Google Capstone Project - Cyclistic Bike-share

Subhash H Jayanna

1/3/2022

Introduction

Welcome to the Cyclistic bike-share analysis case study! This is a capstone case studies conducted for my Google Data Analytics Professional Certificate studies. In this case study, I will be assuming the role of a data analyst at a fictional bike-share company, “Cyclistic”. This bike sharing company is located in Chicago. The director of marketing, Lily Moreno, believes the company’s future success depends on maximizing the number of annual memberships. Although this company is fictional, this is real data collected between August 2020 – July 2021 from a bike share program in Chicago. In order to provide my recommendations to my manager, I will be analyzing the data which has been made available by Motivate International Inc under this license (https://www.divvybikes.com/data-license-agreement).

Scenario

Cyclistic bike-share fictional company features over 5,800 bicycles and 600 docking stations. Cyclistic’s unique selling point is their assisting riding options offering which is used by 8% of their riders. Cyclistic’s finance analysts have concluded that annual members are much more profitable than casual riders. The director of marketing believes the company’s future success depends on maximizing the number of annual memberships. My team is responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy. My team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. I will be attempting to answer the question “How do annual members and casual riders use Cyclistic bikes differently?” in this case study.

A. Ask Phase

Business Task

Improving profitability by increasing annual-membership Riders, the most profitable user type for Cyclistic bike-share.

Key Stakeholders

This project includes Lily Moreno, the director of marketing and my manager, who is responsible for the development of campaigns and initiatives to promote the bike-share program.

I joined Cyclistic Marketing Analytics team about six months ago. I together with a team of analysts will be analyzing and reporting data that helps guide Cyclistic marketing strategy.

Cyclistic executive team will take a decision whether to approve the program recommended by the marketing team.

B. Prepare Phase

Data Files Location

The data (https://divvy-tripdata.s3.amazonaws.com/index.html) has been made available by Motivate International Inc. under this license (https://www.divvybikes.com/data-license-agreement). This is public data that you can use to explore how different customer types are using Cyclistic bikes.

Data Organization

The data is stored in .csv format. Trip data is divided into months and I will be considering a full year worth of trip data from December 2020 till November 2021 for this case study.

Reliability and Credibility of the data

Cyclistic's historical monthly trip data consists of all trips made by all users in a single month. The data has been made available by Motivate International Inc. This is public data that anyone can use to explore how different customer types are using Cyclistic bikes. The City of Chicago makes the new data available to the public every month.

Liscencing, Privacy, Security and Accessbility

This data The data has been swiped to remove any personally identifiable information (PII) as data-privacy issues prohibit us from using riders' PII. Data can be accessed by anyone in the public from an s3.awscloud server using the link provided in the "Data Files Location" section above.

Data Relevancy

Data contains an important attribute, "member_casual" which describes the type of rider. The riders who have annual membership are marked "members" and other riders are marked "casual". Using the data from this column, we can analyze the Trip Duration data versus Day of Month, Month of Year, Day of Week, Hour of Day, to identify contrasting patterns between member riders and casual riders.

Issues and Problems with the data

There are some issues that exist in the data set. There are several trips with zero ride time which can easily be cleaned using R's tidyverse package. Cordinate data columns contain NULL values which will be removed in the Process Phase of the project. The column "rideable_type" has 3 unique entries, "classic_bike", "electric_bike", and "docked_bike". The first two types are self-explanatory but the "docked_bike" type of ambiguous.

C. Process Phase

Tools and Tech used in this case study

I will analyzing one year worth of trip data which includes roughly over 5 million rows of data for 12 month. Since the size of the data is not suitable to work on using Spreadsheets software and RStudio Cloud service, I will be using RStudio Desktop and an R markdown file to compile this case study.

Load the required packages in R

Ensuring Data Integrity

Importing previous 12 months trip data (from December 2020 till November 2021)

Run the code snippet below to load data from all 12 months.

Combining 12 data.frames into a single data.frame

Making sure combining all data frames includes all records

Since data binding was done using rbind, we can compare row count of “all_trips_v1” data frame with sum of rows of all 12 months of trip data.

```
print(sum(nrow(df1), nrow(df2), nrow(df3), nrow(df4), nrow(df5), nrow(df6), nrow(df7), n
row(df8), nrow(df9), nrow(df10), nrow(df11), nrow(df12)))
```

```
## [1] 5479096
```

Now let’s see the total number of rows in the combined data frame.

```
print(nrow(all_trips_v1))
```

```
## [1] 5479096
```

Since row count matches between “all_trips_v1” data frame and sum of all rows in all files, it is safe to declare data frame combining task is a success.

Dealing with duplicate and missing records

Checking for Duplicate Values.

Let us see the number of distinct “ride_id” entries in the data set where each unique ride_id corresponds to a unique trip.

As we see can see, the total number of rows in the data set “all_trips_v1” remains the same after running the distinct() function on it, so we can safely say that there aren’t any duplicate “ride_id” entries.

Checking for NULL Values in all Columns

From colSums(is.na()) function, we can see that the columns “start_station_name”, “start_station_id”, “end_station_name”, “end_station_id”, “end_lat”, “end_lang” all contain missing values(NA).

```
colSums(is.na(all_trips_v1))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##             651445             651442             698909             698909
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0               4738               4738
##      member_casual
##                  0
```

We can clean the data by dropping all records with missing values. Let us run this code to drop all NAs from our data set.

To validate that all NA values are removed from the data set, we can run this code again:

```
colSums(is.na(all_trips_v2))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0                  0                  0
##      member_casual
##                  0
```

This confirms that the data is free of missing values.

Cleansing and Preparing the Data

By running glimpse() function we can get a list of all column headers with their data type next to it.

```
glimpse(all_trips_v2)
```

```
## Rows: 4,525,842
## Columns: 13
## $ ride_id            <chr> "70B6A9A437D4C30D", "15F369FDAED4E8E3", "0CFD61DFE0…
## $ rideable_type      <chr> "classic_bike", "electric_bike", "electric_bike", "…
## $ started_at         <dttm> 2020-12-27 12:44:29, 2020-12-18 13:53:56, 2020-12-…
## $ ended_at           <dttm> 2020-12-27 12:55:06, 2020-12-18 14:01:46, 2020-12-…
## $ start_station_name <chr> "Aberdeen St & Jackson Blvd", "Larrabee St & Armita…
## $ start_station_id   <chr> "13157", "TA1309000006", "KA1503000043", "TA1309000…
## $ end_station_name   <chr> "Desplaines St & Kinzie St", "Wells St & Walton St"…
## $ end_station_id     <chr> "TA1306000003", "TA1306000011", "TA1306000003", "TA…
## $ start_lat          <dbl> 41.87773, 41.91811, 41.88919, 41.96710, 41.88132, 4…
## $ start_lng          <dbl> -87.65479, -87.64380, -87.63858, -87.66743, -87.629…
## $ end_lat            <dbl> 41.88872, 41.90013, 41.88910, 41.96710, 41.88918, 4…
## $ end_lng            <dbl> -87.64445, -87.63445, -87.64248, -87.66743, -87.638…
## $ member_casual      <chr> "member", "member", "member", "casual", "member", "…
```

As per the output, columns “started_at” and “ended_at” both are of date/time data types. Since trip duration is granulated by time of the day, in order to improve running descriptive statistics capabilities on the data set, we should break down the Date/Time columns to Date, Year, Month, Day of Week, Hour of Day. By running the code snippet below, new columns will be added to the “all_trips_v2” data set.

It is very beneficial for us to have another column to denote the length of each ride. Since our data set doesn’t contain that information, we can calculate the duration of each trip by subtracting “ended_at” with “started_at” columns.

To verify whether these new columns are added to the data set, run the str()

```
str(all_trips_v2)
```

```
## tibble [4,525,842 × 20] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:4525842] "70B6A9A437D4C30D" "15F369FDAED4E8E3" "0CFD61D
FE00E6043" "244CB936487039B7" ...
##  $ rideable_type     : chr [1:4525842] "classic_bike" "electric_bike" "electric_bike"
"docked_bike" ...
##  $ started_at        : POSIXct[1:4525842], format: "2020-12-27 12:44:29" "2020-12-18
13:53:56" ...
##  $ ended_at          : POSIXct[1:4525842], format: "2020-12-27 12:55:06" "2020-12-18
14:01:46" ...
##  $ start_station_name: chr [1:4525842] "Aberdeen St & Jackson Blvd" "Larrabee St & Ar
mitage Ave" "Kingsbury St & Kinzie St" "Clark St & Leland Ave" ...
##  $ start_station_id  : chr [1:4525842] "13157" "TA1309000006" "KA1503000043" "TA13090
00014" ...
##  $ end_station_name  : chr [1:4525842] "Desplaines St & Kinzie St" "Wells St & Walton
St" "Desplaines St & Kinzie St" "Clark St & Leland Ave" ...
##  $ end_station_id    : chr [1:4525842] "TA1306000003" "TA1306000011" "TA1306000003"
"TA1309000014" ...
##  $ start_lat         : num [1:4525842] 41.9 41.9 41.9 42 41.9 ...
##  $ start_lng         : num [1:4525842] -87.7 -87.6 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num [1:4525842] 41.9 41.9 41.9 42 41.9 ...
##  $ end_lng           : num [1:4525842] -87.6 -87.6 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr [1:4525842] "member" "member" "member" "casual" ...
##  $ date              : Date[1:4525842], format: "2020-12-27" "2020-12-18" ...
##  $ month             : chr [1:4525842] "12" "12" "12" "12" ...
##  $ day               : chr [1:4525842] "27" "18" "28" "10" ...
##  $ year              : chr [1:4525842] "2020" "2020" "2020" "2020" ...
##  $ day_of_week       : Ord.factor w/ 7 levels "Sunday"<"Monday"<..: 1 6 2 5 1 5 4 6 1
7 ...
##  $ hour_of_day       : chr [1:4525842] "12" "13" "17" "13" ...
##  $ trip_duration     : num [1:4525842] 10.62 7.83 1.8 60.78 6.07 ...
```

From a quick glance we can see that the newly created column, “trip_duration” contains negative trip duration time. Negative time is certainly not possible and we should remove all such occurrences to make the data set more cohesive.

```
all_trips_v3 <- all_trips_v2[!(all_trips_v2$trip_duration <= 0),]
nrow(all_trips_v3)
```

```
## [1] 4525274
```

Run the below code snippet to count the total number of trips with zero or negative trip duration.

```
zero_trip_duration <- all_trips_v2[(all_trips_v2$trip_duration <= 0),]
nrow(zero_trip_duration)
```

```
## [1] 568
```

Presuming trips with less than one minute as false starts, let us identify and remove all trips with trip duration less than 1 minute(60secs).

Run the below code snippet to count the total number of trips with trip duration less than one minute.

```
trip_duration_less_than_1minute <- all_trips_v3[all_trips_v3$trip_duration < 1,]
nrow(trip_duration_less_than_1minute)
```

```
## [1] 57973
```

Also assuming that trips lasting more than 24hours are marked as stolen equipment, let us remove all trips where trip duration is more than 24 hours (1,440 minutes)

Running the code below code snippet to count the total number of trips with trip duration more than 24 hours (1,440 minutes).

```
trips_more_than_24hr <- all_trips_v4[( all_trips_v4$trip_duration > 1440),]
nrow(trips_more_than_24hr)
```

```
## [1] 1263
```

In order to make sure whether all of the filters are applied to the data set, run the code snippet below:

```
summary(all_trips_v5$trip_duration)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    1.000    7.183   12.483   20.280   22.483 1439.367
```

As we can see, the shortest trip is 1 minute and the longest trip is 1439.367 minutes. So we can safely conclude that our cleaning effort on the trip duration column is a success.

This concludes the cleaning phase for this case analysis. There are a total of 568 trips with zero or negative trip duration, 58,595 trips with trip duration in between 0 minutes and 1 minute and 1263 trips where the trip duration has exceeded 24hours. We initially started with 5,479,096 rows of and after all the cleaning, we are left with 4,466,038 rows of clean data ready for analysis.

D. Analyze Phase

From a glance at the data set it is clear that the attributes rideable_type, member_casual, day_of_Week, hour_of _day, month, and trip_duration are useful in answering the question at hand, How do annual members and casual riders use Cyclistic bikes differently?

```
head(all_trips_v5)
```

```
## # A tibble: 6 × 20
##   ride_id rideable_type started_at          ended_at            start_station_n…
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 70B6A9… classic_bike  2020-12-27 12:44:29 2020-12-27 12:55:06 Aberdeen St & J…
## 2 15F369… electric_bike 2020-12-18 13:53:56 2020-12-18 14:01:46 Larrabee St & A…
## 3 0CFD61… electric_bike 2020-12-28 17:10:25 2020-12-28 17:12:13 Kingsbury St & …
## 4 244CB9… docked_bike   2020-12-10 13:36:16 2020-12-10 14:37:03 Clark St & Lela…
## 5 B7AD50… classic_bike  2020-12-20 13:09:04 2020-12-20 13:15:08 Dearborn St & M…
## 6 E60629… docked_bike   2020-12-03 21:06:25 2020-12-03 21:43:18 Sheridan Rd & N…
## # … with 15 more variables: start_station_id <chr>, end_station_name <chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>, date <date>, month <chr>, day <chr>,
## #   year <chr>, day_of_week <ord>, hour_of_day <chr>, trip_duration <dbl>
```

Let us consider “trip_duration” column to see how the values are spread out. By running summary() function on trip_duration column, we can see that average trip duration for all the rides is 20.280 minutes. The average trip duration for casual riders is more than double to that of member riders.

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    1.000    7.183   12.483   20.280   22.483 1439.367
```

```
## # A tibble: 2 × 2
##   member_casual average_duration
##   <chr>                   <dbl>
## 1 casual                   28.6
## 2 member                   13.5
```

I have used adorn_* along with tabyl() function to cross-tab rideable_type and member_casual. By running the code snippet below we can clearly see that “docked_bike” is the least used rideable_type by member riders and casual riders with only 7.1% of total rides done on docked bikes. The highest used bike type is the “classic_bike” type with 71% of total rides done on classic bikes. Member riders have completed 2,459,717 rides (55.1% of total rides) where as casual riders have completed 2,006,321 rides (44.9% of total rides). Another important inference from the cross-tab is that the casual riders, out of a total of 2 million rides, 61.9% of the rides (1,240,021) done on “classic_bike” rideable_type and the least rideable_type used by casual riders is the “docked_bike” type with only 15.3% of total rides (308,958).

rideable_type	casual	member	Total
classic_bike	27.8% (1240021)	43.2% (1930772)	71.0% (3170793)
docked_bike	6.9% (308958)	0.2% (7669)	7.1% (316627)
electric_bike	10.2% (457342)	11.7% (521276)	21.9% (978618)
Total	44.9% (2006321)	55.1% (2459717)	100.0% (4466038)

The average trip duration times for docked_bike type is the highest with 49.5 minutes followed by classic_bike type and electric_bike type with 18.7 minutes and 16.1 minutes respectively.

```
## # A tibble: 3 × 2
##   rideable_type average_duration
##   <chr>                    <dbl>
## 1 classic_bike              18.7
## 2 docked_bike               49.5
## 3 electric_bike             16.1
```

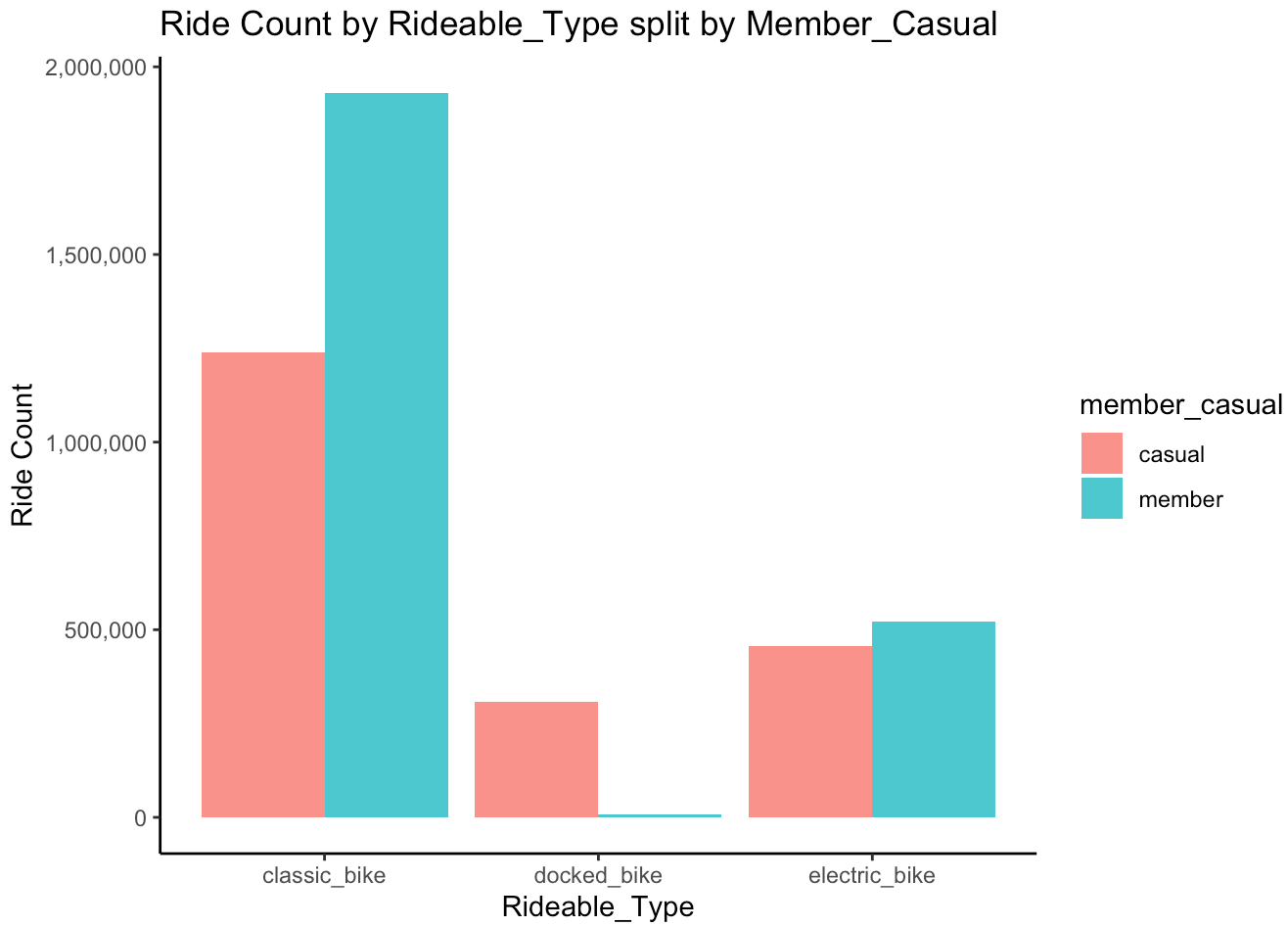
The cross-tab between member_casual and day_of_week comparing average trip_duration denotes that highest average trip_duration times are coming in on Saturdays by casual riders. The lowest average trip duration times are coming in on Fridays by member riders. Similarly, casual riders are most active on Saturday with number of rides going up to almost 460,000 and least active on Friday with number of rides summing up to just above 281,000.

```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.gro
ups` argument.
```

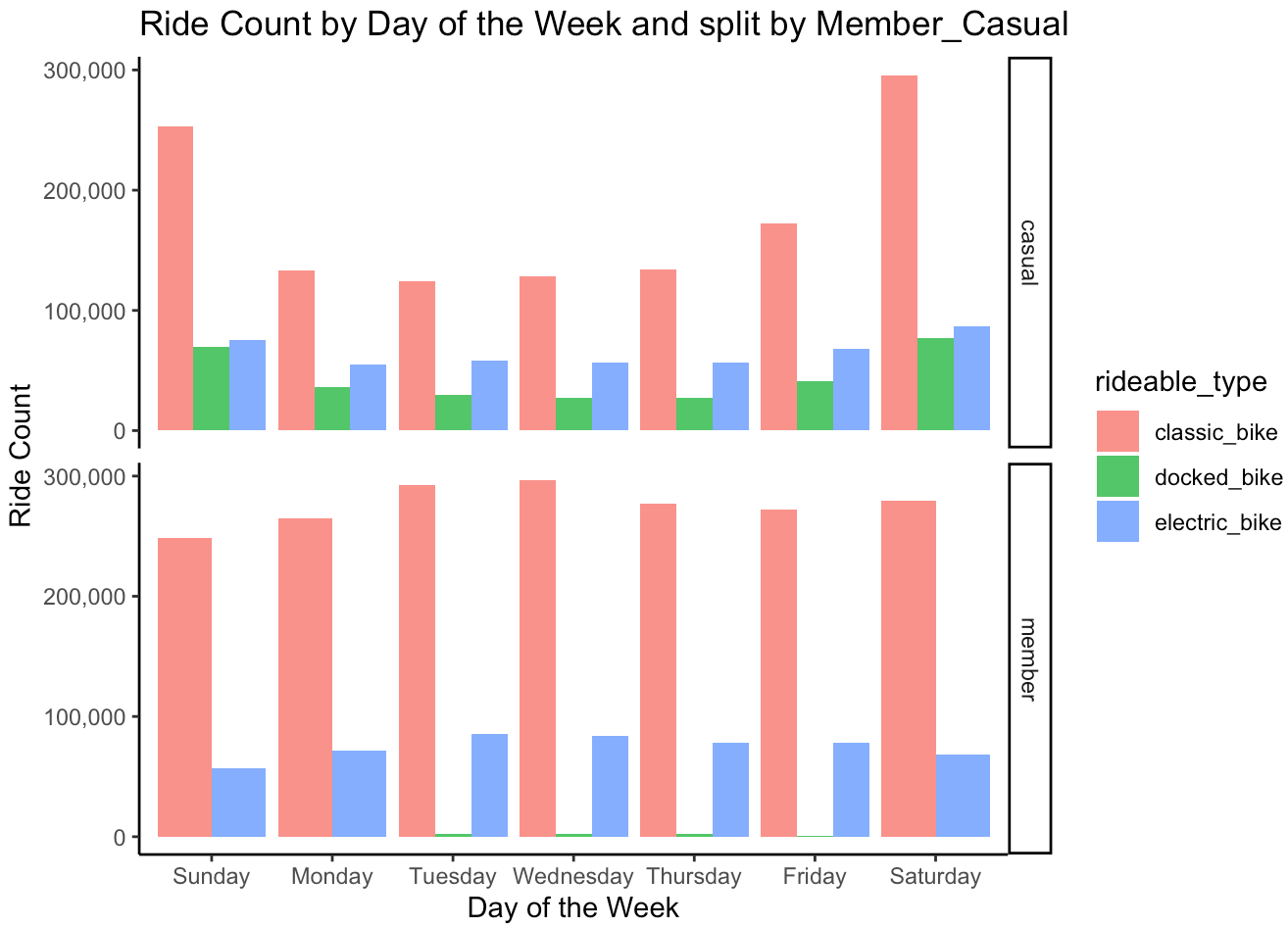
```
## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides average_duration
##    <chr>         <ord>                 <int>            <dbl>
##  1 casual        Saturday             459744             30.9
##  2 casual        Sunday               398055             32.8
##  3 member        Wednesday            382414             12.8
##  4 member        Tuesday              380937             12.6
##  5 member        Thursday             356995             12.6
##  6 member        Friday               349965             13.1
##  7 member        Saturday             347742             15.1
##  8 member        Monday               336445             13.0
##  9 member        Sunday               305219             15.5
## 10 casual        Friday               281531             26.6
## 11 casual        Monday               224390             29.1
## 12 casual        Thursday             217641             24.5
## 13 casual        Wednesday            213002             24.8
## 14 casual        Tuesday              211958             26.2
```

E. Share Phase

From the below viz. we can infer that both members and casual riders prefer classic_bike type. Whereas docked_bike type is the lowest used bike type by both types of riders but casual riders prefer docked bike way more than member riders.

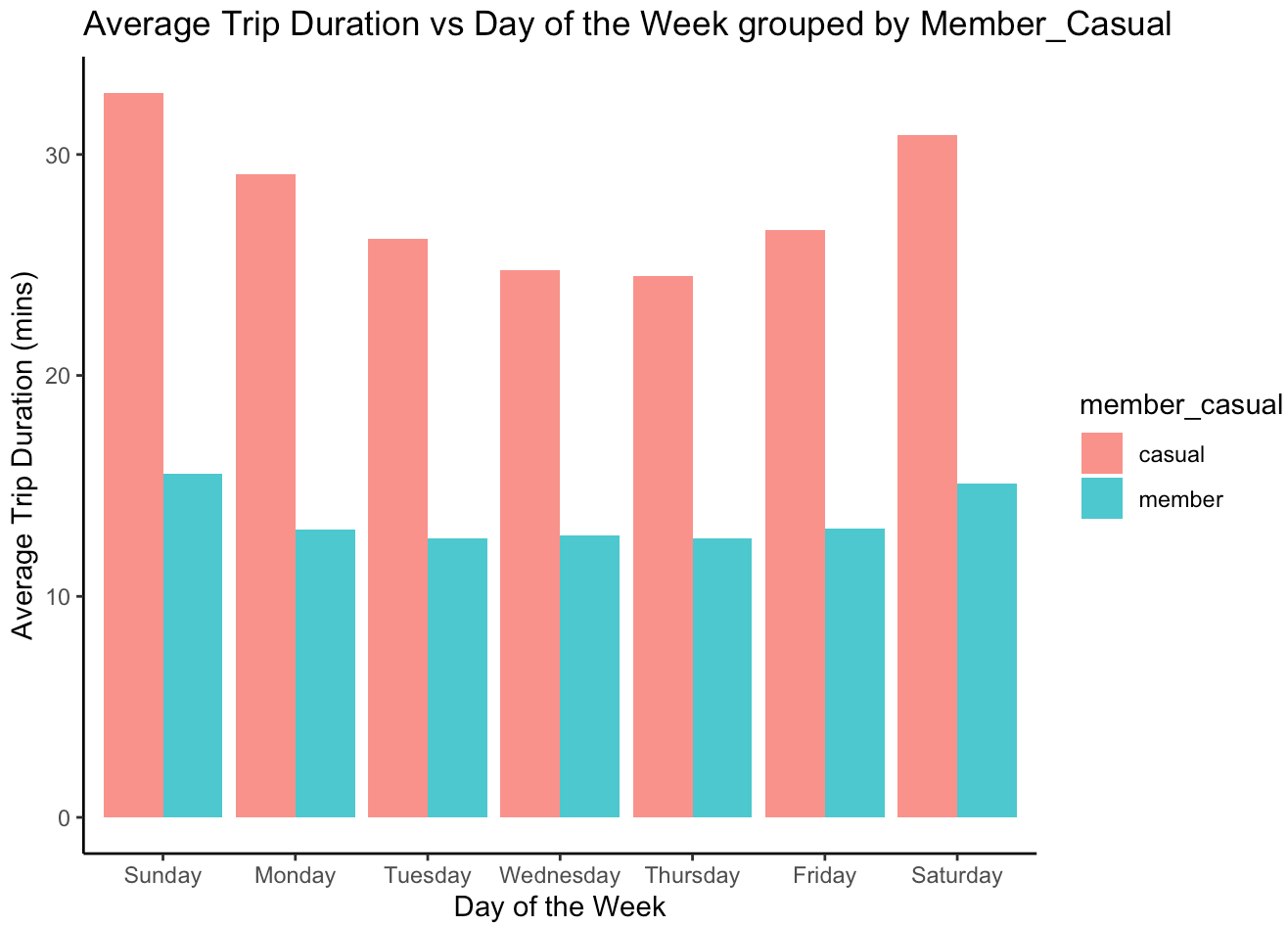


The viz. below segments number of rides by day of the week split by rider type. It is clear that casual riders are most active during the days on weekends with highest number of rides coming on Saturday followed by Sunday and Friday respectively. Whereas member riders are most active on the days member riders are least active i.e. on Tuesday, Wednesday and Thursday.

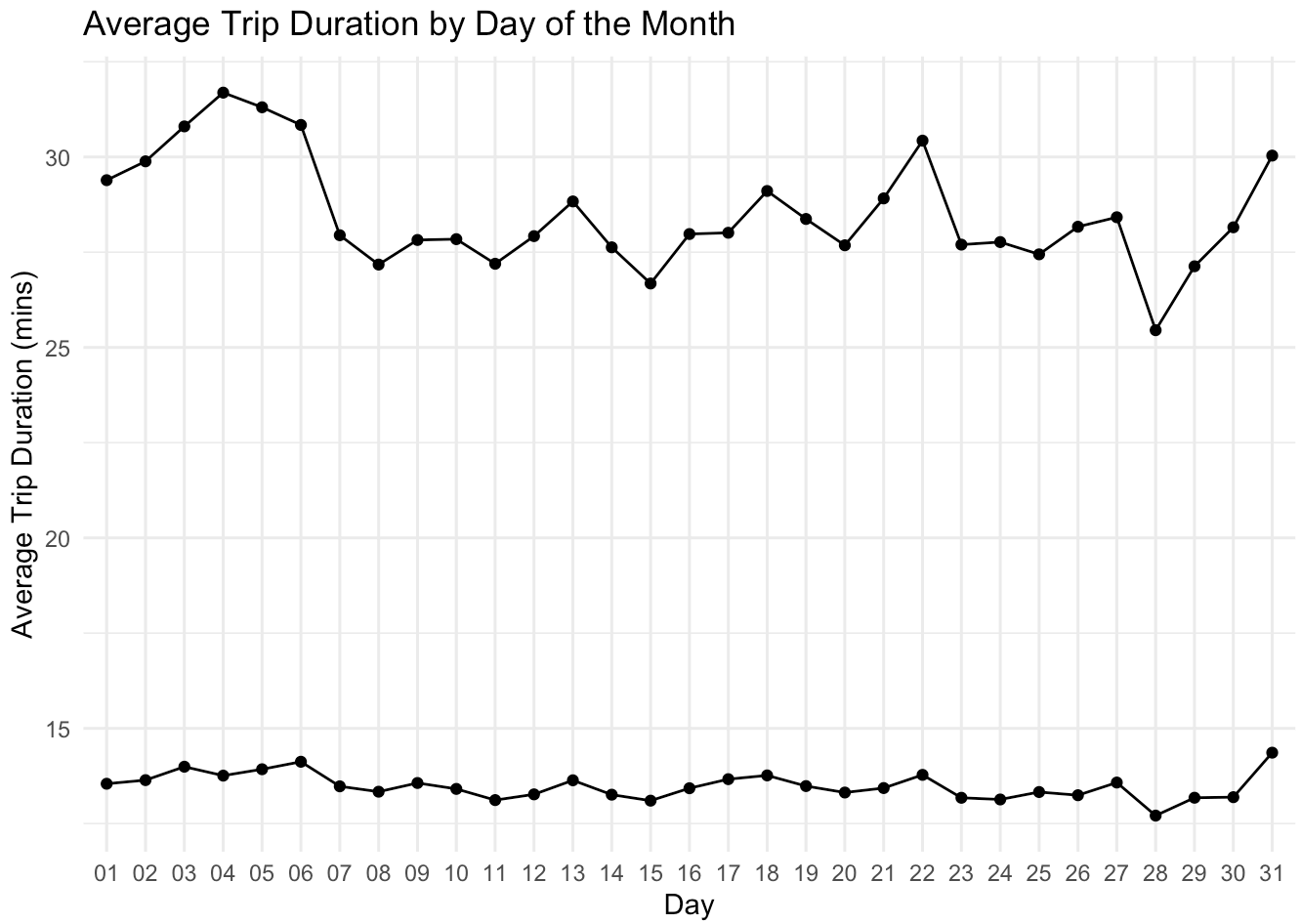


The viz. below compares average trip times across the days of the week. The average trip duration for casual riders is higher than member riders on all days of the week. So, we can conclude that casual riders use Cyclistic bike-share services for longer time than member riders.

```
## `summarise()` has grouped output by 'day_of_week'. You can override using the `.group
s` argument.
```

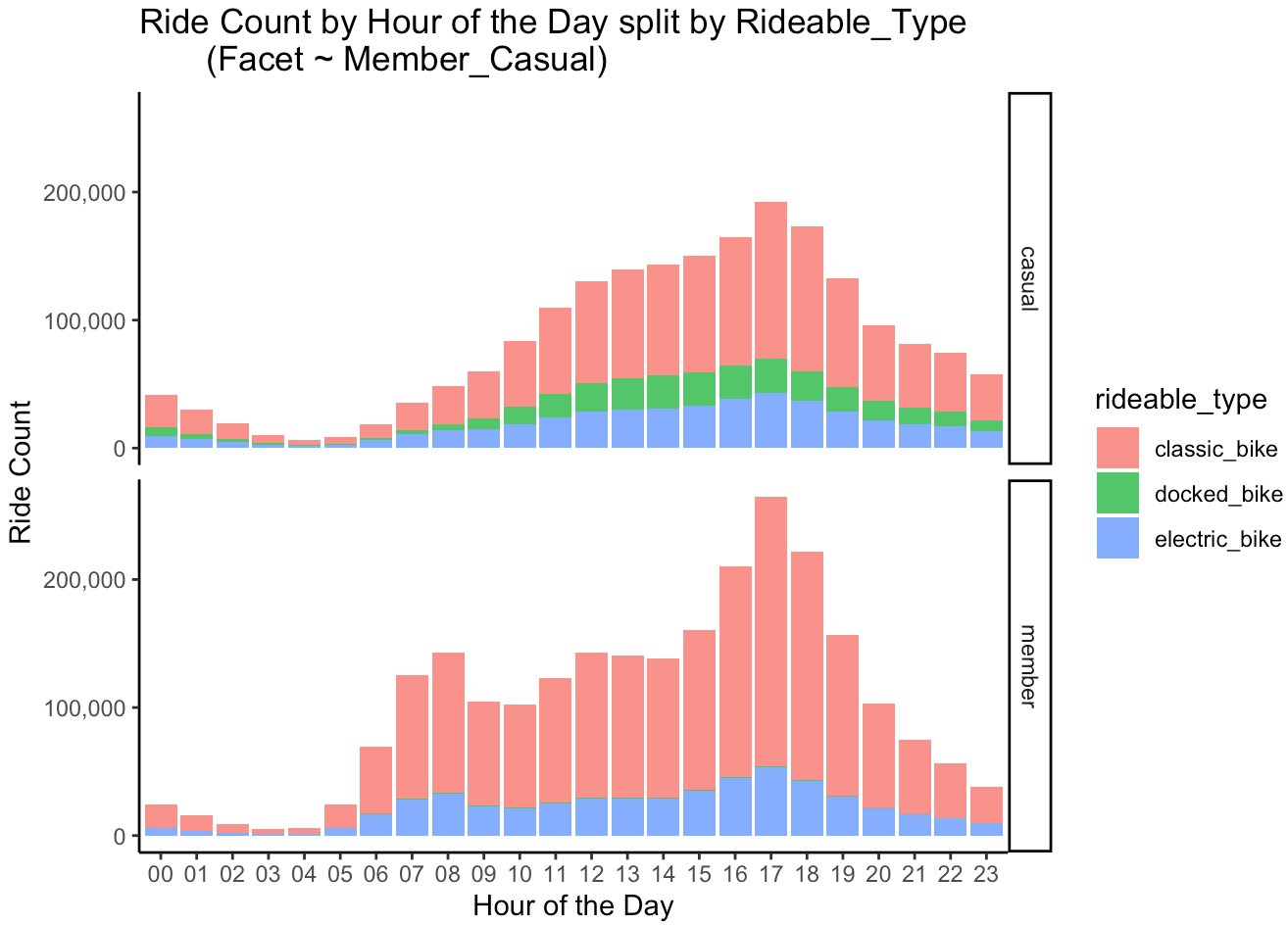


The plot below shows average trip duration spread over all the days of a month grouped by rider type. Lowest average trip duration for rides done by casual members is just above 25 minutes

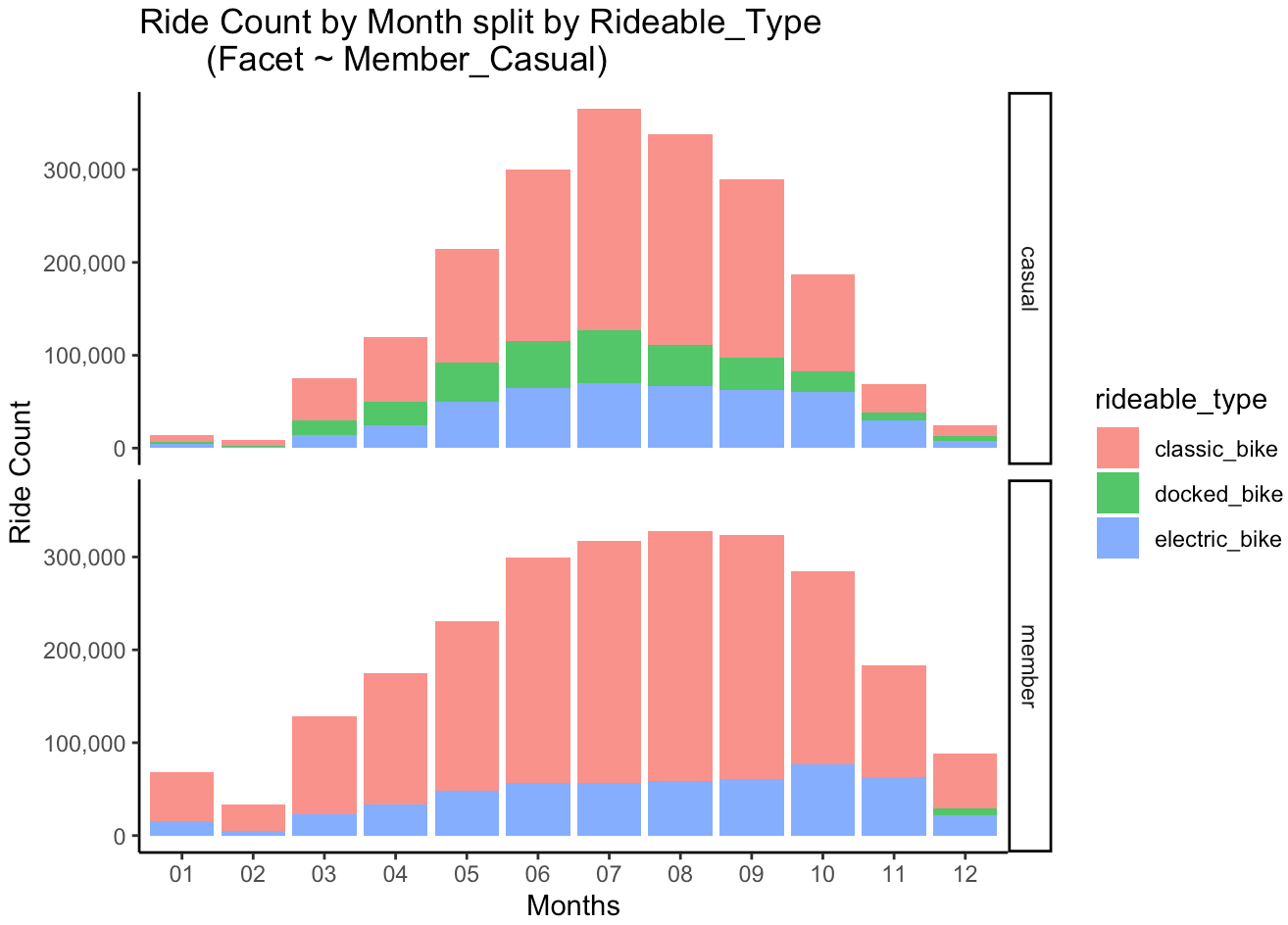


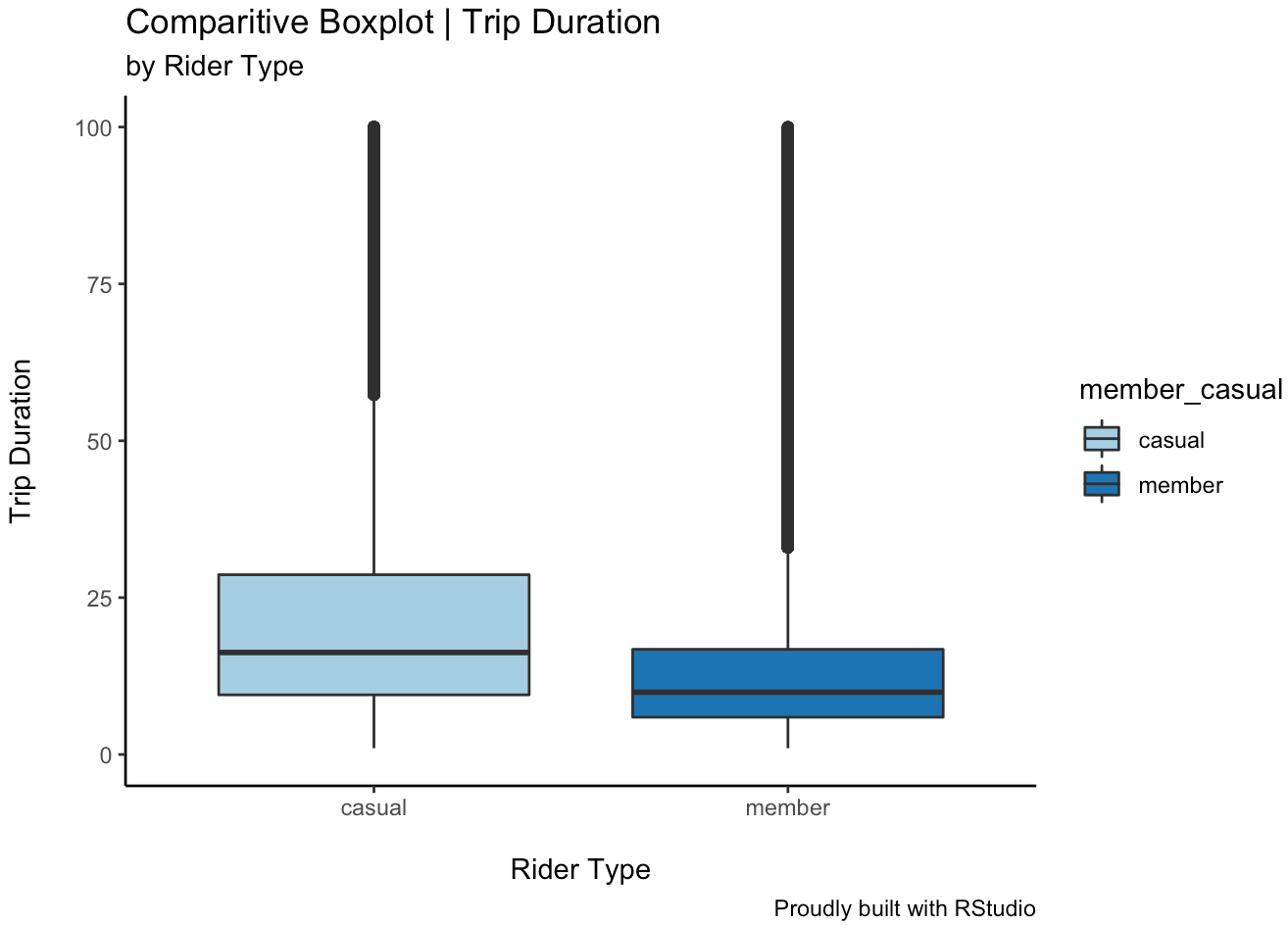
The viz. below groups number of rides by hour of the day faceted by member_casual. We can infer that there's a steady increase in the number of rides as the day goes by with 5 PM, 6 PM and 4 PM being the most active times of the day in that order for both rider types. But one distinction is that member riders when compared to casual riders, prefer to use bike-share services during the morning hours of the day, i.e 8 AM , 7 AM and 9 AM as well as evening hours.

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



The viz. below groups number of rides by Months faceted by member_casual. Both rider types are most active during the middle months of the year. Casual Riders are most active in July and Member riders are most active in August but both rider types are least active in February. Also, another inference can be made from the plot below would be that the docked_bike type is most used by casual riders in the summer months of the year. But, since we do not have an explanation regarding what constitutes as a docked bike no this insight will carry less weight.





Proudly built with RStudio

F. Act Phase

Based on my analysis, member riders have more overall trips done compared to casual riders but the later on average ride twice as long compared to member riders. Both rider types are most active during June, July, August and September months and least active during December, January and February. Number of trips done by casual riders gradually increase through out the day peaking at 5 PM and gradually tapers off as day comes to an end. Whereas member riders have a large spike in ride count at 5 PM and a short spike at 8 AM. This strongly suggests that member riders mostly use bike-sharing services during the start and end hours of a typical workday. Casual riders tend to take significantly more number of rides which are longer on average during the weekends. On the contrary, member riders' usage is concentrated on the weekdays especially during the middle the of week. Bike rider types tend to prefer Classic bikes over the other two bike types but casual riders use more significantly number of docked bikes than member riders. Given these insights, I would like to provide my top 3 recommendations, to the director of marketing and my manager, Lily Moreno.

1. Casual riders on an average ride for twice as longer compared to member riders so my first recommendation would be to introduce rewards programs to the existing annual subscription model to incentivize longer rides. A Majority of the rides done by casual riders is about 30 minutes in length, a recommendation would be to add a rewards program with the annual membership which incentivizes rides longer than 30 minutes.
2. Casual riders ride a lot more during the weekends compared to the weekdays. Given this insight my recommendation to the marketing department should run the promotion campaigns online during the weekdays leading up-to busier weekend days.
3. Along the same lines, casual riders use bike-share services the most during the summer months. My recommendation would be to set up billboards and other static poster advertisements in and around the hot-spots across the city. This would increase exposure for the annual membership program and Cyclistic bike-service in general.

Further analysis can be done using the Geo-coordinates from the data set to identify hot spot across the city which has higher number of casual riders. By doing that billboard advertisements can be set-up strategically across the city to draw more exposure. Geo location data can also be used to improve bike availability around the hot-spots.

If we were to have address data attached to the payment information, it would help us to split the casual riders group into natives and tourists. By identifying riders native to the city, it will allow the marketing folks to target such individual who have higher probability to sign up for annual membership when compared to tourists.