# Cyclistic

Madhav Anand

2023-04-11

## Cyclistic Case Study.

### Scenario

*You are a junior data analyst working in the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.*

*In this case study we will analyze and answer the business question. The following steps will be taken to analyze data.*

- Ask - In this stage, we will define the business question or the main objective.
- Prepare - Then, we will prepare the data. Key questions such as where the data is located or is the data organized will be answered.
- Process - In this step we will clean the data.
- Analyze - Then analyse the data.
- Share - Here, storytelling through data visualization will be done.
- Act - Then, recommendations based on the analysis will be made.

### Ask

#### Questions about business matters that call for answers.

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

## Prepare

*The data on which I'd been working are public datasets, click here to access the data sets. Motivate International Inc. has provided access to the data under this license.*

*Dataset also follows its reliability, originality, and comprehensiveness and does not miss any important information. The data is updated monthly and its* licensed

*Key steps taken in this stage are as follows;*

*I downloaded .csv files into the directory. Period captured: March 2023 to April 2022.*

*Imported the data in **RStudios**.*

- Loading the libraries and setting up my R environment.

```
library(tidyverse)

library(ggplot2)
library(dplyr)
library(lubridate)
```

- Loading .csv files into the data frames. Each data frame has 13 columns or variables with the similar column names.

```
mar_23 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202303-divvy-tripdata.csv")

feb_23 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202302-divvy-tripdata.csv")

jan_23 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202301-divvy-tripdata.csv")

dec_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202212-divvy-tripdata.csv")

nov_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202211-divvy-tripdata.csv")

oct_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202210-divvy-tripdata.csv")

sep_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202209-divvy-
publictripdata.csv")

aug_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202208-divvy-tripdata.csv")

jul_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202207-divvy-tripdata.csv")
```

```
jun_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202206-divvy-tripdata.csv")

may_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202205-divvy-tripdata.csv")

apr_22 <- read_csv("C:\\Users\\User\\OneDrive\\Desktop\\Data
Analytics\\Capstone project\\case 1\\source\\202204-divvy-tripdata.csv")
```

- Stacking the files into a new data frame *cyclistic_df*.

```
cyclistic_df <- rbind(mar_23, feb_23, jan_23, dec_22, nov_22, oct_22, sep_22,
                      aug_22, jul_22, jun_22, may_22, apr_22)


str(cyclistic_df) #to explore the data structure.

colnames(cyclistic_df)

##  [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"
```

## Process

### I cleaned the data in RStudios. Following are the data cleaning steps:-

- Removing start_lat, start_lng, end_lat,end_lng

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

colnames(cyclistic_df)#verification 4 columns are removed

## [1] "ride_id"            "rideable_type"      "started_at"
## [4] "ended_at"           "start_station_name" "start_station_id"
## [7] "end_station_name"   "end_station_id"     "member_casual"
```

- Data exploration.

```
summary(cyclistic_df)# to get summary statistics

##    ride_id          rideable_type        started_at
##  Length:5803720     Length:5803720     Min.   :2022-04-01 00:01:48.00
##  Class :character   Class :character   1st Qu.:2022-06-18 23:27:00.25
##  Mode  :character   Mode  :character   Median :2022-08-13 11:37:32.00
##                                        Mean   :2022-08-25 07:04:55.95
##                                        3rd Qu.:2022-10-14 18:04:21.00
##                                        Max.   :2023-03-31 23:59:28.00
##     ended_at                      start_station_name start_station_id
##  Min.   :2022-04-01 00:02:15.00   Length:5803720     Length:5803720
##  1st Qu.:2022-06-18 23:51:55.75   Class :character   Class :character
```

```
##  Median :2022-08-13 12:00:07.50   Mode  :character   Mode  :character
##  Mean   :2022-08-25 07:23:54.70
##  3rd Qu.:2022-10-14 18:19:10.25
##  Max.   :2023-04-03 11:41:11.00
##  end_station_name   end_station_id     member_casual
##  Length:5803720     Length:5803720     Length:5803720
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
```

```
unique(cyclistic_df$member_casual)
```

```
## [1] "member" "casual"
```

```
unique(cyclistic_df$rideable_type)
```

```
## [1] "electric_bike" "classic_bike"  "docked_bike"
```

- To know how many observations fall under each user type

```
table(cyclistic_df$member_casual)
```

```
##
##  casual  member
## 2337439 3466281
```

- There two member types: casual and member, however the case study says to replace “casual” type to “customer” and “member” to “subscriber”.

```
cyclistic_df$member_casual <- ifelse(cyclistic_df$member_casual ==
                                       'casual', 'Customer', 'Subscriber')
```

- verification, there are similar observations

```
table(cyclistic_df$member_casual)
```

```
##
##   Customer Subscriber
##    2337439    3466281
```

- Adding columns that lists the date, month, day, year, and weekday.

```
cyclistic_df$Date <- as.Date(cyclistic_df$started_at)
cyclistic_df$Day <- format(cyclistic_df$Date, "%d") #day column
cyclistic_df$Month <- format(cyclistic_df$Date, "%m")#month column
cyclistic_df$Year <- format(cyclistic_df$Date, "%y")#Year column
cyclistic_df$weekdays <- weekdays(cyclistic_df$Date)#weekdays column
```

- Adding a new metric ride_length into data frame.

```
cyclistic_df <- cyclistic_df %>%
  mutate(ride_length = ended_at - started_at)
```

- Deleting observation in ride_length variable where ride_length is negative, and start_station_name is “HQ RQ.” Because we don’t want any negative observation, it

will skew the result. And removing "HQ RQ" observations because the case study says it is the station where bikes go for service.

```
cyclistic_df_2 <- cyclistic_df[!(cyclistic_df$start_station_name == "HQ RQ"|
                                 cyclistic_df$ride_length < 0),]
```

- Converting ride_length to numeric variable

```
cyclistic_df_2$ride_length <- as.numeric(cyclistic_df_2$ride_length)
```

- Let's verify it..

```
str(cyclistic_df_2)
```

```
## tibble [5,803,621 × 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5803621] "6842AA605EE9FBB3"
"F984267A75B99A8C" "FF7CF57CFE026D02" "6B61B916032CB6D6" ...
##  $ rideable_type     : chr [1:5803621] "electric_bike" "electric_bike"
"classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:5803621], format: "2023-03-16 08:20:34"
"2023-03-04 14:07:06" ...
##  $ ended_at          : POSIXct[1:5803621], format: "2023-03-16 08:22:52"
"2023-03-04 14:15:31" ...
##  $ start_station_name: chr [1:5803621] "Clark St & Armitage Ave" "Public
Rack - Kedzie Ave & Argyle St" "Orleans St & Chestnut St (NEXT Apts)"
"Desplaines St & Kinzie St" ...
##  $ start_station_id  : chr [1:5803621] "13146" "491" "620" "TA1306000003"
...
##  $ end_station_name  : chr [1:5803621] "Larrabee St & Webster Ave" NA
"Clark St & Randolph St" "Sheffield Ave & Kingsbury St" ...
##  $ end_station_id    : chr [1:5803621] "13193" NA "TA1305000030" "13154"
...
##  $ member_casual     : chr [1:5803621] "Subscriber" "Subscriber"
"Subscriber" "Subscriber" ...
##  $ Date              : Date[1:5803621], format: "2023-03-16" "2023-03-04"
...
##  $ Day               : chr [1:5803621] "16" "04" "31" "22" ...
##  $ Month             : chr [1:5803621] "03" "03" "03" "03" ...
##  $ Year              : chr [1:5803621] "23" "23" "23" "23" ...
##  $ weekdays          : chr [1:5803621] "Thursday" "Saturday" "Friday"
"Wednesday" ...
##  $ ride_length       : num [1:5803621] 138 505 638 943 660 867 415 1320
587 717 ...
```

- Separating Seconds from the the numeric values.

```
cyclistic_df_2 <- separate(cyclistic_df_2, ride_length,
                           into = c('ride_length_in_sec', 'sec'), sep = " ")
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 4964563
rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
cyclistic_df_2$ride_length_in_sec <-
  as.numeric(cyclistic_df_2$ride_length_in_sec)
```

- Removing the temporary column

```
cyclistic_df_2 <- cyclistic_df_2 %>% select(-c(sec))
```

- Converting weekdays into factor.

```
cyclistic_df_2$weekdays <- as.factor(cyclistic_df_2$weekdays)
```

## Analysis

*Key tasks:*

- Aggregate the data so it's useful and accessible.
- Organize and format the data.
- Perform calculations.
- Identify trends and relationships.

*Steps for analysis process are as follows.*

- Conducting descriptive analysis

```
aggregate(cyclistic_df_2$ride_length_in_sec ~ cyclistic_df_2$member_casual,
FUN = mean) #Average values

##   cyclistic_df_2$member_casual cyclistic_df_2$ride_length_in_sec
## 1                     Customer                          1865.3484
## 2                   Subscriber                           755.3508

aggregate(cyclistic_df_2$ride_length_in_sec ~ cyclistic_df_2$member_casual,
FUN = median) #Median values

##   cyclistic_df_2$member_casual cyclistic_df_2$ride_length_in_sec
## 1                     Customer                                793
## 2                   Subscriber                                525

aggregate(cyclistic_df_2$ride_length_in_sec ~ cyclistic_df_2$member_casual,
FUN = max) #Maximum values

##   cyclistic_df_2$member_casual cyclistic_df_2$ride_length_in_sec
## 1                     Customer                            2483235
## 2                   Subscriber                              93580

aggregate(cyclistic_df_2$ride_length_in_sec ~ cyclistic_df_2$member_casual,
FUN = min) #Minimum values

##   cyclistic_df_2$member_casual cyclistic_df_2$ride_length_in_sec
## 1                     Customer                                  0
## 2                   Subscriber                                  0
```

- Calculating average ride time by each day for customers vs subscribers in a whole year. First ordering the values in weekdays column.

```
  ordered(cyclistic_df_2$weekdays, levels = c("Sunday","Monday", "Tuesday",
                                              "Wednesday", "Thursday",
"Friday",
                                              "Saturday"))
```

- Here, we have average ride time in seconds, with distinct weekdays and member type in a whole year.

```
aggregate (cyclistic_df_2$ride_length_in_sec ~ cyclistic_df_2$member_casual +
             cyclistic_df_2$weekdays, FUN = mean)
```

```
##    cyclistic_df_2$member_casual cyclistic_df_2$weekdays
## 1                      Customer                  Friday
## 2                    Subscriber                  Friday
## 3                      Customer                  Monday
## 4                    Subscriber                  Monday
## 5                      Customer                Saturday
## 6                    Subscriber                Saturday
## 7                      Customer                  Sunday
## 8                    Subscriber                  Sunday
## 9                      Customer                Thursday
## 10                   Subscriber                Thursday
## 11                     Customer                 Tuesday
## 12                   Subscriber                 Tuesday
## 13                     Customer               Wednesday
## 14                   Subscriber               Wednesday
##    cyclistic_df_2$ride_length_in_sec
## 1                          1814.7435
## 2                           744.4002
## 3                          1857.2519
## 4                           726.4449
## 5                          2100.1787
## 6                           850.0833
## 7                          2182.5023
## 8                           840.6973
## 9                          1615.5363
## 10                          730.4094
## 11                         1652.2085
## 12                          717.8231
## 13                         1564.4952
## 14                          715.6650
```

- Analyzing data set by member type and weekdays and saving it in a new dataframe. Here I calculated how many numbers of observation and average ride time there are weekly.

```
final_analysis <- cyclistic_df_2 %>%
  group_by(member_casual, weekdays) %>%
  summarize(number_of_rides = n(), average_duration =
mean(ride_length_in_sec)) %>%
  arrange(member_casual, weekdays) %>%
  drop_na()
```

## Share

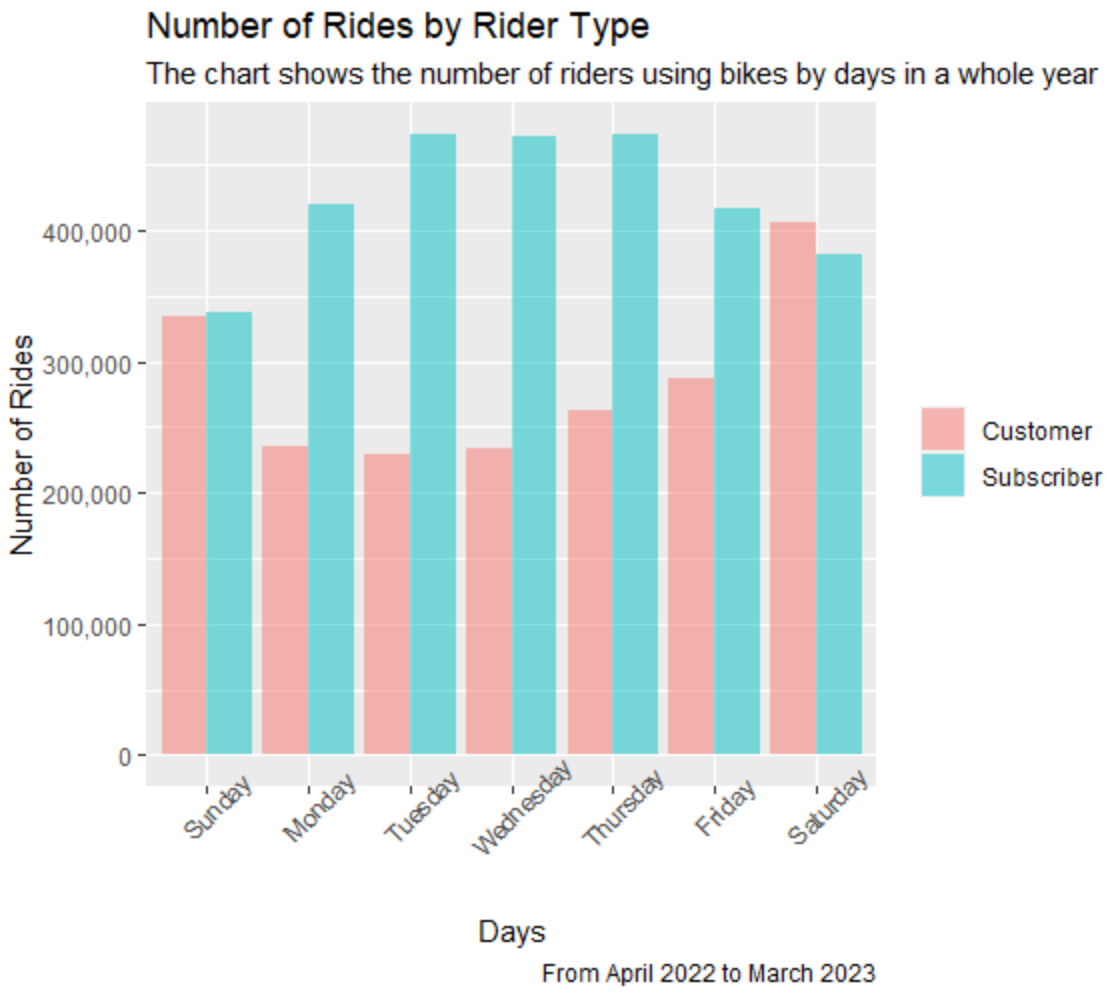
*Here, I made some viz using R and Tableau. Key tasks are as followed.*

- Create effective data visualizations.
- Present the findings.

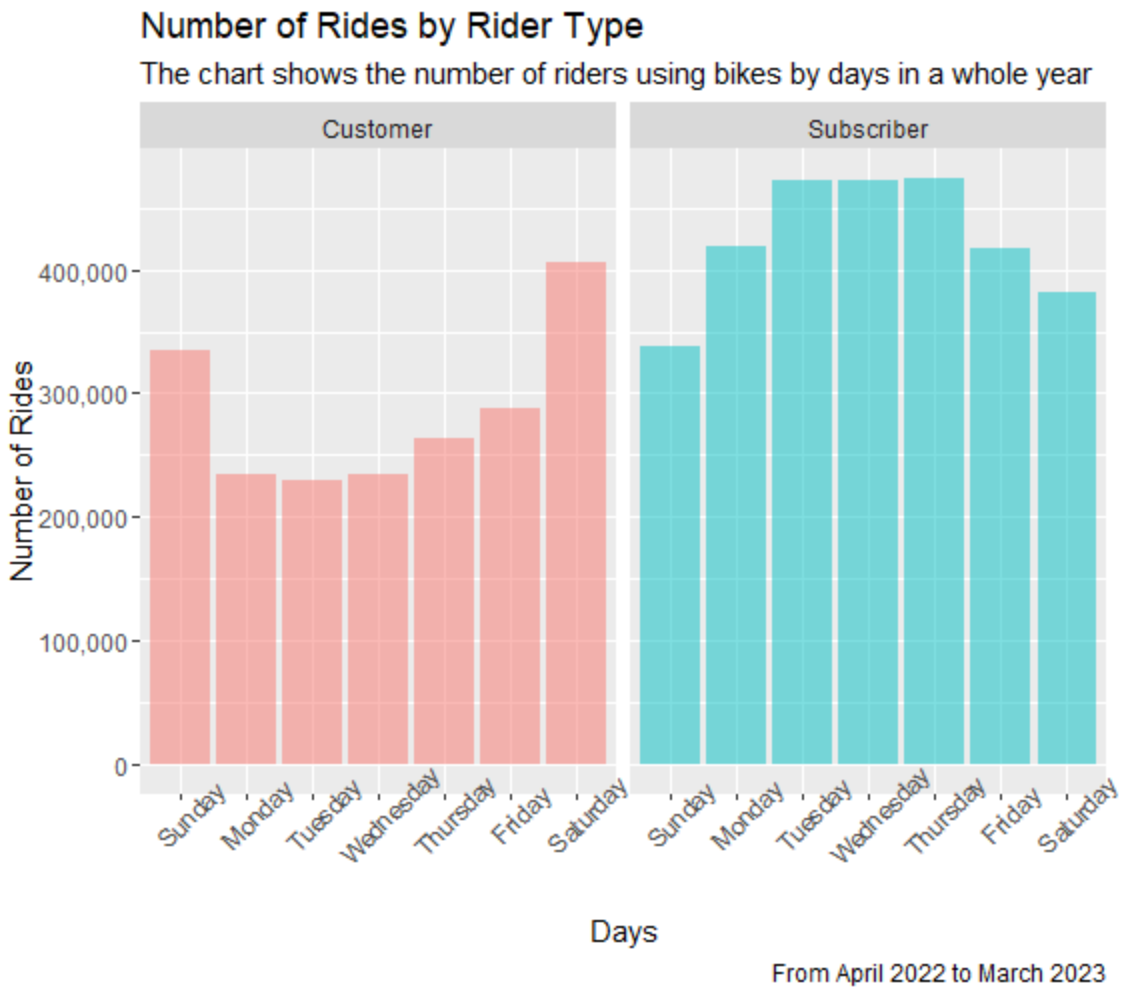
*Following are the code chunks for building data viz.*

- Visualizing number of rides by rider type

```
ggplot(final_analysis, aes(x = weekdays, y = number_of_rides, fill =
member_casual))+
  geom_col(position = "dodge", alpha = 0.6)+
  theme(axis.text.x = element_text(angle = 45), legend.title =
element_blank())+
  labs(title = "Number of Rides by Rider Type",
       subtitle = "The chart shows the number of riders using bikes by days
in a whole year",
      caption = "From April 2022 to March 2023" , x = "Days", y = "Number of
Rides")+
  scale_y_continuous(label = scales::comma)
```




```
ggplot(final_analysis, aes(x = weekdays, y = number_of_rides, fill =
member_casual))+
  geom_col(position = "dodge", alpha = 0.6)+
  facet_wrap(~member_casual)+
  theme(axis.text.x = element_text(angle = 90), legend.title =
element_blank(),
        legend.position = "none")+
  labs(title = "Number of Rides by Rider Type",
       subtitle = "The chart shows the number of riders using bikes by days
in a whole year",
       caption = "From April 2022 to March 2023" , x = "Days", y = "Number of
Rides")+
  scale_y_continuous(label = scales::comma)# to convert from scientific
notation to readable format
```




`The chart above is the summary of distinct weekdays which shows the number of members using the bike in a whole year. In this dataset I found that:

a. On weekends, both customers and subscribers have almost similar ratio of using bikes.

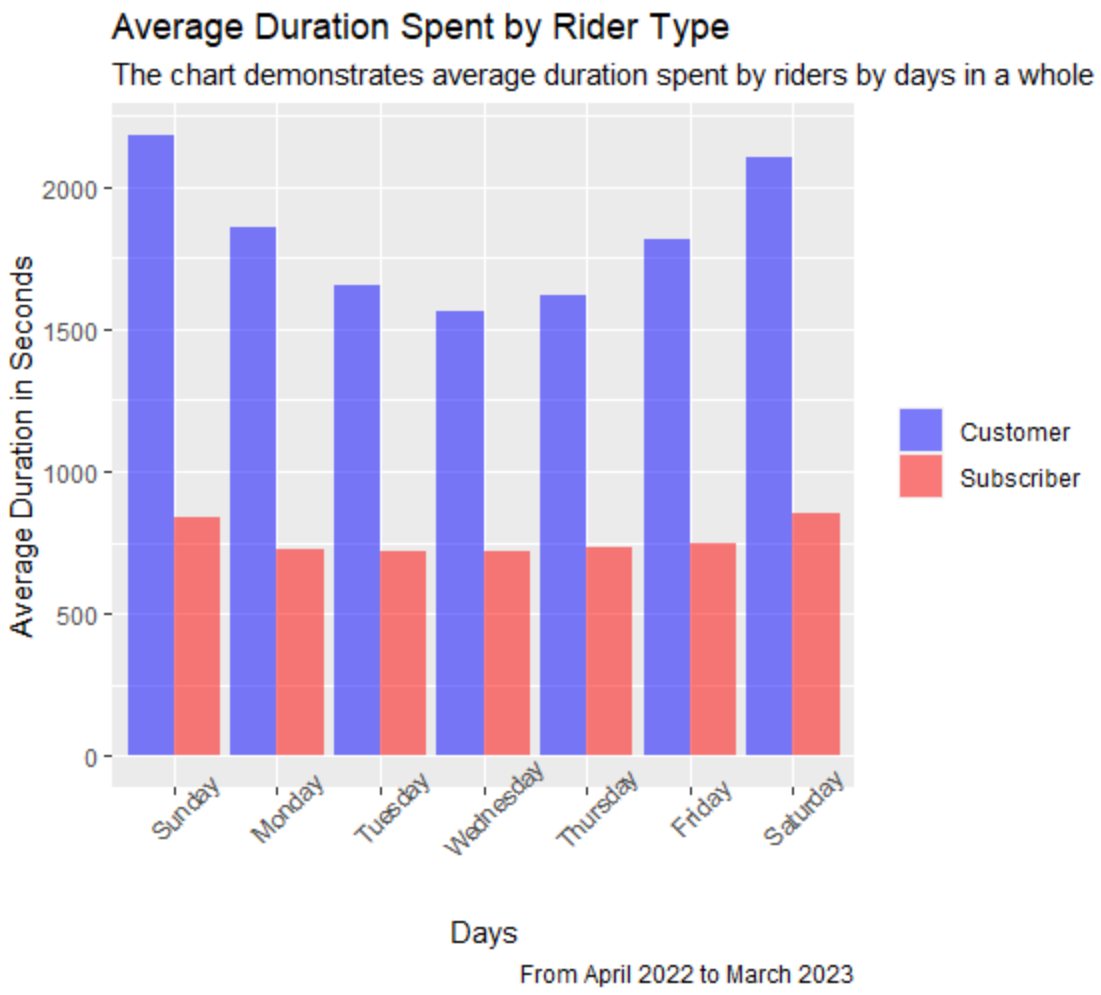
b. However, the ratio of subscribers is more than the customers during weekdays.

c. That makes a clear objective to sell subscriptions to customers.

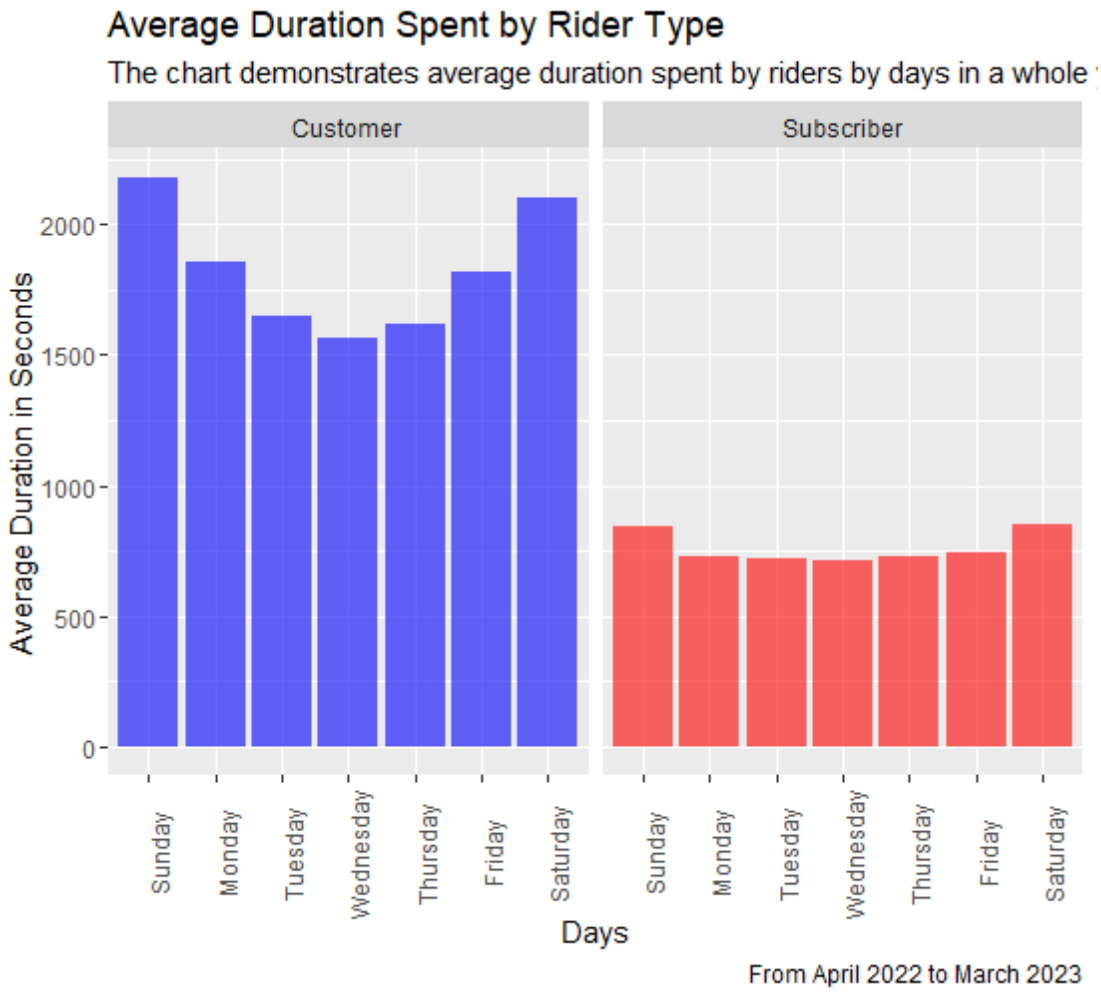
- Visualization for average duration

```
ggplot(final_analysis, aes(x = weekdays, y = average_duration, fill =
member_casual))+
  geom_col(position = "dodge", alpha = 0.5)+
  labs(title = "Average Duration Spent by Rider Type ",
       subtitle = "The chart demonstrates average duration spent by riders by
days in a whole year",
       caption = "From April 2022 to March 2023", x = "Days", y = "Average
```

```
Duration in Seconds")+
  theme(axis.text.x = element_text(angle = 45), legend.title  =
element_blank())+
  scale_fill_manual(values = c("blue", "red"))
```




```
ggplot(final_analysis, aes(x = weekdays, y = average_duration, fill =
member_casual))+
  geom_col(position = "dodge", alpha = 0.6)+
  facet_wrap(~member_casual)+
  labs(title = "Average Duration Spent by Rider Type ",
       subtitle = "The chart demonstrates average duration spent by riders by
days in a whole year",
       caption = "From April 2022 to March 2023", x = "Days", y = "Average
Duration in Seconds")+
  theme(axis.text.x = element_text(angle = 90), legend.title  =
element_blank(), legend.position = "none")+
  scale_fill_manual(values = c("blue", "red"))
```

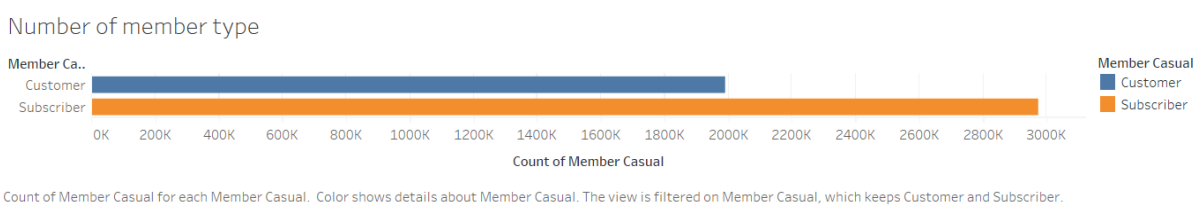



The chart above is the summary of distinct weekdays which shows the average duration spent by members using the bike in a whole year. Here I found that, on average, bike usage time is more in the case of customers than the subscribers.

- Exporting files to the system location for further analysis or exploration.

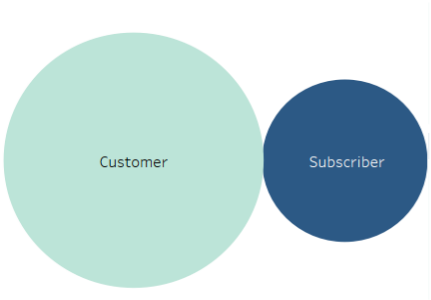
```
write_csv(final_analysis,"C:\\Users\\User\\OneDrive\\Desktop\\Data Analytics\\Capstone project\\case 1\\final_data set\\Final_Analysis.csv")
write_csv(cyclistic_df_2, "C:\\Users\\User\\OneDrive\\Desktop\\Data Analytics\\Capstone project\\case 1\\final_data set.Cyclistic_Final.csv")
```

- Data viz via Tableau



The company has more subscribers than the customers. 60% of member types are subscribers. Therefore, it's a great opportunity to sell subscriptions to customers.

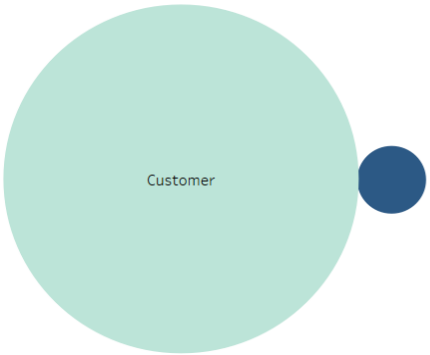
Average Ride Time




Average ride time in seconds

| Member Ca.. | |
|---|---|
| Customer | 1,865 |
| Subscriber | 755 |

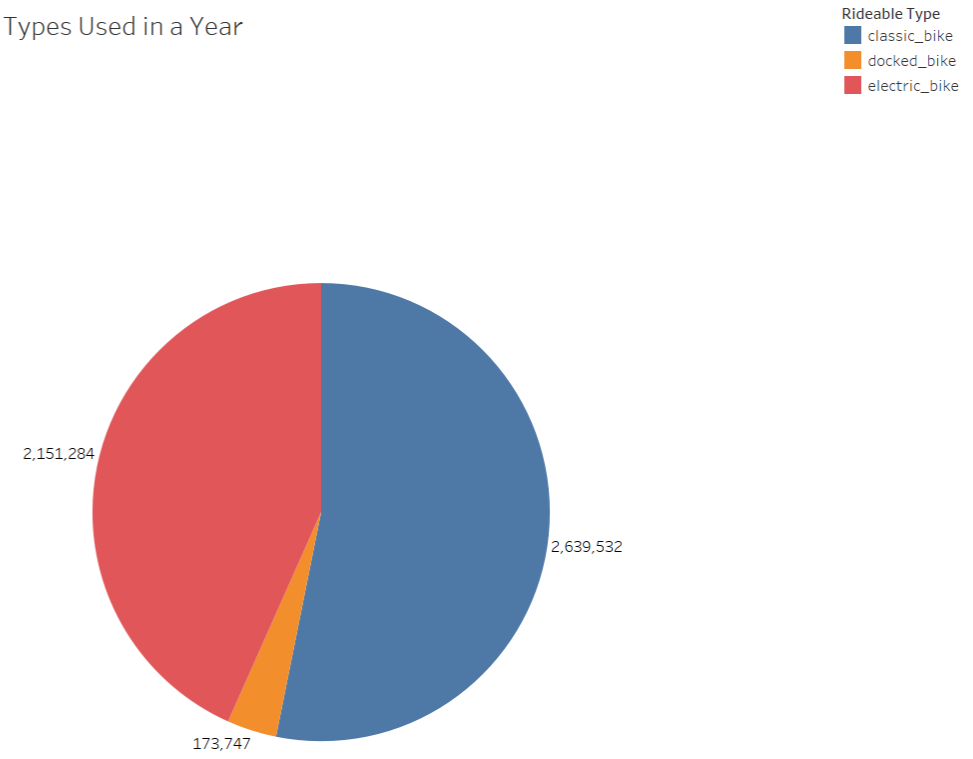
Maximum Ride Time




Maximum ride time in seconds

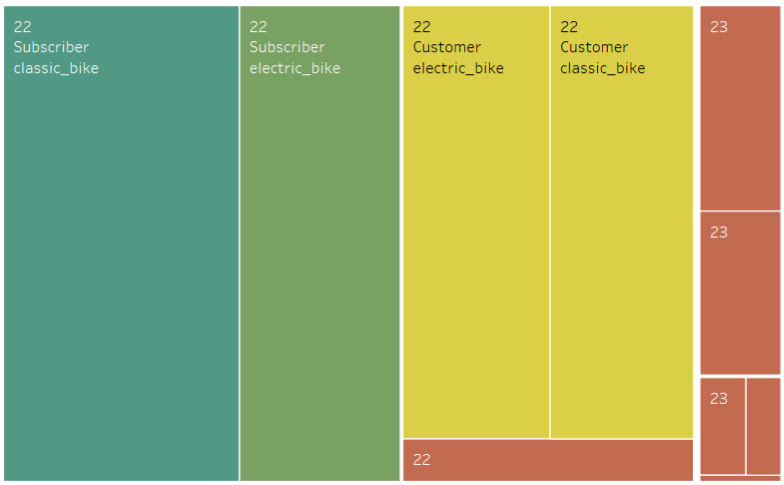
| Member Ca.. | |
|---|---|
| Customer | 2,483,235 |
| Subscriber | 93,580 |

Number of Bike Types Used in a Year



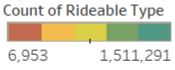

From the graph above, we can notice that classic bikes are more popular among people, however docked bikes were used less than the other two.

Number of Ridable Type Per Year by each Member Type

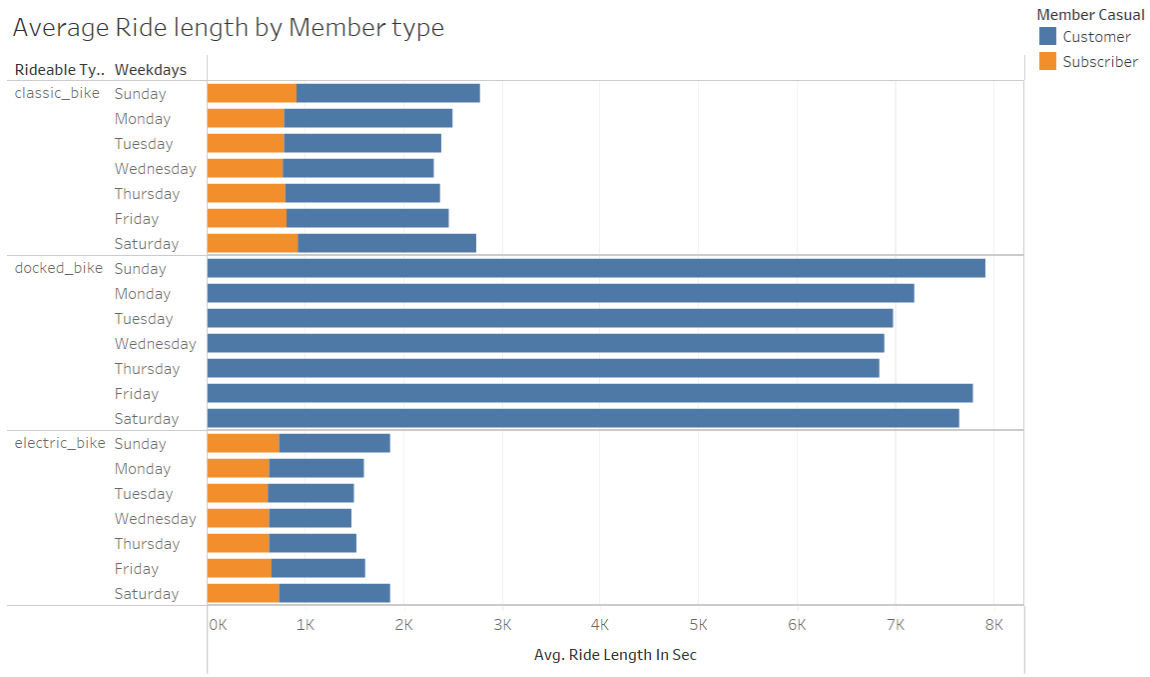



Number of Ridable Types

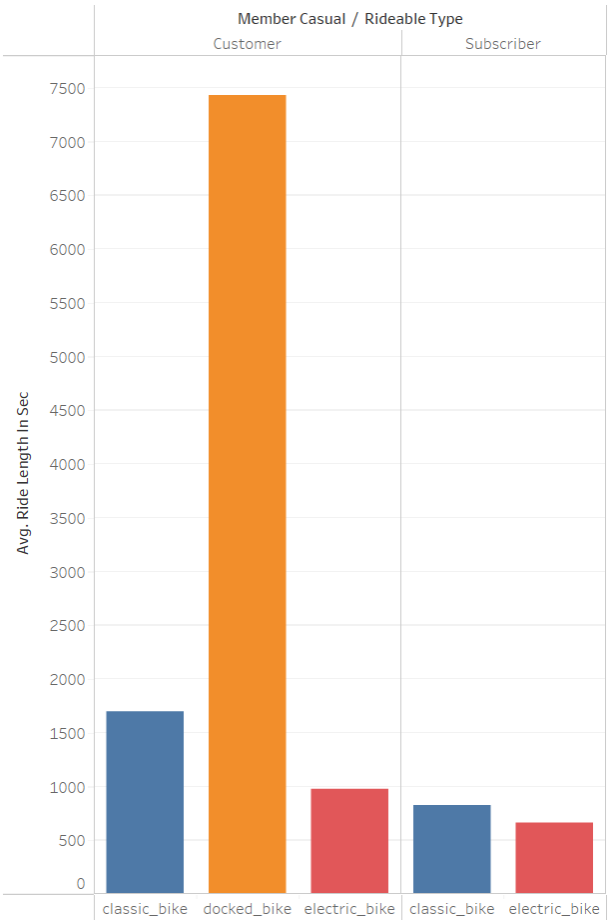
| Year | Rideable Ty.. | Member Casual: Customer | Member Casual: Subscriber |
|---|---|---|---|
| 22 | classic_bike | 840,976 | 1,511,291 |
| | docked_bike | 166,794 | |
| | electric_bike | 859,457 | 1,034,725 |
| 23 | classic_bike | 48,899 | 238,366 |
| | docked_bike | 6,953 | |
| | electric_bike | 67,209 | 189,893 |



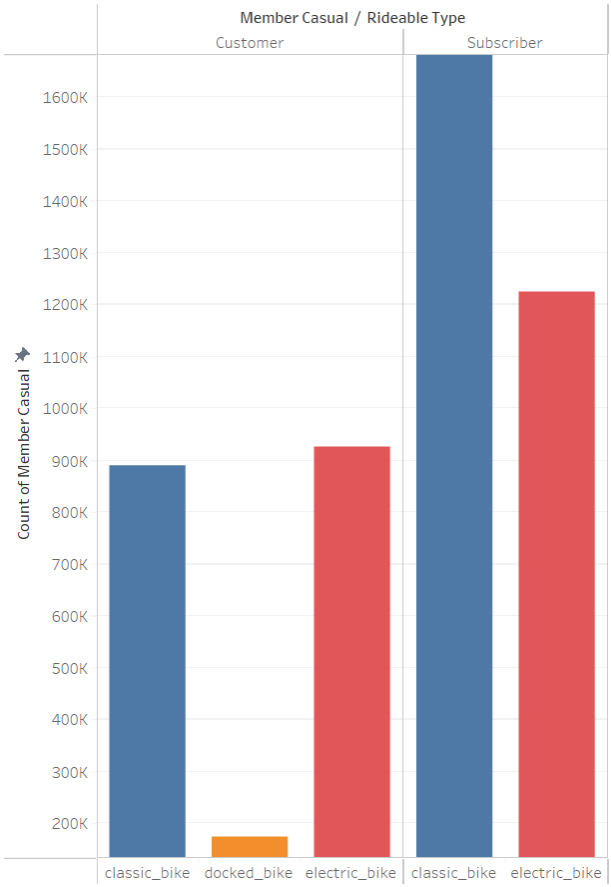
The Tree map above represents that classic and electric bikes were the most preferable options in the case of customers in 2022. We can also see this same trend in the year 2023 from January to March.



Average Ride Time by Member type




Number of Member Type using different Ridable Type




## Act

*The following are the case recommendations.*

To sell subscriptions to the customers.

- The company should increase the price of the rental bikes since we have seen from the column chart above the average duration spent is more in the case of customers or non-subscribers.
- The company can also give some good deals for classic and electric bikes in their subscription and educate nonsubscribers about it through marketing campaigns, since we know that the customers use more classic and electric bikes. Giving some extra rideable time in a subscription will be effective.
- Giving customers some seasonal discounts on subscriptions will also be helpful to make them buy.