Google Data Analytics Course Capstone Project: Case Study 1 "Cyclistic"

This is my approach and work to solve the problem of Google Data Analytics Course **Capstone Project: Case Study 1 "Cyclistic"**.
The main objective of this case study is **"How to convert casuals to members?"** or to be specific,
a successful bike-sharing company desires to increase the number of their annual memberships.

As I learned from the Google Data Analytics program, I will follow the steps of the data analysis process: **ask, prepare, process, analyze, share and act**.

Ask

These questions will guide the future marketing program:

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual memberships?
3. How can Cyclistic use digital media to influence casual riders to become members?

Prepare

In this step, we prepare the data by obtaining the dataset and storing it. The datasets are given as a monthly based trip data in a .zip file. I downloaded the last 12 months of trip data i.e. **September 2020 to August 2021** as 12 different .zip files and extracted them. We don't need to mine or scrape the data, its given as a .csv file for each month. The data which is provided is public data that helps us to explore how different customer types are using Cyclistic bikes.

Process

In this step we process the data and prepare it for our next step where we will find answers to our questions. I used RStudio it is **an Integrated Development Environment (IDE) for R, a programming language for statistical computing and graphics**. It is available in two formats: RStudio Desktop is a regular desktop application while RStudio Server runs on a remote server and allows accessing RStudio using a web browser. I used RStudio Desktop for merging and processing all 12 .csv files

At, first I uploaded all the 12 .csv files using **read.csv()** function, it is part of **read.** table in the R utils package (installed by default).

```
data2020_09 <- read.csv(file.choose())
data2020_10 <- read.csv(file.choose())
data2020_11 <- read.csv(file.choose())
data2020_12 <- read.csv(file.choose())
data2021_01 <- read.csv(file.choose())
data2021_02 <- read.csv(file.choose())
data2021_03 <- read.csv(file.choose())
data2021_04 <- read.csv(file.choose())
data2021_05 <- read.csv(file.choose())
data2021_06 <- read.csv(file.choose())
data2021_07 <- read.csv(file.choose())
data2021_08 <- read.csv(file.choose())
```

Then I merged and all .csv files into one large dataset. I merged all the .csv files using **rbind()** function, it is part of the **plyr** package

```
all_data <- rbind(data2020_09, data2020_10,data2020_11,data2020_12,data2021_01,
                  data2021_02,data2021_03,data2021_04,data2021_05,data2021_06,
                  data2021_07,data2021_08)
```

Let's observe the number of rows and columns:

```
> summary(all_data)
   ride_id           rideable_type        started_at       
 Length:4913072     Length:4913072     Length:4913072    
 Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character  
                                                          
                                                          
                                                          
                                                          
   ended_at          start_station_name start_station_id  
 Length:4913072     Length:4913072     Length:4913072    
 Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character  
                                                          
                                                          
                                                          
                                                          
 end_station_name   end_station_id       start_lat       start_lng     
 Length:4913072     Length:4913072     Min.   :41.64   Min.   :-87.84  
 Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66  
 Mode  :character   Mode  :character   Median :41.90   Median :-87.64  
                                       Mean   :41.90   Mean   :-87.65  
                                       3rd Qu.:41.93   3rd Qu.:-87.63  
                                       Max.   :42.08   Max.   :-87.52  
                                                                        
    end_lat         end_lng        member_casual     
 Min.   :41.51   Min.   :-88.07   Length:4913072    
 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.   :-87.44                     
 NA's   :5015    NA's   :5015                       
```

As you see our dataset become very large with nearly **5 million rows** (50 Lakhs) and **13 columns**. Now let's have a peek to the dataset using **head()**:

```
> head(all_data,1)
          ride_id rideable_type       started_at         ended_at    start_station_name start_station_id       end_station_name
1 2B22BD5F95FB2629 electric_bike 17-09-2020 14:27 17-09-2020 14:44 Michigan Ave & Lake St               52 Green St & Randolph St
  end_station_id start_lat start_lng  end_lat   end_lng member_casual
1            112  41.88669 -87.62356 41.88357 -87.64873        casual
```

We have 13 columns and we can look into their content:

- **ride_id**: Id for each trip taken, it may contain duplicate values but we are not sure if they are unique or not, we have to find out
- **rideable_type**: Represents the type of a bike
- **started_at**: Date and time of the start time
- **ended_at**: Date and time of the end time
- **start_station_name**: Name of the starting station
- **start_station_id**: Id of the starting station
- **end_station_name**: Name of the ending station
- **end_station_id**: Id of the ending station
- **start_lat**: Latitude of the starting point
- **start_lng**: Longitude of the starting point
- **end_lat**: Latitude of the ending point
- **end_lng**: Longitude of the ending point
- **member_casual**: Represents the membership status

Then I used **dropDuplicates()** function to remove duplicates rows but the count didn't change and the number of rows did not change, it means that **ride_id** is unique for each row.

Next, I would like to find out the time each trip took. For that, I created a new column named ride_length(in Min) were using the started_at and ended_at columns, for that I separate the date and time which were given to us in a single column for that I used **sapply()** function and save start time as **stime** and end time as **etime.**

```
all_data$stime <- sapply(strsplit(as.character(all_data$started_at), " "),"[",2)
all_data$etime <- sapply(strsplit(as.character(all_data$ended_at)," "),"[",2)
head(all_data)
```

And after separating date and time I used **format()** function to set time in time datatype using **as.POSIXct()** function and then calculate the length of each ride in a minute using **difftime()** function

```
all_data$stime <- format(as.POSIXct(all_data$stime, format = "%H:%M"))
all_data$etime <- format(as.POSIXct(all_data$etime, format = "%H:%M"))

all_data$ride_length <- difftime(all_data$etime, all_data$stime, units = "mins")
```

After creating ride_length I notice that some of the values are negative and these rows clearly indicate wrong input and should be removed from the dataset for that I used **filter()** function with respect to **ride_length** column.

Next, I create a column day which will represent the day of the trip. To do this I used **weekdays()** function which is available in RStudio

```
demo_data$day <- weekdays(as.Date(demo_data$started_at))
head(demo_data)
```

Then I removed the unnecessary column from our dataset which was not required to us. So now our data is going to look like this.So I only **select ridabale_type**, **start_station_name**, **end_station_name**, **member_casual**, **ride_length** and **day** columns from our original dataset.

```
  rideable_type           start_station_name         end_station_name member_casual
1 electric_bike        Michigan Ave & Lake St    Green St & Randolph St        casual
2 electric_bike Ashland Ave & Belle Plaine Ave          Montrose Harbor        casual
3 electric_bike       Fairbanks Ct & Grand Ave Fairbanks Ct & Grand Ave        casual
4 electric_bike         Clark St & Armitage Ave                                casual
5 electric_bike       Wells St & Evergreen Ave    Broadway & Sheridan Rd        casual
6 electric_bike                                                                casual
  ride_length       day
1     17 mins Wednesday
2     25 mins Wednesday
3     36 mins Wednesday
4     46 mins Wednesday
5     14 mins   Tuesday
6     25 mins Wednesday
```

If you notice in the previous image some of our columns still contain blank data (i.e row no. 4 and 6). So first I replace that blank data with **NA.** let's look at this data now

```
  rideable_type            start_station_name         end_station_name member_casual
1 electric_bike         Michigan Ave & Lake St   Green St & Randolph St        casual
2 electric_bike Ashland Ave & Belle Plaine Ave          Montrose Harbor        casual
3 electric_bike        Fairbanks Ct & Grand Ave Fairbanks Ct & Grand Ave        casual
4 electric_bike         Clark St & Armitage Ave                     <NA>        casual
5 electric_bike        Wells St & Evergreen Ave   Broadway & Sheridan Rd        casual
6 electric_bike                            <NA>                     <NA>        casual
  ride_length       day
1     17 mins Wednesday
2     25 mins Wednesday
3     36 mins Wednesday
4     46 mins Wednesday
5     14 mins   Tuesday
6     25 mins Wednesday
```

Then I removed the rows which contain **NA** in our data for that I used na.omit() function which is available in R. So now our data is going to look like this

```
  rideable_type                 start_station_name          end_station_name
1 electric_bike             Michigan Ave & Lake St    Green St & Randolph St
2 electric_bike     Ashland Ave & Belle Plaine Ave           Montrose Harbor
3 electric_bike           Fairbanks Ct & Grand Ave  Fairbanks Ct & Grand Ave
5 electric_bike           Wells St & Evergreen Ave    Broadway & Sheridan Rd
8 electric_bike Mies van der Rohe Way & Chestnut St W Oakdale Ave & N Broadway
9 electric_bike                Halsted St & Polk St     Emerald Ave & 31st St
  member_casual ride_length       day
1        casual     17 mins Wednesday
2        casual     25 mins Wednesday
3        casual     36 mins Wednesday
5        casual     14 mins   Tuesday
8        casual     25 mins   Tuesday
9        casual     17 mins   Tuesday
```

Now let's check the number of rows and columns are there in our dataset now that we filter the data. When we started we have around 5 million rows now we have around 4 million-plus rows.

```
> nrow(blank_replace)
[1] 4165163
> ncol(blank_replace)
[1] 6
```

Now let's observe the distribution for some categorical columns:

I used **group_by()** function with **summarise()** function and inside **summarise()** I used **count()** function to show the frequency distributions. Since we already removed null so we can see only casual member count and member count.

```
  count(member_casual).x count(member_casual).freq
1                 casual                   1850300
2                 member                   2314863
```

So according to the frequency of casual and actual members, we can observe that there are more member than casual.

Now I compare how many minutes casual member use the bike with respect to an actual member for that I used **group_by()** function with **summarise()** function and inside **summarise()** I used **tally()** function to show the this data

```
  member_casual n
  <chr>         <drtn>
1 casual        52481547 mins
2 member        32055344 mins
```

According to data, we were able to found out that casual member used bicycles for more minutes of time as compare to actual member.

Also, the busiest day of the week is Monday followed by Sunday. And we can see from how many minutes bike has been used on each day of the week so we can get a clear idea.

	count(day).x	count(day).freq
1	Monday	627243
2	Tuesday	597650
3	Wednesday	571795
4	Thursday	592173
5	Friday	601423
6	Saturday	563268
7	Sunday	611611

	day	n
	<fct>	<drtn>
1	Monday	12930635 mins
2	Tuesday	12288182 mins
3	Wednesday	11378768 mins
4	Thursday	11996163 mins
5	Friday	12194576 mins
6	Saturday	11037343 mins
7	Sunday	12711224 mins

Furthermore, there 3 types of bikes available such as classic bike, electric bike and docked bike. Where classic bike is the most popular one among the three.

	count(rideable_type).x	count(rideable_type).freq
1	classic_bike	2250098
2	docked_bike	1021736
3	electric_bike	893329

And we can see from the following data, how many minutes each bike has been ridden.

1	classic_bike	42034846 mins
2	docked_bike	27604577 mins
3	electric_bike	14897468 mins

We will do further detailed analysis in the next step, I found the frequency distribution for each **start_station_name**. I'm able to show only the first six rows of the data because it contains so many rows

	count(start_station_name).x	count(start_station_name).freq
1	2112 W Peterson Ave	875
2	351	1
3	63rd St Beach	2244
4	900 W Harrison St	5699
5	Aberdeen St & Jackson Blvd	10262
6	Aberdeen St & Monroe St	9422

Now, in the Process step, we clean (i.e. we remove the rows which contain blank values) and filter the data and try to understand data as well as try to find insights about the data. And then take only columns which are necessary to us for the next step, which is the Analyze step

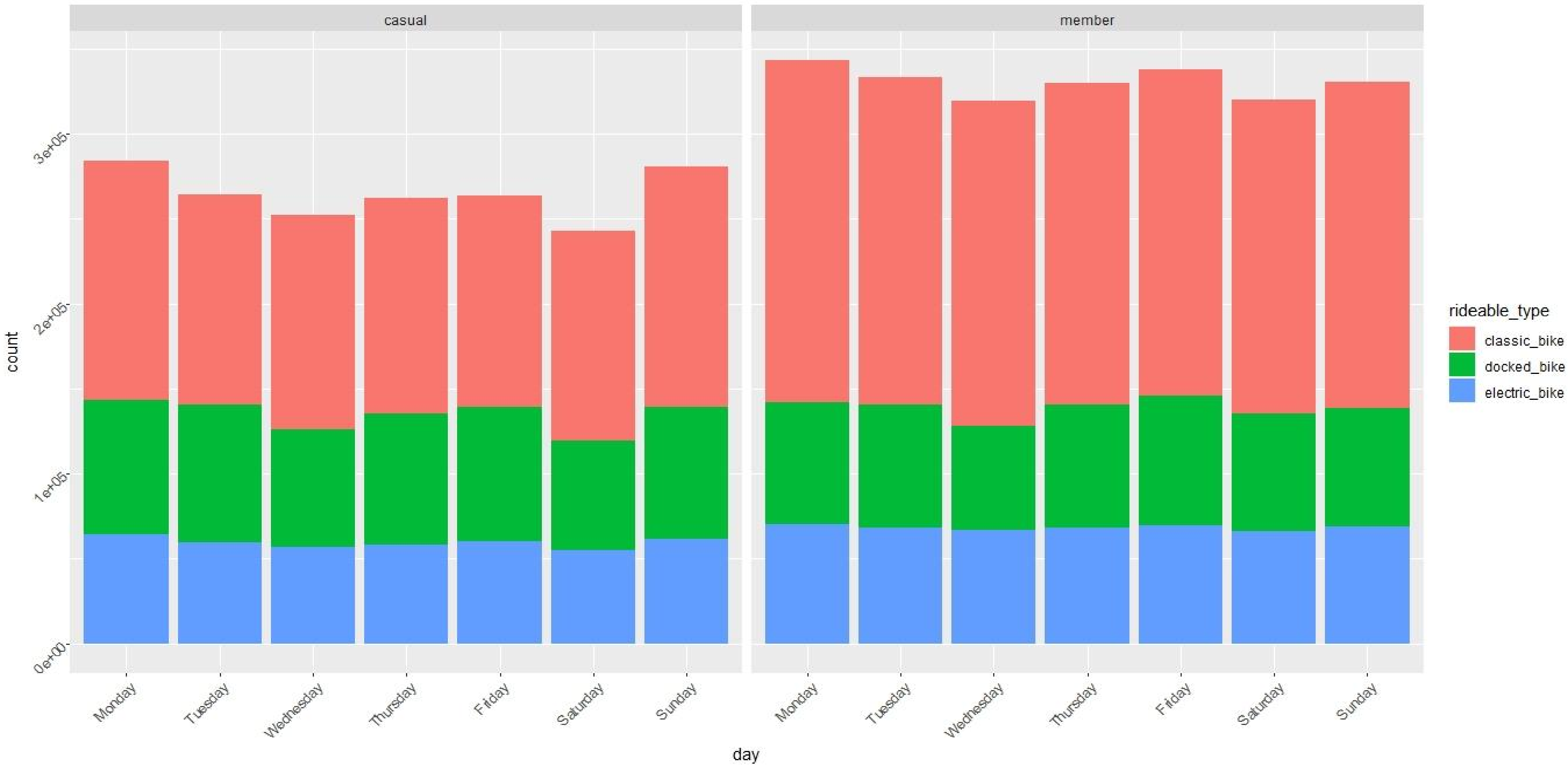
Analyze

In this step, we will analyze our processed and cleaned data. As we already cleaned our data we don't need to do that again.

Now, let's plot a bar graph that shows the weekly frequency distribution of the member and casual customers with bike types. For this, I organized days in order from Monday to Sunday. Then applied **geom_bar** and **fill** with **rideable_type**.

```
blank_replace$day <- factor(blank_replace$day, levels = c("Monday","Tuesday", "Wednesday",
                                                          "Thursday","Friday","Saturday",
                                                          "Sunday"))
```

```
ggplot(blank_replace) + geom_bar(aes(x = day, fill = rideable_type))  +
  facet_wrap(~member_casual) +
  theme(axis.text = element_text(angle = 45, hjust = 1))
```


casual
member
count
3e+05
2e+05
1e+05
0e+00
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
day
rideable_type
classic_bike
docked_bike
electric_bike

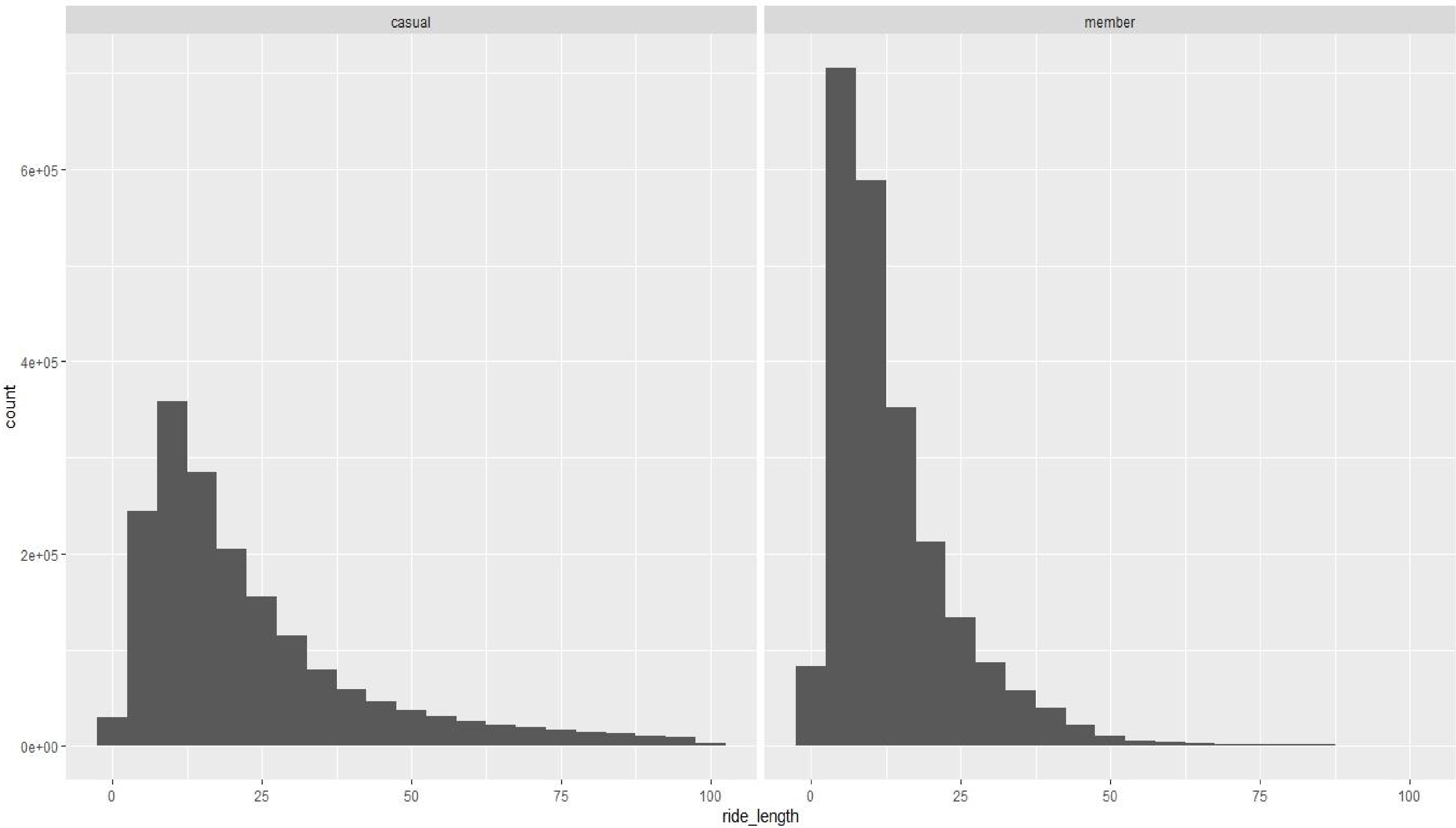
This plot shows us few observations about members and casuals. Some of them are:

- Members usage are quite similar throughout the week. It is very hard to conclude anything from this plot.
- Casual usage is slow for weekdays(i.e from Tuesday to Saturday) but on Monday and Sunday are above average.
- Classic bike is the most popular for both members and casuals. Followed by Docked bike

Now let's observe trip duration behavior for member and casuals. For this I used **geom_histogram** and filtered the duration times to less than 100 minutes for the better plot:

```
ggplot(filter(blank_replace, blank_replace$ride_length<100)) +
  geom_histogram(aes(x = ride_length), binwidth = 5) +
  facet_wrap(~member_casual)
```

.


casual
member
count
6e+05
4e+05
2e+05
0e+00
0
25
50
75
100
0
25
50
75
100
ride_length

The only observation here is that members tend to take short trips than casuals, So we can assume that member are Students or Workers who used to ride bicycles to reach their college or workplace. Or casuals take longer trips than members. We will talk about the mean trip duration later using summary function.

Next, I filtered dataset into two, according to member-casual status. Then applied summary function to numeric columns only to get some details. Below is the summary for members dataset.

```
member_only <- blank_replace %>%
  filter(member_casual == "member")

summary(select(member_only, c('day','ride_length')))
```

```
       day            ride_length
 Monday   :343067   Length:2314863
 Tuesday  :333526   Class :difftime
 Wednesday:319606   Mode  :numeric
 Thursday :329747
 Friday   :337850
 Saturday :320263
 Sunday   :330804
```

```
> min(member_only$ride_length)
Time difference of 1 mins
> mean(member_only$ride_length)
Time difference of 13.84762 mins
> median(member_only$ride_length)
Time difference of 10 mins
> max(member_only$ride_length)
Time difference of 1249 mins
```

Now, I did the same for casuals:

```
casual_only <- blank_replace %>%
  filter(member_casual == "casual")

summary(select(casual_only, c('day','ride_length')))
```

```
       day            ride_length
Monday   :284176   Length:1850300
Tuesday  :264124   Class :difftime
Wednesday:252189   Mode  :numeric
Thursday :262426
Friday   :263573
Saturday :243005
Sunday   :280807
```

```
> min(casual_only$ride_length)
Time difference of 1 mins
> mean(casual_only$ride_length)
Time difference of 28.3638 mins
> median(casual_only$ride_length)
Time difference of 18 mins
> max(casual_only$ride_length)
Time difference of 1362 mins
```

From the above summary, we can observe that members mean trip duration ~14 min. is almost twice less than casual mean trip duration ~28 min.

Next, let's see the most popular start and end station with their frequency for member. For that, First I filter member from member_casual column and then count the frequency of each station and sort it in descending order and from that I able to find 10 most populer start and end sation for member :

```
M <- blank_replace %>%
  filter(blank_replace$member_casual == "member")
```

```
M1 <- count(M$start_station_name)
most_freq_start_station <- M1[order(-M1$freq),]
head(most_freq_start_station,10)
```

```
                          x  freq
         Clark St & Elm St 23013
     Wells St & Concord Ln 20348
Kingsbury St & Kinzie St 19665
         Wells St & Elm St 18514
      Dearborn St & Erie St 17581
     St. Clair St & Erie St 17236
       Wells St & Huron St 17051
      Broadway & Barry Ave 17042
       Theater on the Lake 16244
  Clark St & Armitage Ave 15742
```

Now, let's see the most popular end station with their frequency for member:

```
P1 <- count(M$end_station_name)
most_freq_end_station <- P1[order(-P1$freq),]
head(most_freq_end_station,10)
```

```
                          x  freq
         Clark St & Elm St 23398
     Wells St & Concord Ln 20959
Kingsbury St & Kinzie St 20060
         Wells St & Elm St 18718
      Dearborn St & Erie St 18130
     St. Clair St & Erie St 17683
      Broadway & Barry Ave 17371
       Wells St & Huron St 16333
       Theater on the Lake 15194
  Clark St & Armitage Ave 14872
```

Let's apply the same steps for casual dataset as well, Lets look at start station frequency frist:

```
N <- blank_replace %>%
  filter(blank_replace$member_casual == "casual")
head(N)
```

```
N2 <- count(N$start_station_name)
most_freq_start_station <- N2[order(-N2$freq),]
head(most_freq_start_station,10)
```

```
                           x  freq
    Streeter Dr & Grand Ave 54476
            Millennium Park 29427
        Michigan Ave & Oak St 26451
  Lake Shore Dr & Monroe St 25955
         Theater on the Lake 21051
              Shedd Aquarium 20362
Lake Shore Dr & North Blvd 17891
       Wells St & Concord Ln 16322
Indiana Ave & Roosevelt Rd 15720
      Clark St & Lincoln Ave 14865
```

Now, let's see the most popular end station with their frequency for casual member:

```
N1 <- count(N$end_station_name)
most_freq_end_station <- N1[order(-N1$freq),]
head(most_freq_end_station,10)
```

```
                           x  freq
    Streeter Dr & Grand Ave 57413
            Millennium Park 31200
        Michigan Ave & Oak St 27892
  Lake Shore Dr & Monroe St 24871
         Theater on the Lake 23079
Lake Shore Dr & North Blvd 21184
              Shedd Aquarium 18237
       Wells St & Concord Ln 16362
Indiana Ave & Roosevelt Rd 15813
      Clark St & Lincoln Ave 15352
```

As you see from above results, casuals tend to start and end trips from the same station while its little different for members.

We have done quite a lot of observations above. Next, I would summarize them into one table using data.table and formattable packages of R. It is little pain to fill the table manually, but I think the result is worth it because everything becomes easier to understand.

```
data_table <- data.frame("User_type" = c("Member", "Casual"),
                         "Amount" = c("2,314,863 (55.6%)", "1,850,300 (44.4%)"),
                         "Avg_and_median_trip_duration" = c("13.80 min - 10 min",
                                                            "28.40 min - 18 min"),
                         "Busiest_day" = c("Monday", "Monday"),
                         "Preffered_bike_type" = c("Classic Bike", "Classic_Bike"))

formattable(data_table,
            align = c("l","c","c","c","c"),
            list("User_type" = formatter("span", style = ~style(color = "gray",
                                                             font.weight = "bold"))))
```

User_type	Amount	Avg_and_median_trip_duration	Busiest_day	Preffered_bike_type
Member	2,314,863 (55.6%)	13.80 min - 10 min	Monday	Classic Bike
Casual	1,850,300 (44.4%)	28.40 min - 18 min	Monday	Classic_Bike

Share

After tons of codes and analysis, it's time to share our results and to answer the question "How can we convert casuals to members?".

We can't fully answer to this question and come up with a solution. Because the data given to us only shows one instance of each unique bike users. The best dataset we require is the instances of a user as casual and after becoming a member. Analyzing those observations, we could find some trend or pattern for users to convert from casual to members.

However, we still have some observations and inferences from our analysis that it's possible to come up with a possible solution. Although, it might not be effective fully. Now, let's summarize what we have observed from our analysis:

- **Member bike usage is quite similar throughout the week. We can conclude that members are mostly working people or students that getting a membership is financially and time wise viable option.**
- **Casual usage is slow for weekdays(i.e from Tuesday to Saturday) but on Monday and Sunday are above average.**
- **Classic bike is the most popular for both members and casuals. Followed by Docked bike**
- **Members mean trip duration ~14 min. is almost twice less than casual mean trip duration ~28 min.**
- **Casual users tend to start and end trips from the same station while its little different for members.**
- **Most lengthy trips are taken by casuals and they are abnormally long. For instance, top five lengthy trips are 38, 37, 36, 35, 35 <u>days</u> all taken by casuals.**
- **All occurrence of the missing data of bike type, start and end station names, member type, are around 800k.**

Considering the above observations and insights we can suggest the following:

We see that members take shorter trips to work with bikes during Monday to Sunday, since it is financially viable and fast transportation. However, casuals prefer longer trips especially on Monday and Sunday. Thus:

1. We could increase the renting price of the bikes for the weekend to target casual users into having a membership.
2. Providing a special service or perks for only members might motivate casual users to have a membership. Services might include free tour guide, or fast line for renting without any line, or if member able to convert casual member to become member, then can provide benefit to both in the form of addons to

continue membership, benefit to member who mange to ride bike given amount of time in particular week, same for the month etc.

Also, since we know the most popular start station names and routes for casual users, we can put banners or special discount advertisements in those areas or routes that would target casual users.

Act

However, since act step is for executives to decide, So I didn’t focus on this step here.