CyclisticCaseStudy

Madhu Santhanam

21/08/2021

CYCLISTIC DATA ANALYSIS

Cyclistic Data Analysis to determine how annual members and casual riders use the Cyclistic bike share program differently

Executive summary:

Cyclistic is a bike share program with more than 5,800 bicylces of three kinds and over 600 docking stations in Chicago. There are two types of riders - members and casual. Members have annual membership with Cyclistic whereas casual riders either get a day pass or single-ride pass. The company's financial analysts have concluded that annual members are much more profitable than casual riders.The director of marketing wants her team to analyze the data and tell her how to increase the membership.In order to do that we need to understand how the two rider types differ in using the bike share program , why would casual riders buy memberships and how can the company use digital media to influence casual riders to become members.

Business task: The Ask

The director of Marketing, my manager, wants me, a junior data analyst in her team, to analyse and answer the question: how do annual members and casual riders use the Cyclistic bikes differently?

Prepare:

In order to complete the analysis and to answer the question, I will use the past 12 months' ridership data from the data source: https://divvy-tripdata.s3.amazonaws.com/index.html. As the data is a first party data, the data source is deemed trust worthy. The data is in comma separated format.The data includes unique ride_ID for each ride, date and time the rides started and ended, starting and ending station name, id, latitude and longitude. The data does not contain any personally identifiable information about the riders.

There are some nulls/blanks in the data which we will ignore while doing our analysis as they are less than 0.001% of the total data.

Process

I downloaded the data into one of my folders and converted them to .xlsx files. I removed any rows that had a start_at time later than the end_at time as they are the data from Cyclistic employees checking the bikes for quality and not actual ride data. As this is a large amount of data of around 4.45 million observations, I will be using R to complete the analysis.

First I begin by installing the required packages (tidyverse and readxl) necessary for the analysis and setting up the working directory.

```
install.packages("tidyverse",repos = "http://cran.us.r-project.org")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
tinytex::install_tinytex()
```

```
## The directory /usr/local/bin is not writable. I recommend that you make it writable. See https://gitl
```

```
setwd("~/Cyclistic data")
```

Importing the 12 monthly data in xlsx into R

Data for files from the month of July 2020 to June 2021 were imported into R for analysis

```
jan21 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

feb21 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl


mar21 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

apr21 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

may21 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl
```

```
jun21 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

jul20 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

aug20 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

sep20 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

oct20 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

nov20 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl

dec20 <- read_excel("~/Library/Mobile Documents/com~apple~CloudDocs/Google Data Analytics/CleanData Cycl
```

Checking the data to see if all the files have the same data types for all the variables before combining the 12 files.Cleaning the files to have the same data types.

```
nov20 <- mutate(nov20, start_station_id = as.character(start_station_id), end_station_id =as.character(e
str(nov20)

oct20 <- mutate(oct20, start_station_id =as.character(start_station_id), end_station_id = as.character(e
str(oct20)

sep20 <- mutate(sep20, start_station_id = as.character(start_station_id), end_station_id = as.character
str(sep20)

aug20 <- mutate(aug20, start_station_id =as.character(start_station_id), end_station_id =as.character(en
str(aug20)

jul20 <- mutate(jul20, start_station_id = as.character(start_station_id), end_station_id = as.character
jul20 <- mutate(jul20, started_at = as.character(started_at), ended_at = as.character(ended_at), ride_le
str(jul20)
```

Combining the 12 monthly files into one data frame to get the data for one year.

```
one_year_trip_data <- bind_rows(jul20, aug20, sep20, oct20, nov20, dec20, jan21, feb21, mar21, apr21, ma
str(one_year_trip_data)
head(one_year_trip_data)
```

Checking the combined one year data to see the data types, column names etc.

```
colnames(one_year_trip_data)
```

```
##  [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"      "ride_length"        "day_of_week"
```

```
dim(one_year_trip_data)
```

```
## [1] 4450281      15
```

Checking the data types in the combined data frame using str() function.

Checking to see how many different values the member_casual column and the rideable_type columns have. In other words, we are checking to see if there are only two rider types and three bike types.

```
table(one_year_trip_data$member_casual)
```

```
##
##  casual  member
## 1926311 2523970
```

```
table(one_year_trip_data$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##       1278596       2040548       1131137
```

Converting the started_at and ended_at columns from string to date using lubridate ymd_hms

```
one_year_trip_data <- mutate(one_year_trip_data, started_at = ymd_hms(started_at), ended_at = ymd_hms(en
```

```
## Warning: 14 failed to parse.
```

```
## Warning: 26 failed to parse.
```

A quick check to see if the changes have happened with str() function.

```
str(one_year_trip_data)
```

```
## tibble [4,450,281 x 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:4450281] "762198876D69004D" "BEC9C9FBA0D4CF1B" "D2FD8EA432C77EC1" "54A
##  $ rideable_type     : chr [1:4450281] "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:4450281], format: "2020-07-09 15:22:02" "2020-07-24 23:56:30" ...
##  $ ended_at          : POSIXct[1:4450281], format: "2020-07-09 15:25:52" "2020-07-25 00:20:17" ...
##  $ start_station_name: chr [1:4450281] "Ritchie Ct & Banks St" "Halsted St & Roscoe St" "Lake Shore D
##  $ start_station_id  : chr [1:4450281] "180" "299" "329" "181" ...
##  $ end_station_name  : chr [1:4450281] "Wells St & Evergreen Ave" "Broadway & Ridge Ave" "Clark St &
##  $ end_station_id    : chr [1:4450281] "291" "461" "156" "94" ...
##  $ start_lat         : num [1:4450281] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:4450281] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:4450281] 41.9 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:4450281] -87.6 -87.7 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:4450281] "member" "member" "casual" "casual" ...
##  $ ride_length       : chr [1:4450281] "1899-12-31 00:03:50" "1899-12-31 00:23:47" "1899-12-31 00:07
##  $ day_of_week       : num [1:4450281] 5 6 4 6 7 3 5 2 5 2 ...
```

Removing the column ride_length and day_of_week that I previously added to the .xlsx file, because I want to calculate it within R; Renaming the resulting data frame to final_data

```
final_data <- one_year_trip_data %>% select(-c(ride_length, day_of_week))
str(final_data)
```

Adding columns that list the date, month, day, and year of each ride (start of the ride). This will allow us to aggregate ride data for each month, day, or year. If we don't complete these operations we could only aggregate at the ride level.

```
final_data$date <- as.Date(final_data$started_at) #The default format is yyyy-mm-dd
final_data$month <- format(as.Date(final_data$date), "%m")
final_data$day <- format(as.Date(final_data$date), "%d")
final_data$year <- format(as.Date(final_data$date), "%Y")
final_data$day_of_week <- format(as.Date(final_data$date), "%A")
```

Quick check to see if the new columns have been added and if the contents look right using colnames()...

```
colnames(final_data)
```

```
##  [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"      "date"               "month"
## [16] "day"                "year"               "day_of_week"
```

...and str() functions.

```
str(final_data)
```

Adding a "ride_length"column to calculate the difference between between starting and ending time to final_data (in seconds).

```
final_data$ride_length <- difftime(final_data$ended_at,final_data$started_at)
colnames(final_data)
```

```
##  [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"      "date"               "month"
## [16] "day"                "year"               "day_of_week"
## [19] "ride_length"
```

Converting the ride_length to numeric so that we can perform calculations.

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

Calculating the mean, max,min of ride_length in general for both membership types.

```
## [1] 1577.787
```

rm.na = TRUE is used to tell r to ignore na values in the calculations.

Cleaning the data to remove any residual negative ride_length values and renaming the resulting data frame to clean_data.

```
clean_data <- final_data[!(final_data$ride_length<0),]
```

Summarizing the ride_length to find out the min, max, median, mean etc using summary().

```
summary(clean_data$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0     453     820    1578    1511 3356649      40
```

Comparing the mean, median, maximum and minimum ride lengths for members and casual users.

```
##   clean_data$member_casual clean_data$ride_length
## 1                   casual               2454.252
## 2                   member                908.872
```

```
##   clean_data$member_casual clean_data$ride_length
## 1                   casual                   1155
## 2                   member                    648
```

```
##   clean_data$member_casual clean_data$ride_length
## 1                   casual                3356649
## 2                   member                2005282
```

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

Ordering the days of the week in the right order as they are not in the right order in the above output.

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

Checking the average ride time by each day for members vs casual users, in the right day of the week order.

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

```
##    clean_data$member_casual clean_data$day_of_week clean_data$ride_length
## 1                    casual                 Sunday              2820.9588
## 2                    member                 Sunday              1026.8455
## 3                    casual                 Monday              2375.1082
## 4                    member                 Monday               875.7175
## 5                    casual                Tuesday              2171.9933
## 6                    member                Tuesday               857.2856
## 7                    casual              Wednesday              2215.7134
## 8                    member              Wednesday               864.2032
## 9                    casual               Thursday              2221.2814
## 10                   member               Thursday               855.0411
## 11                   casual                 Friday              2351.0734
## 12                   member                 Friday               897.7143
## 13                   casual               Saturday              2612.9661
## 14                   member               Saturday               996.5146
```

We will visualize the above results shortly below, so that it will make more sense and will be easy to understand.

Analyze

Analysis of the one year Cyclistic data to see how the members and casual riders use the ride share services differently.

Analyzing the ridership data by weekday and rider type: the below pipe creates a weekday field using wday() from started_at date, and then groups it by rider type, weekday and then, summarizes the number of rides and the average duration and then orders them by rider type first and then weekday.

```
clean_data %>%
  filter(!is.na(started_at)) %>%
  mutate(weekday = wday(started_at, label =TRUE)) %>%
  group_by(member_casual,weekday) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(desc(member_casual), weekday)
```

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

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 member        Sun              322522            1027.
##  2 member        Mon              336809             876.
##  3 member        Tue              363607             857.
##  4 member        Wed              388299             864.
##  5 member        Thu              363081             855.
##  6 member        Fri              373843             898.
##  7 member        Sat              375797             997.
##  8 casual        Sun              366048            2821.
##  9 casual        Mon              209136            2375.
## 10 casual        Tue              203933            2172.
## 11 casual        Wed              215331            2216.
## 12 casual        Thu              212285            2221.
## 13 casual        Fri              280647            2351.
## 14 casual        Sat              438901            2613.
```

Analyzing the data to see how many total number of rides per rider type.

```
clean_data %>%
  filter(!is.na(member_casual)) %>%
  group_by(member_casual) %>%
  summarize(total_rides = n(), average_duration_seconds = mean(ride_length)) %>%
  arrange(desc(member_casual))
```

```
## # A tibble: 2 x 3
##   member_casual total_rides average_duration_seconds
##   <chr>               <int>                    <dbl>
## 1 member            2523958                     909.
## 2 casual            1926281                    2454.
```

It is evident from the result above that the members have more total number of rides whereas the casual riders have ride the bike longer on average than members. The reason may be due to the probability that the members are using the service more for daily commute and most of the casual riders may be tourists or visitors to the city.

Analysis to see what type of bike was popular among the two different rider types

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

```
## # A tibble: 7 x 4
## # Groups:   member_casual [3]
##   member_casual rideable_type number_of_rides average_duration
##   <chr>         <chr>                  <int>            <dbl>
## 1 casual        classic_bike          453760            1887.
## 2 casual        docked_bike           968172            3322.
## 3 casual        electric_bike         504349            1298.
## 4 member        classic_bike          824822             891.
## 5 member        docked_bike          1072362             979.
## 6 member        electric_bike         626774             813.
## 7 <NA>          <NA>                      40              NA
```

Analysis to see which are the top 5 stations for members and casual riders

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

```
## Selecting by most_popular_stations
```

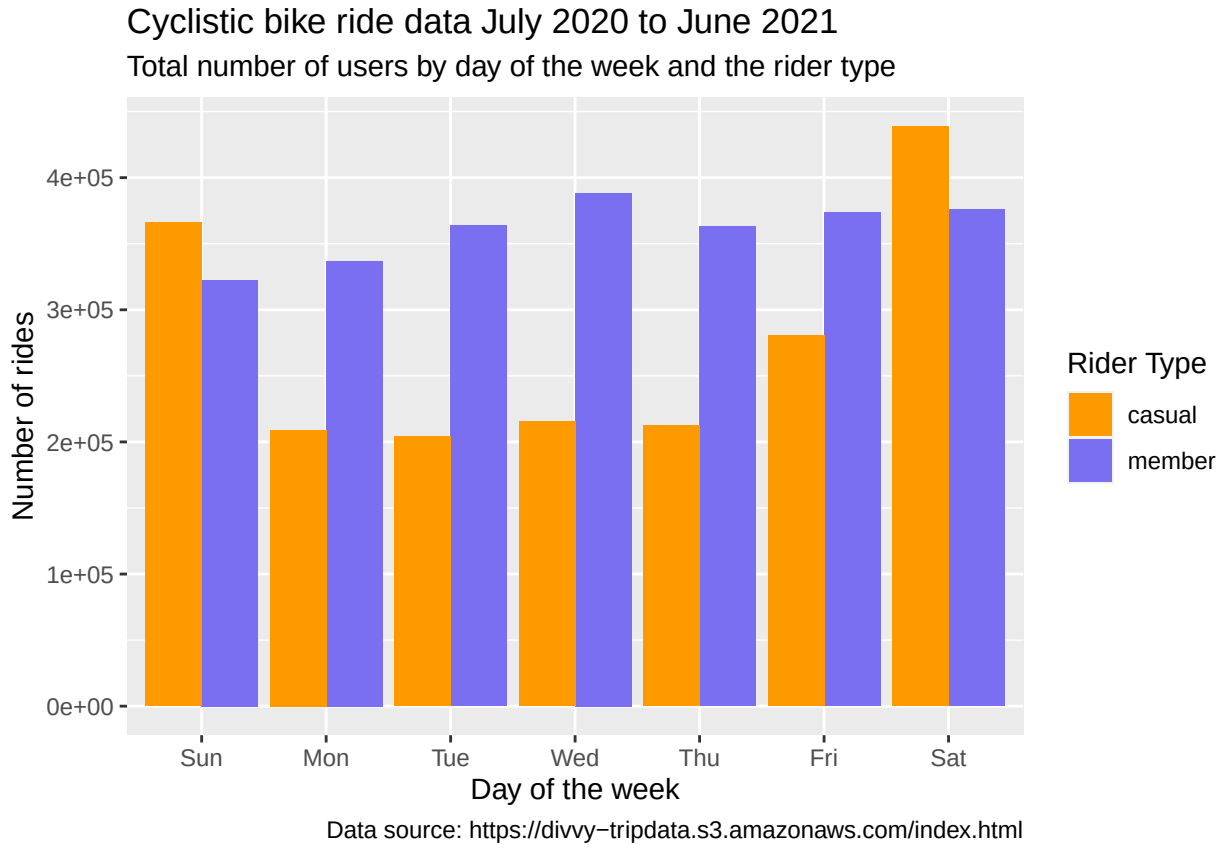
```
## # A tibble: 10 x 3
## # Groups:   member_casual [2]
##    member_casual start_station_name        most_popular_stations
##    <chr>         <chr>                                     <int>
##  1 casual        Streeter Dr & Grand Ave                   45004
##  2 casual        Lake Shore Dr & Monroe St                 31946
##  3 casual        Millennium Park                           28078
##  4 casual        Michigan Ave & Oak St                     22174
##  5 casual        Theater on the Lake                       21018
##  6 member        Clark St & Elm St                         24130
##  7 member        Wells St & Concord Ln                     19331
##  8 member        Kingsbury St & Kinzie St                  19026
##  9 member        Theater on the Lake                       18510
## 10 member        Dearborn St & Erie St                     18201
```

Visualization

Visualizing the different scenarios such as the total number of rider by rider type by day of the week, monthly &bike type and the average ride duration by day of the week, monthly, rider type, and by bike types. The colours in the visualizations are bright and of high contrast as they are accessible colours and makes it easier for anyone to see the difference in the colours denoting the member and the casual type of riders.

Visualizing the total number of rides by rider type and day of the week:

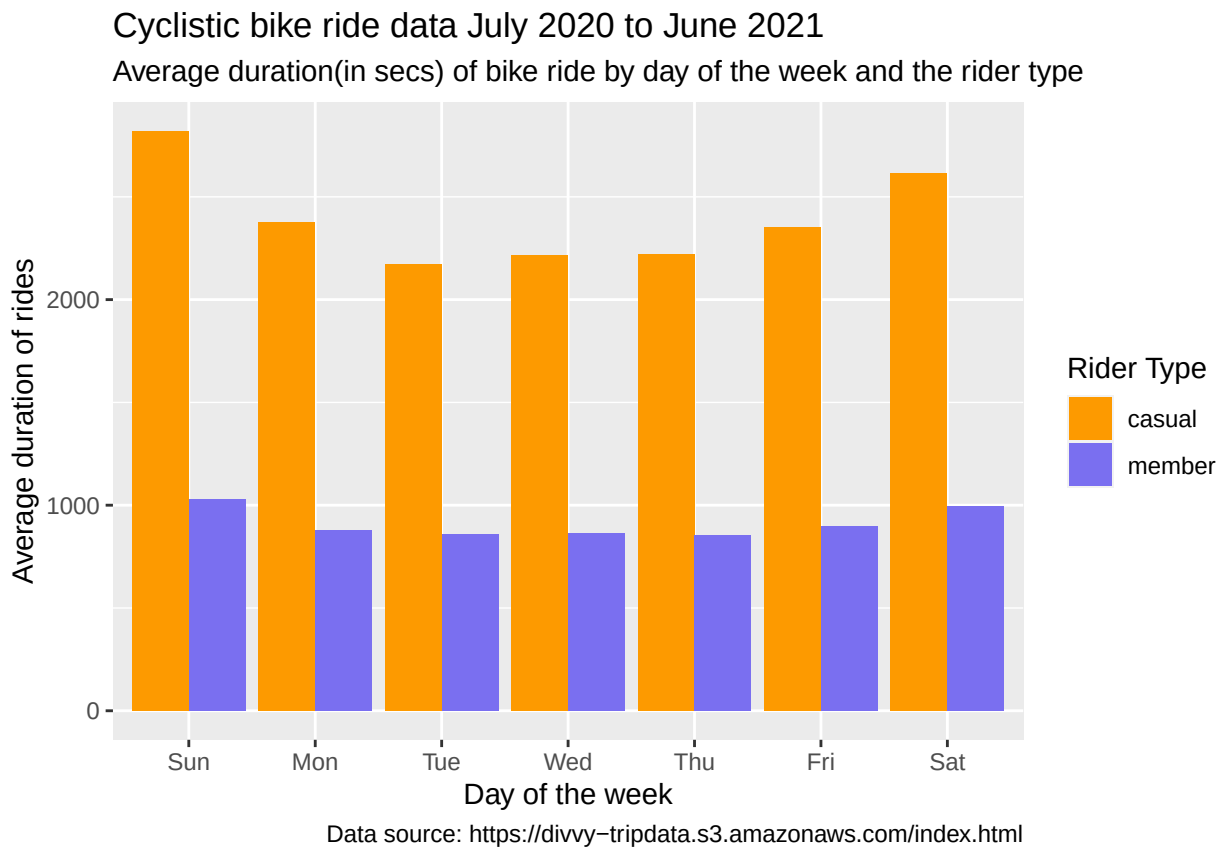
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```



It is clear from the data viz above that the members complete more number of rides on weekdays and the casual riders complete more rides on weekends.

Visualizing the average duration of rides by rider type and day of the week:

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



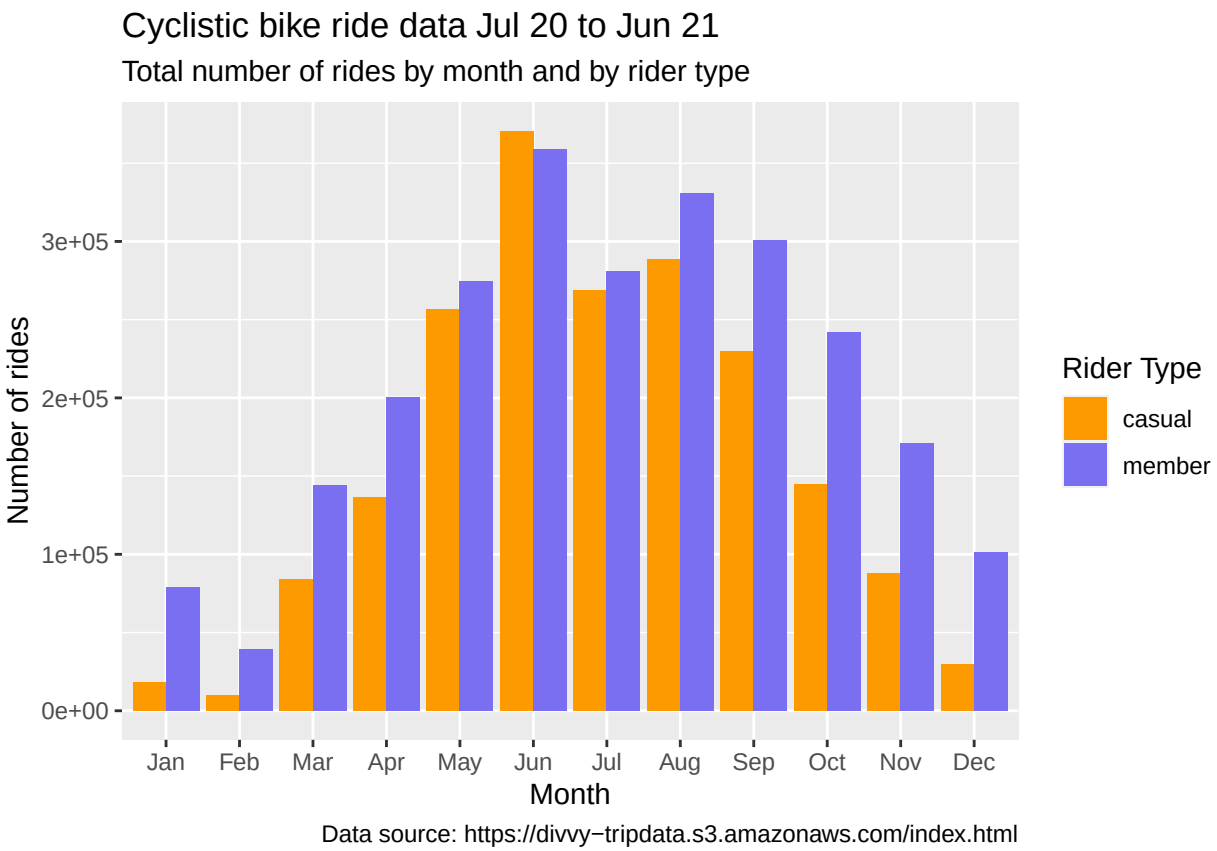
It is obvious from the data viz above that the casual riders ride for longer duration on average than the members on any given day of the week.

Let us fix the order of the months beginning jul 20 till jun 21 before beginning to analyze the number of bike rides by each rider type by month.

```
clean_data$month <- ordered(clean_data$month, levels=c("jul","aug","sep","oct","nov","dec","jan","feb",
```

Visualization of the total number of rides by month and by user type:

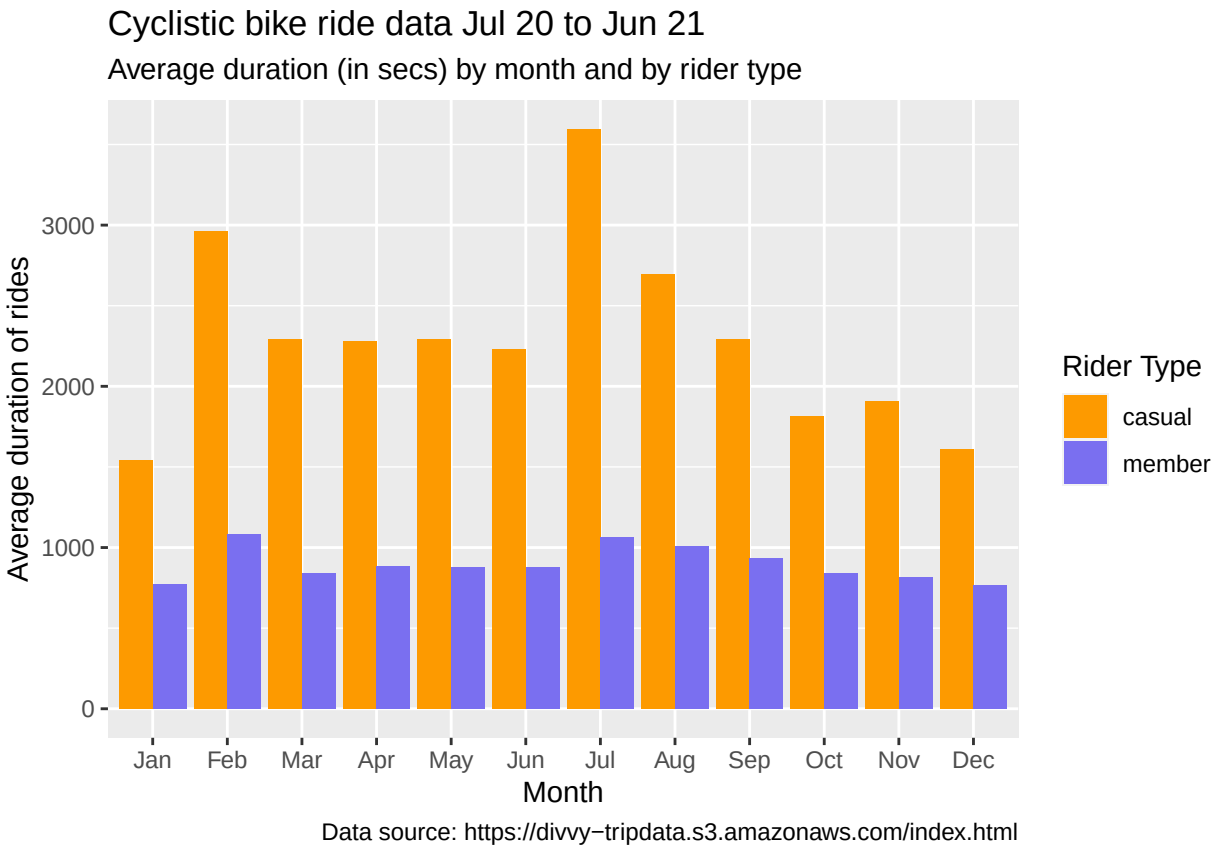
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```



The total number of rides is very low for casual members in winter months than the members. During the summer months the total number of rides are almost the same for casual riders and members.

Visualization of the average duration of rides by month and by user type:

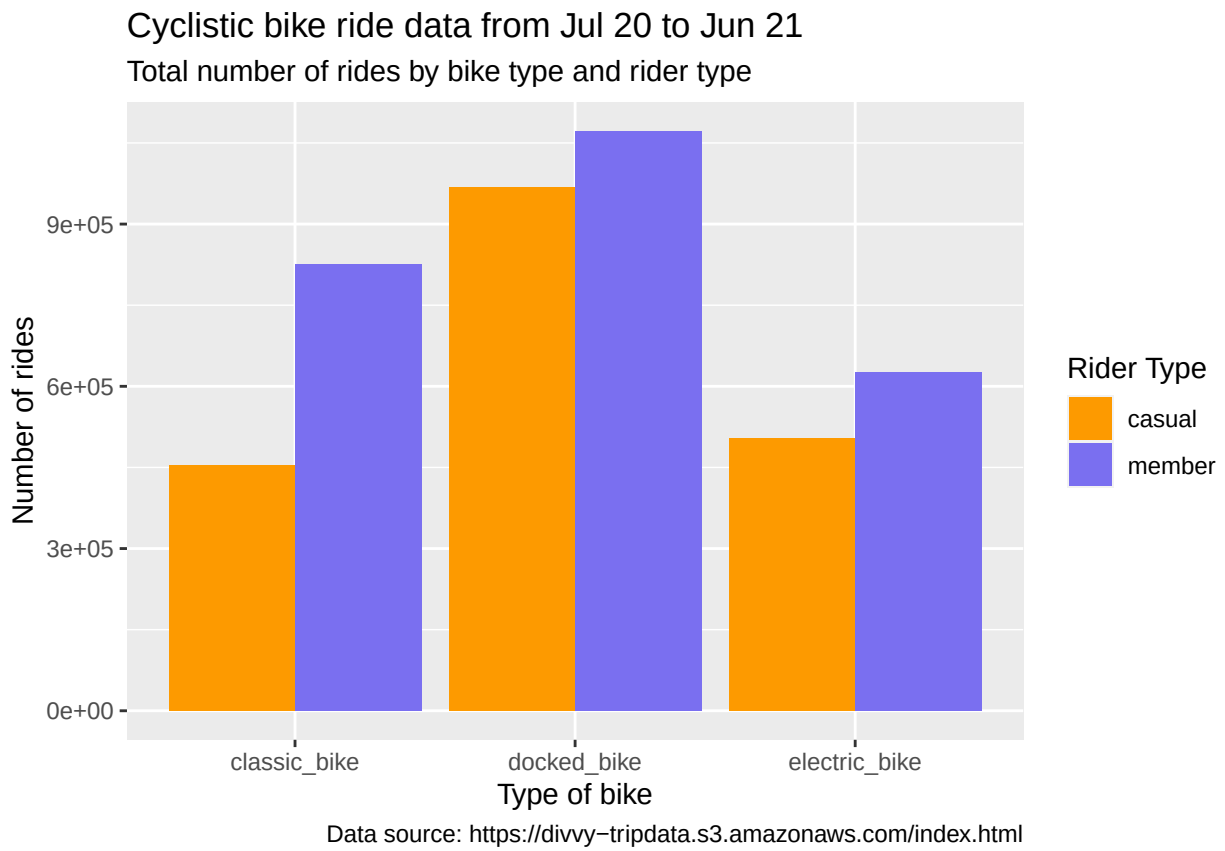
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```



The average duration of rides remains more or less same all through the year for members whereas it is higher during the summer vacation months of July and August for casual members. Overall, casual riders tend to ride for longer duration than the members.

Visualization of total rides by bike type and by rider type:

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

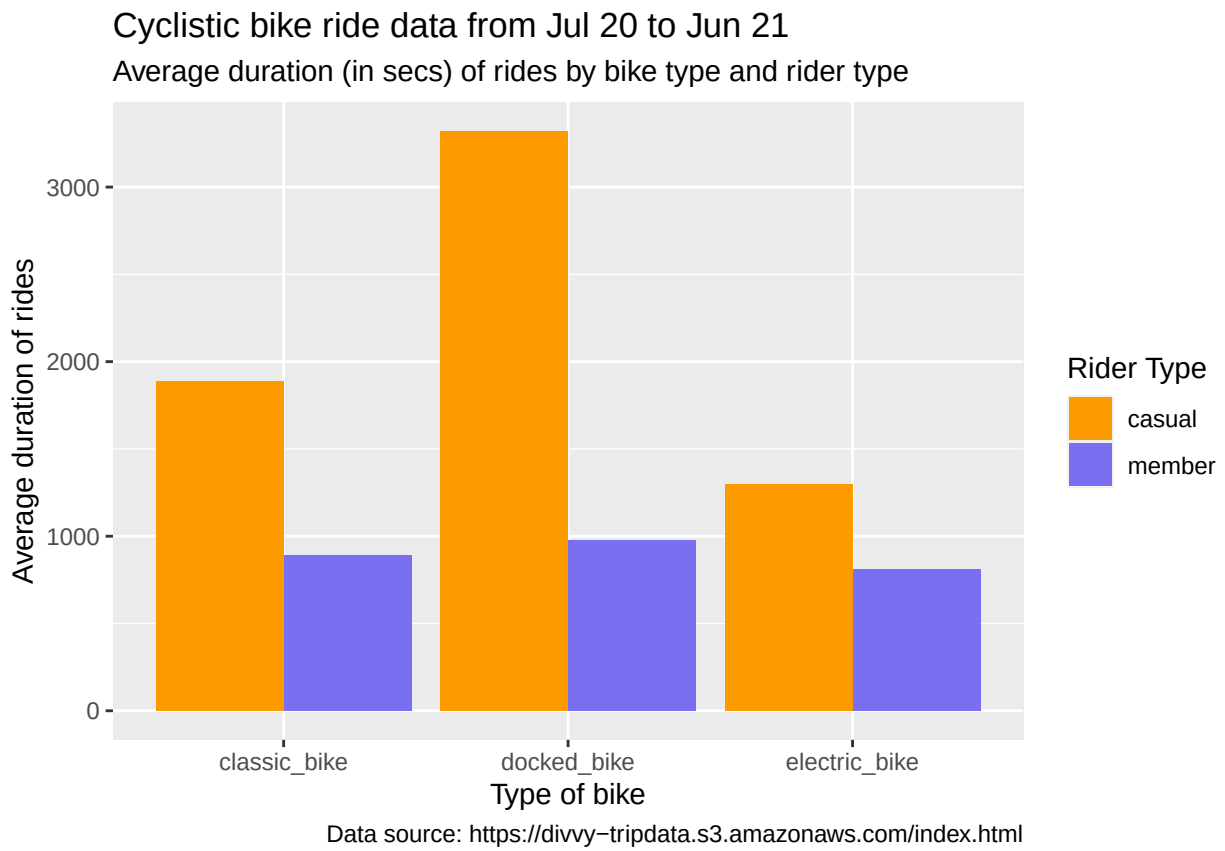


Both members and casual riders seem to ride the docked bike type more than the electric or classic bikes. It is hard to say the exact position of classic bikes as they were introduced only in 2021 and we only have about 6 months worth of data for them.

Visualization of average duration of rides by bike type and rider type: Showing the code as well as the results below:

```
clean_data %>%
  filter(!is.na(member_casual)) %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides =n(), average_duration = mean(ride_length)) %>%
  arrange(member_casual, rideable_type) %>%
  ggplot(mapping = aes(x=rideable_type, y= average_duration, fill = member_casual)) + geom_col(position
  labs(title = "Cyclistic bike ride data from Jul 20 to Jun 21", subtitle = "Average duration (in secs)
  scale_fill_manual(values = c("#FD9A00", "#7A6FF0"))
```

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



Even though the members put same average duration on all the three kinds of bikes, casual riders put in longer average duration on docked bikes than the other types.

Share

Key findings:

- Riders with annual membership complete more number of rides than the casual riders.
- Casual members ride for longer average duration than the members with annual membership
- Average duration of the ride for members stayed almost the same on weekdays and slightly higher on weekends.
- Average ride duration of casual riders were at least twice as much as that of members on any given day.
- Total number of rides is generally higher on warmer months compared to winter months for both members and casual riders.
- Docked bike has been the most popular among both members and casual riders.

Recommendations:

- For casual riders that are out of town (we need more data to determine what percentage of casual riders are in-town and out-of-town) Cyclistic can introduce a weekly or bi-weekly subscription. The week can start on a Monday at 12 a.m. and end on a Sunday at 11:59:59 p.m. This will benefit both Cyclistic and the casual rider. Make this membership a little expensive compared to annual membership but cheaper than one-ride or one-day pass to make attractive annual membership for in-town riders.

- Create a weekend subscription program with casual riders that are visitors or tourists. This will give access to the bikes for Friday, Saturday and Sunday.

Further Actions

- Collecting data on whether a casual rider is from Chicago or from out of town will be very helpful to determine the percentage of our target market among the casual riders for annual subscriptions.