and the summer months, but otherwise had fewer rides than paying members. However, regardless of the time of the year or type of bike, casual riders had a higher mean and median bike ride duration. Additionally, casual riders were the only riders who had use docked bikes during the period being evaluated.

Since the goal of Cyclistic is to expand the number of paying members, my recommendations are as follows:

- Create a weekend membership program to encourage weekend casual riders to sign up for membership that is active only on weekends and holidays.
- Invest advertising resources from May through October to reach out to casual riders, and encourage them to sign up for annual memberships.
- Conduct surveys of casual bikers and paid members to figure out their main purposes for their bike rides, including the long-duration docked bike rides that are solely used by casual riders. This additional data will give the company a more refined sense of which types of bikers tend to become members, and how much membership growth can be expected.

I hope that the executive team carefully evaluates my analysis and approves my proposal, because I believe another membership option that can capture the high quantity and duration of casual riders during weekends will improve the long-term profitability of Cyclistic. Also, since the warmer months contain the highest number of casual riders, this period is the greatest and most efficient opportunity to invest marketing

resources for membership growth. Finally, the proposed survey will help refine the company's marketing strategy and growth expectations.

Hopefully with the executive team's approval, my manager, Lily Moreno, will benefit from my analysis and will be able to more effectively promote the growth of Cyclistic.