Google Data Analytics Case study: How does a bike-share navigate speedy success?

Rhafael Chandra

2023-12-30

Table of Contents

1. Introduction

This is a case study for completing Google Data Analytics Professional Certificate. In this case study, you work for a fictional company, Cyclistic. Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the 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.

2. Scenario

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

3. Data Analysis

This case study will be using the 6 stages of the data analysis process:

1. Ask

2. Prepare
3. Process
4. Analyze
5. Share
6. Act

1) Ask

Three 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?

For the study case, the marketing director, who is also the manager, assigned me the first question to answer: **How do annual members and casual riders use Cyclistic bikes differently?**

2) Prepare

Source: For the purposes of this case study, the datasets that has been made available by Motivate International Inc. will be used for analysis. (Dataset)

Date Range: December 2022 to November 2023

There are 12 files following the name template of "*yyyymm-divvy-tripdata*". All of the files shares the same columns:

columns			
ride_id	start_station_name	start_lat	member_casual
rideable_type	start_station_id	start_lng	
started_at	end_station_name	end_lat	
ended_at	end_station_id	end_lng	

3) Process

The data are uploaded in R and combined using `bind_rows` from `dplyr` library in R.

```r
library(tidyverse)

df_202212 <- read_csv("divvy-tripdata/202212-divvy-tripdata.csv")
df_202301 <- read_csv("divvy-tripdata/202301-divvy-tripdata.csv")
df_202302 <- read_csv("divvy-tripdata/202302-divvy-tripdata.csv")
df_202303 <- read_csv("divvy-tripdata/202303-divvy-tripdata.csv")
df_202304 <- read_csv("divvy-tripdata/202304-divvy-tripdata.csv")
df_202305 <- read_csv("divvy-tripdata/202305-divvy-tripdata.csv")
df_202306 <- read_csv("divvy-tripdata/202306-divvy-tripdata.csv")
df_202307 <- read_csv("divvy-tripdata/202307-divvy-tripdata.csv")
df_202308 <- read_csv("divvy-tripdata/202308-divvy-tripdata.csv")
df_202309 <- read_csv("divvy-tripdata/202309-divvy-tripdata.csv")
df_202310 <- read_csv("divvy-tripdata/202310-divvy-tripdata.csv")
df_202311 <- read_csv("divvy-tripdata/202311-divvy-tripdata.csv")
```

```
df <- bind_rows(df_202212,
                df_202301,
                df_202302,
                df_202303,
                df_202304,
                df_202305,
                df_202306,
                df_202307,
                df_202308,
                df_202309,
                df_202310,
                df_202311)
```

Add `ride_length` and `day_of_week` columns We need to add some columns for further analysis,

- `ride_length` = `started_at` - `ended_at`
- `day_of_week` = extract day of week from `started_at` (1 = Sunday)

```
df <- mutate(df,
            ride_length = difftime(ended_at, started_at, units = "secs"),
            day_of_week = wday(df$started_at))
glimpse(df)
```

```
## Rows: 5,677,610
## Columns: 15
## $ ride_id            <chr> "65DBD2F447EC51C2", "0C201AA7EA0EA1AD", "E0B148CCB3~
## $ rideable_type      <chr> "electric_bike", "classic_bike", "electric_bike", "~
## $ started_at         <dttm> 2022-12-05 10:47:18, 2022-12-18 06:42:33, 2022-12-~
## $ ended_at           <dttm> 2022-12-05 10:56:34, 2022-12-18 07:08:44, 2022-12-~
## $ start_station_name <chr> "Clifton Ave & Armitage Ave", "Broadway & Belmont A~
## $ start_station_id   <chr> "TA1307000163", "13277", "TA1306000015", "KA1503000~
## $ end_station_name   <chr> "Sedgwick St & Webster Ave", "Sedgwick St & Webster~
## $ end_station_id     <chr> "13191", "13191", "13016", "13134", "13288", "KA150~
## $ start_lat          <dbl> 41.91824, 41.94011, 41.88592, 41.83846, 41.89595, 4~
## $ start_lng          <dbl> -87.65711, -87.64545, -87.65113, -87.63541, -87.667~
## $ end_lat            <dbl> 41.92217, 41.92217, 41.89435, 41.88137, 41.92008, 4~
## $ end_lng            <dbl> -87.63889, -87.63889, -87.62280, -87.67493, -87.677~
## $ member_casual      <chr> "member", "casual", "member", "member", "casual", "~
## $ ride_length        <drtn> 556 secs, 1571 secs, 726 secs, 1741 secs, 851 secs~
## $ day_of_week        <dbl> 2, 1, 3, 3, 4, 6, 3, 3, 3, 4, 2, 3, 5, 5, 7, 5, 4, ~
```

Data can be analyzed now.

4) Analyze

```
glimpse(df)
```

```
## Rows: 5,677,610
## Columns: 15
## $ ride_id            <chr> "65DBD2F447EC51C2", "0C201AA7EA0EA1AD", "E0B148CCB3~
## $ rideable_type      <chr> "electric_bike", "classic_bike", "electric_bike", "~
## $ started_at         <dttm> 2022-12-05 10:47:18, 2022-12-18 06:42:33, 2022-12-~
## $ ended_at           <dttm> 2022-12-05 10:56:34, 2022-12-18 07:08:44, 2022-12-~
## $ start_station_name <chr> "Clifton Ave & Armitage Ave", "Broadway & Belmont A~
## $ start_station_id   <chr> "TA1307000163", "13277", "TA1306000015", "KA1503000~
## $ end_station_name   <chr> "Sedgwick St & Webster Ave", "Sedgwick St & Webster~
## $ end_station_id     <chr> "13191", "13191", "13016", "13134", "13288", "KA150~
## $ start_lat          <dbl> 41.91824, 41.94011, 41.88592, 41.83846, 41.89595, 4~
## $ start_lng          <dbl> -87.65711, -87.64545, -87.65113, -87.63541, -87.667~
## $ end_lat            <dbl> 41.92217, 41.92217, 41.89435, 41.88137, 41.92008, 4~
## $ end_lng            <dbl> -87.63889, -87.63889, -87.62280, -87.67493, -87.677~
## $ member_casual      <chr> "member", "casual", "member", "member", "casual", "~
## $ ride_length        <drtn> 556 secs, 1571 secs, 726 secs, 1741 secs, 851 secs~
## $ day_of_week        <dbl> 2, 1, 3, 3, 4, 6, 3, 3, 3, 4, 2, 3, 5, 5, 7, 5, 4, ~
```

The combined data has 5,677,610 data rows and 15 columns.

Data Cleaning

We need to check if there are any duplicate data and missing values.

```
df[duplicated(df$ride_id),]
```

```
## # A tibble: 0 x 15
## # i 15 variables: ride_id <chr>, rideable_type <chr>, started_at <dttm>,
## #   ended_at <dttm>, start_station_name <chr>, 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>,
## #   ride_length <drtn>, day_of_week <dbl>
```

There are no duplicated ride_id, hence there are no duplicated row.

```
sum(is.na(df))
```

```
## [1] 3597481
```

However, the data has 3,597,481 missing values.

```
missing_val <- sapply(df, function(col) sum(is.na(col)))
miss_val <- data.frame(
  missing_values = missing_val
)
miss_val
```

```
##                    missing_values
## ride_id                         0
## rideable_type                   0
## started_at                      0
## ended_at                        0
## start_station_name         869289
## start_station_id           869421
## end_station_name           922436
## end_station_id             922577
## start_lat                       0
## start_lng                       0
## end_lat                      6879
## end_lng                      6879
## member_casual                   0
## ride_length                     0
## day_of_week                     0
```

There are

- 869,289 missing values at `start_station_name` column
- 869,421 missing values at `start_station_id` column
- 922,436 missing values at `end_station_name` column
- 922,577 missing values at `end_station_id` column
- 6,879 missing values at `end_lat` column
- 6,879 missing values at `end_lng` column

We need to check how many rows with missing values at

- `start_station_name` and `start_station_id`
- `end_station_name` and `end_station_id`
- `end_lat` and `end_lng`

to see if imputing the missing values is necessary or not.

```
nrow(filter(df,
            is.na(start_station_name),
            is.na(start_station_id)))
```

```
## [1] 869289
```

```
nrow(filter(df,
            is.na(end_station_name),
            is.na(end_station_id)))
```

```
## [1] 922436
```

Seems like for every missing values at `station_name`, the `station_id` is also missing. The number of rows where `station_id` has missing values, but have values at `station_name` is also small (~136.5).

```
nrow(filter(df,
            !is.na(start_station_name),
            is.na(start_station_id)))
```

```
## [1] 132
```

```
nrow(filter(df,
            !is.na(end_station_name),
            is.na(end_station_id)))
```

```
## [1] 141
```

Every missing values at `end_lat` also have missing values at `end_lng`, so it is not missing at random.

```
nrow(filter(df,
            is.na(end_lat),
            is.na(end_lng)))
```

```
## [1] 6879
```

We can check if the `end_lat` and `end_lang` can be imputed using `end_station_name`. Imputing `end_lat` and `end_lng` is not necessary because the number of rows where they have `end_station_name` is small (116).

```
nrow(filter(df,
            is.na(end_lat),
            is.na(end_lng),
            !is.na(end_station_name)))
```

```
## [1] 116
```

So, imputing missing values is not necessary and can be dropped because most of the missing values cannot be imputed using another feature.

```


df <- na.omit(df)
```

Remove rows containing NA

Remove outliers Remove row with

- `ride_length` <= 0

```
df_cleaned <- df %>%
              filter(ride_length > 0)

nrow(df) - nrow(df_cleaned)
```

```
## [1] 554
```

Removed 554 rows.

Analysis

```
glimpse(df_cleaned)
```

Data Exploration

```
## Rows: 4,299,413
## Columns: 15
## $ ride_id            <chr> "65DBD2F447EC51C2", "0C201AA7EA0EA1AD", "E0B148CCB3~
## $ rideable_type      <chr> "electric_bike", "classic_bike", "electric_bike", "~
## $ started_at         <dttm> 2022-12-05 10:47:18, 2022-12-18 06:42:33, 2022-12-~
## $ ended_at           <dttm> 2022-12-05 10:56:34, 2022-12-18 07:08:44, 2022-12-~
## $ start_station_name <chr> "Clifton Ave & Armitage Ave", "Broadway & Belmont A~
## $ start_station_id   <chr> "TA1307000163", "13277", "TA1306000015", "KA1503000~
## $ end_station_name   <chr> "Sedgwick St & Webster Ave", "Sedgwick St & Webster~
## $ end_station_id     <chr> "13191", "13191", "13016", "13134", "13288", "KA150~
## $ start_lat          <dbl> 41.91824, 41.94011, 41.88592, 41.83846, 41.89595, 4~
## $ start_lng          <dbl> -87.65711, -87.64545, -87.65113, -87.63541, -87.667~
## $ end_lat            <dbl> 41.92217, 41.92217, 41.89435, 41.88137, 41.92008, 4~
## $ end_lng            <dbl> -87.63889, -87.63889, -87.62280, -87.67493, -87.677~
## $ member_casual      <chr> "member", "casual", "member", "member", "casual", "~
## $ ride_length        <drtn> 556 secs, 1571 secs, 726 secs, 1741 secs, 851 secs~
## $ day_of_week        <dbl> 2, 1, 3, 3, 4, 6, 3, 3, 3, 4, 2, 3, 5, 5, 7, 5, 4, ~
```

The combined data has 4,299,413 data rows and 15 columns after cleaning.

Let's see the average `ride_length` for members and casual riders.

```
df_cleaned %>%
  group_by(member_casual) %>%
  summarize(avg_ride_length=mean(ride_length))
```

```
## # A tibble: 2 x 2
##   member_casual avg_ride_length
##   <chr>         <drtn>
## 1 casual        1375.6177 secs
## 2 member         727.3618 secs
```

- casual rider have average ride length of 1375.6177 seconds (22.92696167 minutes)
- member rider have average ride length of 727.3618 seconds (12.1226967 minutes)

Now, for the average `ride_length`for users by `day_of_week`

```
df_cleaned %>%
  group_by(day_of_week) %>%
  summarize(avg_ride_length=mean(ride_length))
```

```
## # A tibble: 7 x 2
##   day_of_week avg_ride_length
##         <dbl> <drtn>
## 1           1 1170.5861 secs
## 2           2  897.8904 secs
## 3           3  851.2703 secs
## 4           4  834.7041 secs
## 5           5  848.3608 secs
## 6           6  948.5352 secs
## 7           7 1164.6140 secs
```

And for the number of rides by `day_of_week`

```
df_cleaned %>%
  group_by(day_of_week) %>%
  summarize(num_of_rides = n())
```

```
## # A tibble: 7 x 2
##   day_of_week num_of_rides
##         <dbl>        <int>
## 1           1       556363
## 2           2       559671
## 3           3       629871
## 4           4       628664
## 5           5       651003
## 6           6       619484
## 7           7       654357
```

5) Share

Distribution of Member and Casual rider in Cyclistic

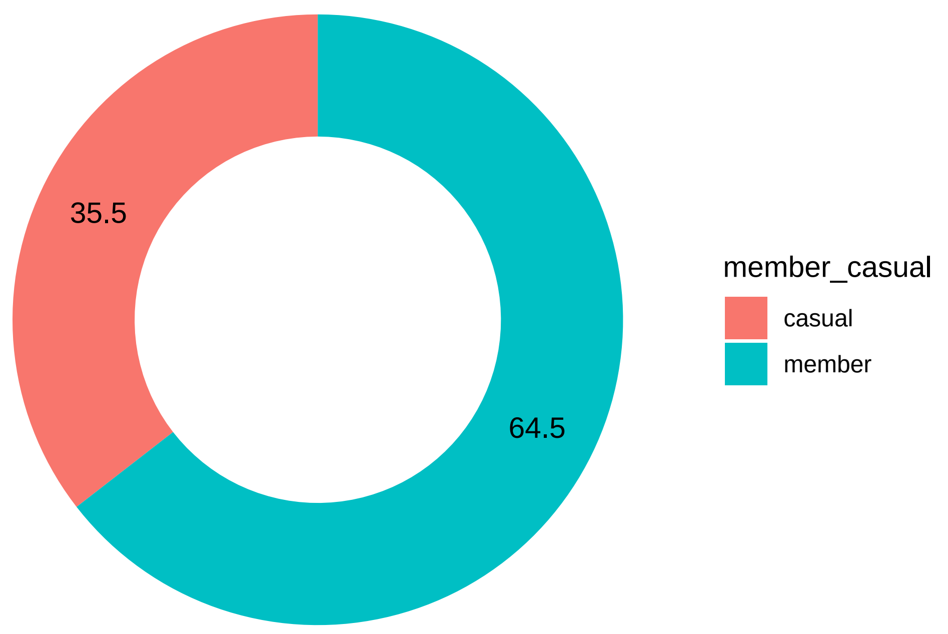
```
hsize <- 2
temp <- df_cleaned %>%
  group_by(member_casual) %>%
  summarise(val=n())

temp <- temp %>%
  mutate(x=hsize)

temp <- temp %>%
  mutate(val=signif(val/sum(val)*100, digits=3))

ggplot(data=temp, aes(x=hsize, y=val, fill=member_casual)) +
  geom_col() +
  coord_polar(theta="y") +
  xlim(c(0.2, hsize + 0.5)) +
  geom_text(aes(label = val),
            position = position_stack(vjust = 0.5)) +
  labs(title="Distribution of Member and Casual rider in Cyclistic (%)",
       caption="Data are taken from Dec 2022 - Nov 2023") +
  theme_void()
```

Distribution of Member and Casual rider in Cyclistic (%)



Data are taken from Dec 2022 − Nov 2023

Takeaway:

- 64.5% of number of rides are from member rider, while 35.5% are from casual rider.

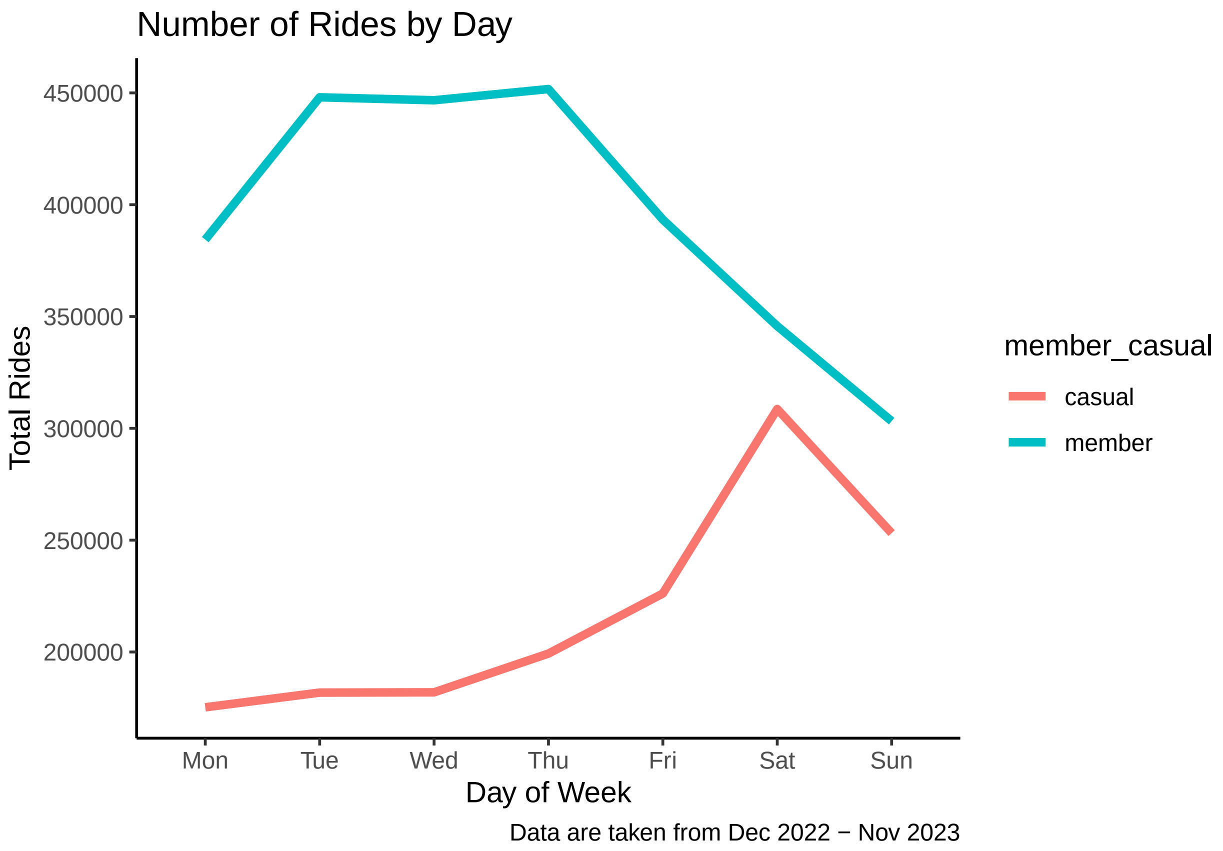
Number of Rides by Day

```
temp <- df_cleaned %>%
  group_by(member_casual, day_of_week) %>%
  summarize(total_rides=n())
```

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

```
temp <- temp %>%
  mutate(day=(day_of_week - 2) %% 7)

ggplot(data=temp) +
  geom_line(mapping=aes(x=factor(day), y=total_rides, group=member_casual, color=member_casual), linewi
  xlab("Day of Week") +
  ylab("Total Rides") +
  theme_classic() +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  labs(title="Number of Rides by Day",
       caption="Data are taken from Dec 2022 - Nov 2023") +
  scale_x_discrete(labels=c('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'))
```



Takeaway:

- Casual rider use the bike-share mostly on the weekend, while member rider on the weekday.

Average Ride Duration by Day

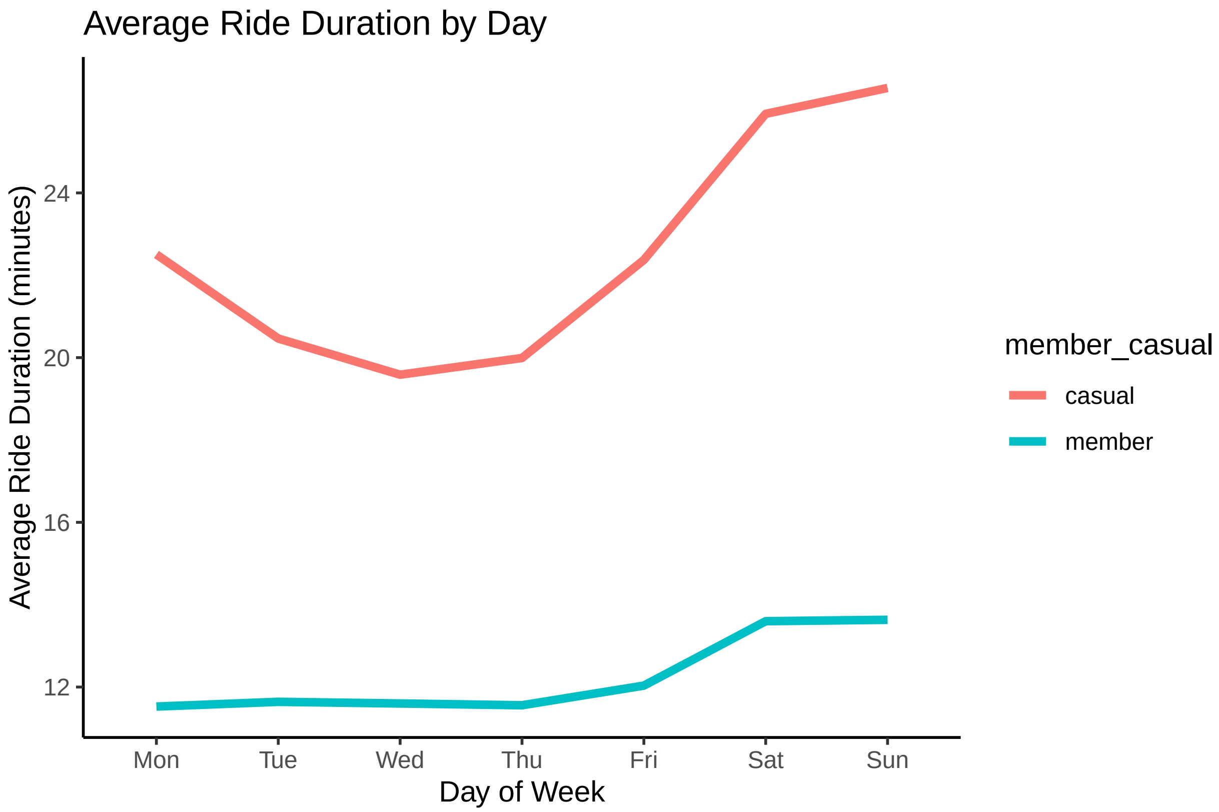
```
temp <- df_cleaned %>%
  group_by(member_casual, day_of_week) %>%
  summarize(avg_ride_length=mean(ride_length))
```

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

```
temp <- temp %>%
  mutate(day=(day_of_week - 2) %% 7)

ggplot(data=temp) +
  geom_line(mapping=aes(x=factor(day), y=avg_ride_length / 60, group=member_casual, color=member_casual
  xlab("Day of Week") +
  ylab("Average Ride Duration (minutes)") +
  theme_classic() +
  theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  labs(title="Average Ride Duration by Day",
       caption="Data are taken from Dec 2022 - Nov 2023") +
  scale_x_discrete(labels=c('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'))
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



Data are taken from Dec 2022 − Nov 2023

Takeaway:

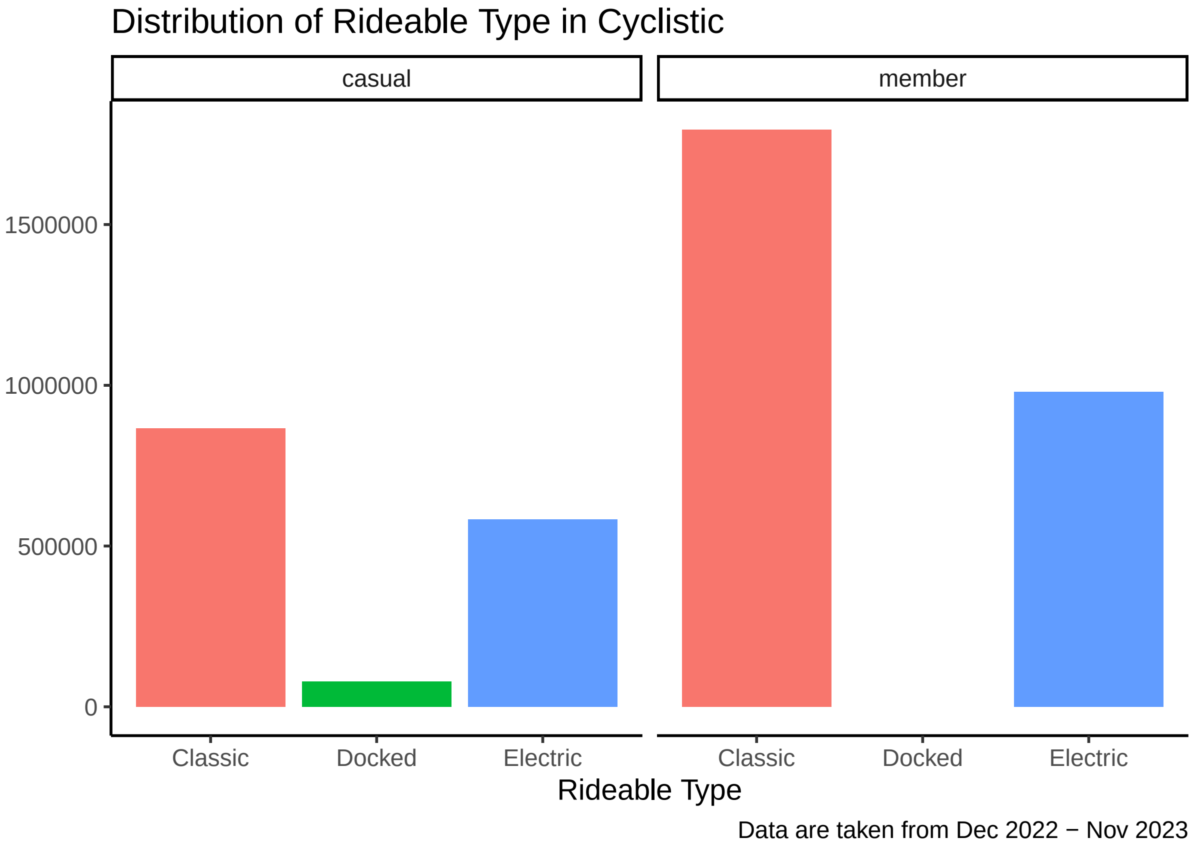
- Casual rider have longer average ride duration than member rider in Cyclistic.
- Casual rider average ride duration is lower on the weekday than on the weekend.
- Member rider average ride duration is stagnant on the weekday and higher on the weekend.

Distribution of Rideable Type in Cyclistic

```
temp <- df_cleaned %>%
  group_by(rideable_type, member_casual) %>%
  summarize(count_type=n())
```

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

```
ggplot(data=df_cleaned) +
  geom_bar(mapping=aes(x=rideable_type, fill=rideable_type), show.legend=FALSE) +
  labs(title="Distribution of Rideable Type in Cyclistic",
       caption="Data are taken from Dec 2022 - Nov 2023") +
  xlab("Rideable Type") +
  ylab("") +
  scale_x_discrete(labels=c('Classic', 'Docked', 'Electric')) +
  facet_wrap(~member_casual) +
  theme_classic()
```



Takeaway:

- The docked bike type has not been used by the member rider.
- The distribution is similar, from highest to lowest usage, classic bike, electric bike, and then docked bike.

Summary:

Casual rider Casual rider are the minority class on the Cyclistic bike-share program. Casual rider have a preference to use the bike-share on the weekend than on the weekday, their rate of using the bike-share decreases through the mid of the weekday and increases through the end of the weekday until the weekend. Compared to the member rider, they have approximately two times higher average ride duration. The distribution of rideable type they use is similar to the member rider.

Member rider Member rider are the majority class on the Cyclistic bike-share program. Member rider have a preference to use the bike-share on the weekend than on the weekday, their rate of using the bike-share plateaued on the weekday and increases at the weekend. Compared to the casual rider, they have a lower average ride duration.The distribution of rideable type they use is similar to the member rider.

6) Act

- Giving a discount for extended ride durations only for the members, could possibly encourage casual riders to join the membership and use it more often and also encourage members to ride for longer periods.