Google Data Analytics Capstone project

Cyclistic Case Study

Giovanni Silva

2025-03-26

Introdution

This case study involves data analysis for the fictional company Cyclistic, a bike-sharing service. The goal is to answer business questions by following the data analysis process: Ask, Prepare, Process, Analyze, Share, and Act. Throughout the study, roadmap tables will guide the steps with key questions and tasks.

The scenario

Company: Cyclistic, a bike-share service in Chicago.

Goal: Increase annual memberships by converting casual riders into subscribers.

Role of the Analyst: You are a junior analyst on the marketing analytics team.

Cyclistic's Differentiation:

1. Over 5,800 bikes and 600 docking stations.
2. Offers inclusive bikes, such as hand tricycles and reclining bikes (used by 8% of customers).
3. 30% of users commute to work daily using Cyclistic bikes.

Teams and Characters:

1. Lily Moreno: Marketing director responsible for promotional campaigns.
2. Marketing Analytics Team: Collects and analyzes data to support the company's strategies.
3. Executive Team: Reviews data and decides whether to approve the new marketing program.

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?

Moreno has assigned you the first question to answer: How do annual members and casual

Prepare

The data from the first half of 2021 for the bike-sharing service was extracted from **01/01/2021 to 30/06/2021** in **12 compressed .csv files**.
The data is available and licensed by Motivate International Inc under this license.

Data Organization & Description:

- **File naming convention:** `YYYY_MM`
- **File type:** `.csv` format
- **File content:**
  - Each file contains **13 columns** with information such as **ride ID, rider type, ride start and end time, start and end location**, among others.
  - The number of rows varies between **49,000 and 729,000** in different Excel files.

Data Credibility:

The dataset is reliable, complete, and accurate for the chosen time period.
- **The data is original:** It is first-party information.
- **The data is comprehensive:** The dataset contains all necessary information to answer the question.
- **The data is current:** Rider data from the last 12 months was used.
- **The data is cited and vetted:** Reviewed by the Chicago Department of Transportation.

Data Security:

Riders' personal identifiable information is hidden through tokenization.

- **Original files** are backed up in a separate folder.

Data Limitations:

As riders' personal identifiable information is hidden, it will not be possible to connect past purchases to credit card numbers to determine if casual riders live in the Cyclistic service area or if they have purchased multiple single passes.

Analyzis

For data cleaning and preparation, I will use **RStudio**. Due to the large volume of data, **Excel** is not suitable for handling the dataset. Additionally, to unify these tables for analysis, **BigQuery** is not supported. RStudio provides an efficient environment to process and manipulate large datasets, making it the ideal tool for this task.

```
library(tidyverse)
library(janitor)
library(skimr)
library(here)
library(hablar)
library(readxl)
library(data.table)
library(chron)
library(readr)
library(lubridate)
library(magrittr)
library(DescTools)
library(metR)
```

Importing data Cyclist data from 01/2021 until 06/2021 is imported and read as csv. files.

```r
# Usando o caminho absoluto
data_01 <- read_csv("/Users/giovannisilva/Downloads/Cyclistic/Dados_cvs/2021_01.csv")
data_02 <- read_csv("/Users/giovannisilva/Downloads/Cyclistic/Dados_cvs/2021_02.csv")
data_03 <- read_csv("/Users/giovannisilva/Downloads/Cyclistic/Dados_cvs/2021_03.csv")
data_04 <- read_csv("/Users/giovannisilva/Downloads/Cyclistic/Dados_cvs/2021_04.csv")
data_05 <- read_csv("/Users/giovannisilva/Downloads/Cyclistic/Dados_cvs/2021_05.csv")
data_06 <- read_csv("/Users/giovannisilva/Downloads/Cyclistic/Dados_cvs/2021_06.csv")
```

Before merging the files into one, it is important to compare the column names of each. While the order of the columns doesn't need to be the same, **the names must match perfectly**.

```r
colnames(data_01)
```

```
##  [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"
```

```r
colnames(data_02)
```

```
##  [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"
```

```r
colnames(data_04)
```

```
##  [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"
```

```r
colnames(data_05)
```

```
##  [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"
```

```r
colnames(data_06)
```

```
##  [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"
```

Combine all tables into one data frame:

```
all_trips <- bind_rows(data_01, data_02, data_03, data_04, data_05, data_06)
```

Data Cleaning for Analysis Start

Inspect the new table that has been created.

```
colnames(all_trips)  #List of column names
```

```
##  [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"
```

```
dim(all_trips)  #Dimensions of the data frame
```

```
## [1] 1973410      13
```

```
head(all_trips)  #See the first 6 rows of data frame
```

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 E19E6F1B8D4C42ED electric_bike 2021-01-23 16:14:19 2021-01-23 16:24:44
## 2 DC88F20C2C55F27F electric_bike 2021-01-27 18:43:08 2021-01-27 18:47:12
## 3 EC45C94683FE3F27 electric_bike 2021-01-21 22:35:54 2021-01-21 22:37:14
## 4 4FA453A75AE377DB electric_bike 2021-01-07 13:31:13 2021-01-07 13:42:55
## 5 BE5E8EB4E7263A0B electric_bike 2021-01-23 02:24:02 2021-01-23 02:24:45
## 6 5D8969F88C773979 electric_bike 2021-01-09 14:24:07 2021-01-09 15:17:54
## # i 9 more variables: 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>
```

```
str(all_trips)  #See list of columns and data types (numeric, character, etc)
```

```
## spc_tbl_ [1,973,410 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1973410] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA
##  $ rideable_type     : chr [1:1973410] "electric_bike" "electric_bike" "electric_bike" "electric_bik
##  $ started_at        : POSIXct[1:1973410], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" ...
##  $ ended_at          : POSIXct[1:1973410], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" ...
##  $ start_station_name: chr [1:1973410] "California Ave & Cortez St" "California Ave & Cortez St" "Cal
##  $ start_station_id  : chr [1:1973410] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:1973410] NA NA NA NA ...
##  $ end_station_id    : chr [1:1973410] NA NA NA NA ...
##  $ start_lat         : num [1:1973410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:1973410] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:1973410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:1973410] -87.7 -87.7 -87.7 -87.7 -87.7 ...
```

```
##  $ member_casual     : chr [1:1973410] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Adding columns for date-related information

Extracting year, month, day, and day of the week from the ride start time

```
all_trips$date <- as.Date(all_trips$started_at) #The default format is yyyy-mm-dd
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%u") #"%A" would deliver names of weekdays
```

Add a "ride_length" calculation to all_trips in seconds and in minutes

```
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
all_trips$ride_length_m <- (difftime(all_trips$ended_at,all_trips$started_at))/60
```

Check the structure of the newly added columns

```
str(all_trips)
```

```
## spc_tbl_ [1,973,410 x 20] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1973410] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA
##  $ rideable_type     : chr [1:1973410] "electric_bike" "electric_bike" "electric_bike" "electric_bik
##  $ started_at        : POSIXct[1:1973410], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" ...
##  $ ended_at          : POSIXct[1:1973410], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" ...
##  $ start_station_name: chr [1:1973410] "California Ave & Cortez St" "California Ave & Cortez St" "Ca
##  $ start_station_id  : chr [1:1973410] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:1973410] NA NA NA NA ...
##  $ end_station_id    : chr [1:1973410] NA NA NA NA ...
##  $ start_lat         : num [1:1973410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:1973410] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:1973410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:1973410] -87.7 -87.7 -87.7 -87.7 -87.7 ...
```

```
##  $ member_casual     : chr [1:1973410] "member" "member" "member" "member" ...
##  $ date              : Date[1:1973410], format: "2021-01-23" "2021-01-27" ...
##  $ month             : chr [1:1973410] "01" "01" "01" "01" ...
##  $ day               : chr [1:1973410] "23" "27" "21" "07" ...
##  $ year              : chr [1:1973410] "2021" "2021" "2021" "2021" ...
##  $ day_of_week       : chr [1:1973410] "6" "3" "4" "4" ...
##  $ ride_length       : 'difftime' num [1:1973410] 625 244 80 702 ...
##   ..- attr(*, "units")= chr "secs"
##  $ ride_length_m     : 'difftime' num [1:1973410] 10.4166666666667 4.06666666666667 1.33333333333333
##   ..- attr(*, "units")= chr "secs"
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Convert ride_length, ride_length_m, day, and month to numeric

```
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
all_trips$ride_length_m <- as.numeric(as.character(all_trips$ride_length_m))
all_trips$month <- as.numeric(all_trips$month)
all_trips$day <- as.numeric(all_trips$day)
is.numeric(all_trips$ride_length)
```

```
## [1] TRUE
```

```
is.numeric(all_trips$ride_length_m)
```

```
## [1] TRUE
```

```
is.numeric(all_trips$month)
```

```
## [1] TRUE
```

```
is.numeric(all_trips$day)
```

```
## [1] TRUE
```

Remove rows where ride_length is negative

```r
all_trips_v1 <- all_trips[!( all_trips$ride_length < 0),]
```

1ª Analysis step: Descriptive statistics for ride length (in minutes)

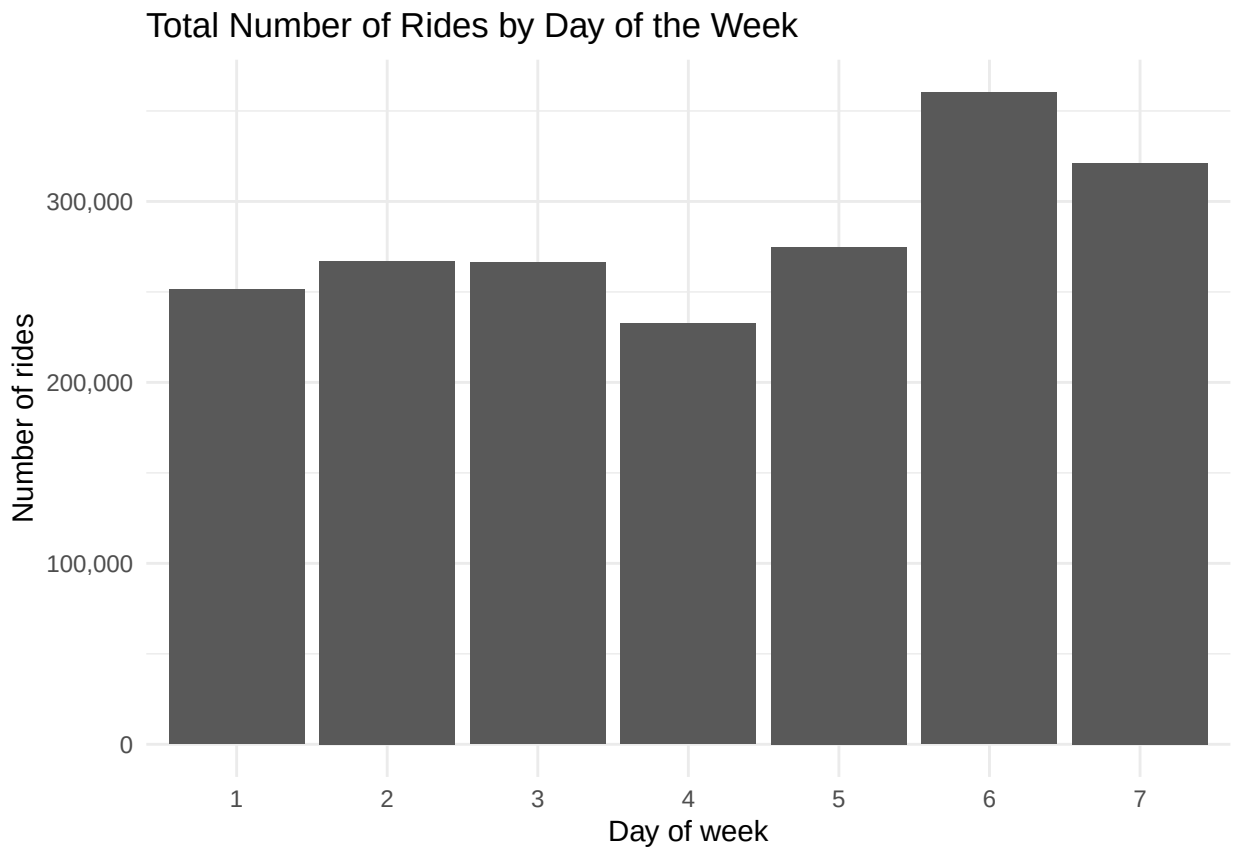
```r
all_trips_v1 %>%
  summarise(max(ride_length_m),min(ride_length_m),mean(ride_length_m))
```

```
## # A tibble: 1 x 3
##   `max(ride_length_m)` `min(ride_length_m)` `mean(ride_length_m)`
##                  <dbl>                <dbl>                 <dbl>
## 1               55944.                    0                  24.8
```

The overall average ride length is 24.8 minutes.

Calculate the most common weekday

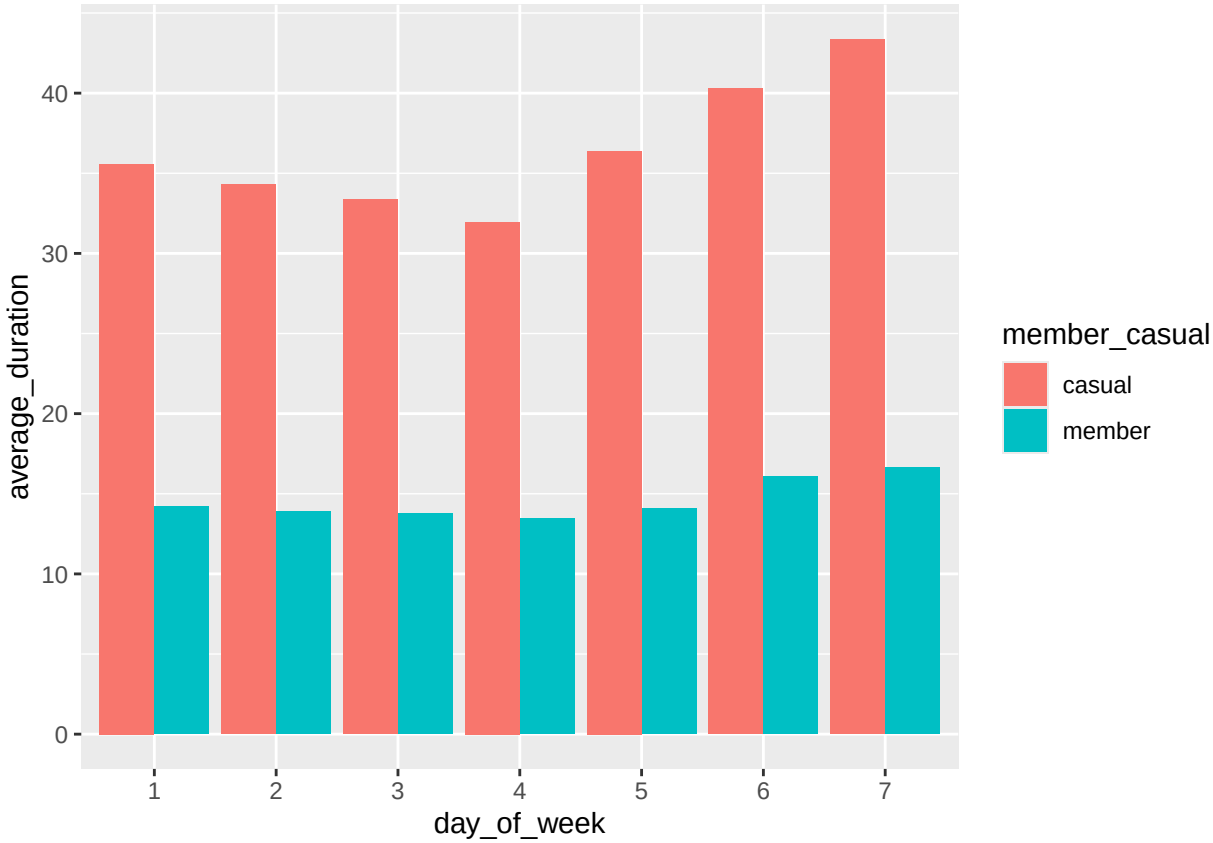
```r
  all_trips_v1 %>%
  group_by(day_of_week) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(mapping = aes(x = day_of_week, y = number_of_rides)) + geom_col()+
  scale_y_continuous(labels = scales::comma) + # Remove scientific notation from the Y axis
  labs(title = "Total Number of Rides by Day of the Week",
       x = "Day of week",
       y = "Number of rides") +
  theme_minimal()
```



The graph shows that the days with the highest number of rides are weekends: **Saturday** comes first and **Sunday** follows.

"Next, a plot of the **average_duration or ride_length** (in minutes) for each day of the week, comparing members and casual riders, is shown.

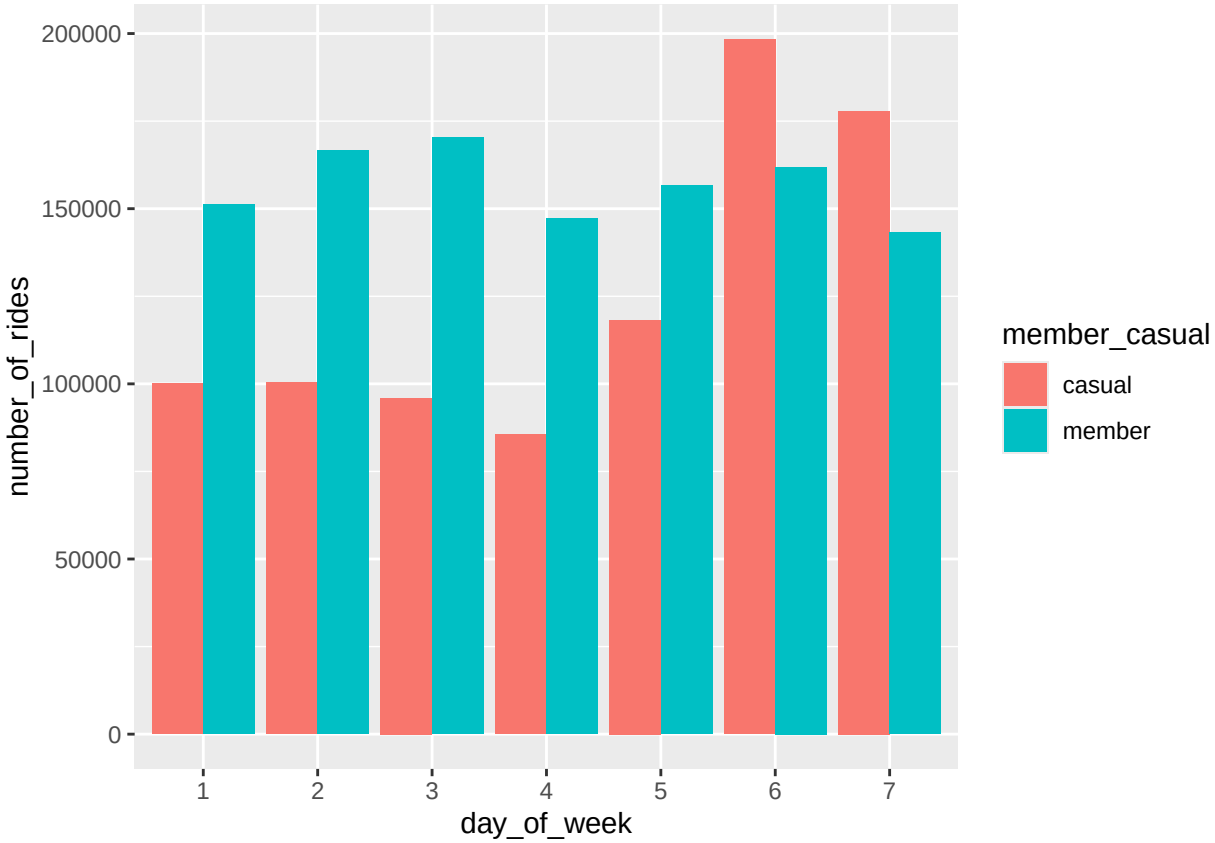
```
all_trips_v1 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length_m),.groups = "drop") %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```



The graph shows that casual riders use their bikes much longer than members. The most popular days for casual riders are the weekends. Members also tend to use the bikes more on weekends, but no day of the week exceeds a 20% higher average duration.

Here, number of rides per day for every rider type is plotted.

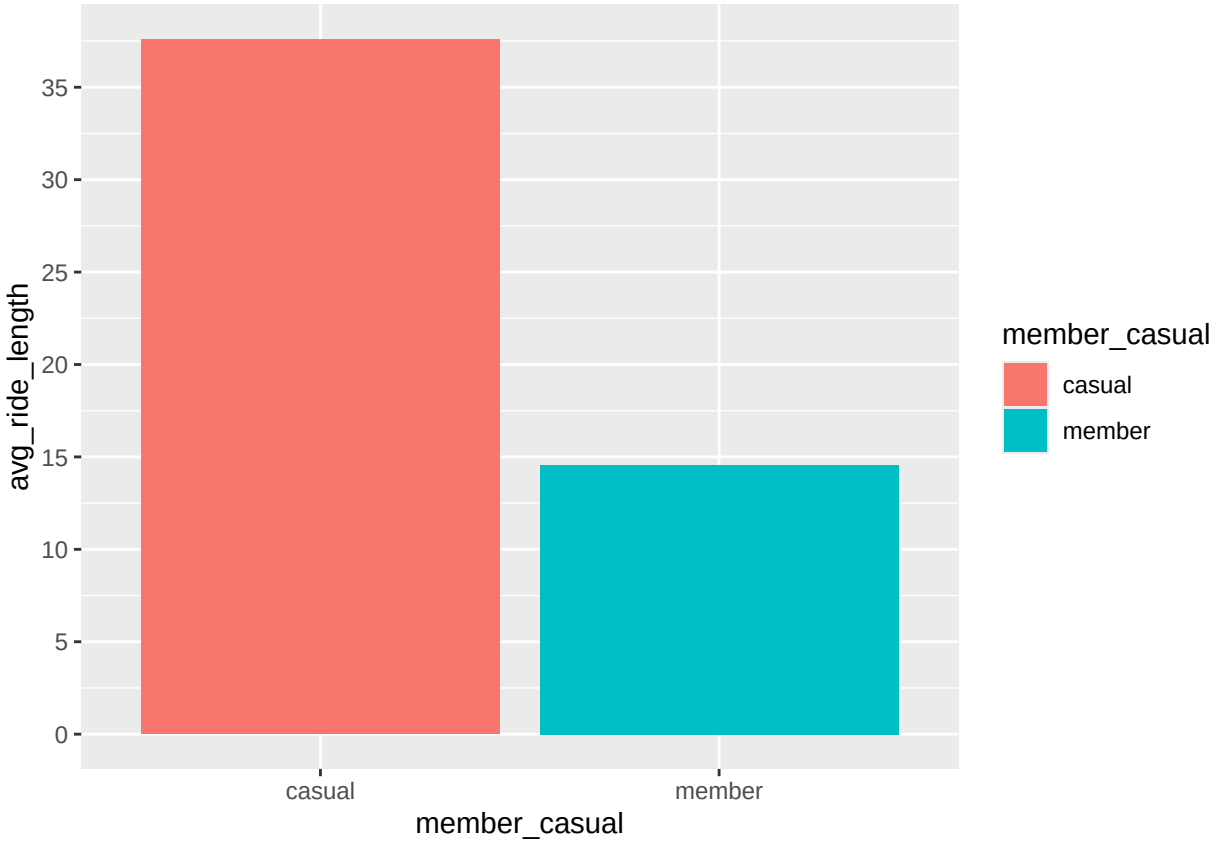
```
all_trips_v1 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length_m),.groups = "drop") %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```



Unlike the previous graph, it is observed that the number of rides taken by members is significantly higher on weekdays. However, on weekends, casual users take the lead, recording a higher volume of rides. This suggests that members primarily use the bikes as a daily mode of transportation, while casual users tend to use them more for leisure on their days off.

2ª Analysis step: Average ride duration and distribution by rider type.

```
all_trips_v1 %>%
  group_by(member_casual) %>%
  summarise(max(ride_length_m), min(ride_length_m),avg_ride_length = mean(ride_length_m)) %>%
  ggplot(aes(x = member_casual, y = avg_ride_length,fill=member_casual)) +
  geom_col()+ scale_y_continuous(breaks = seq(0, 40, by = 5))
```



Thus, the results show that casual riders tend to rent bikes for a longer average duration than members (37 minutes versus 15 minutes), in accordance with Graph 2. Members likely use bikes for daily commuting, while casual riders may be exercising, sightseeing, or attending special events, among other activities.

Here, a chart is presented showing the total rider count, categorized by rider type.

```
all_trips_v1 %>%
  group_by(member_casual) %>%
  summarise(rider_count = n()) %>%
  ggplot(aes(x = member_casual, y = rider_count,fill=member_casual )) +
  geom_col() +
  scale_y_continuous(labels = scales::comma) +  # Remove scientific notation from the Y axis
  labs(title = "Total Number of Rides by Rider Type",
       x = "Rider Type",
       y = "Number of Rides") +
  theme_minimal()
```

The chart shows that, relative to the total number of bicycle users, members make up a larger proportion compared to casual users, with a 10% higher share for members.

3ª Analysis Step: Exploring the Effect of Seasonality

Here, the Function "season" of the library "metR" was used to assign season to months:

DJF:winter

MAM:Spring

JJA:Summer

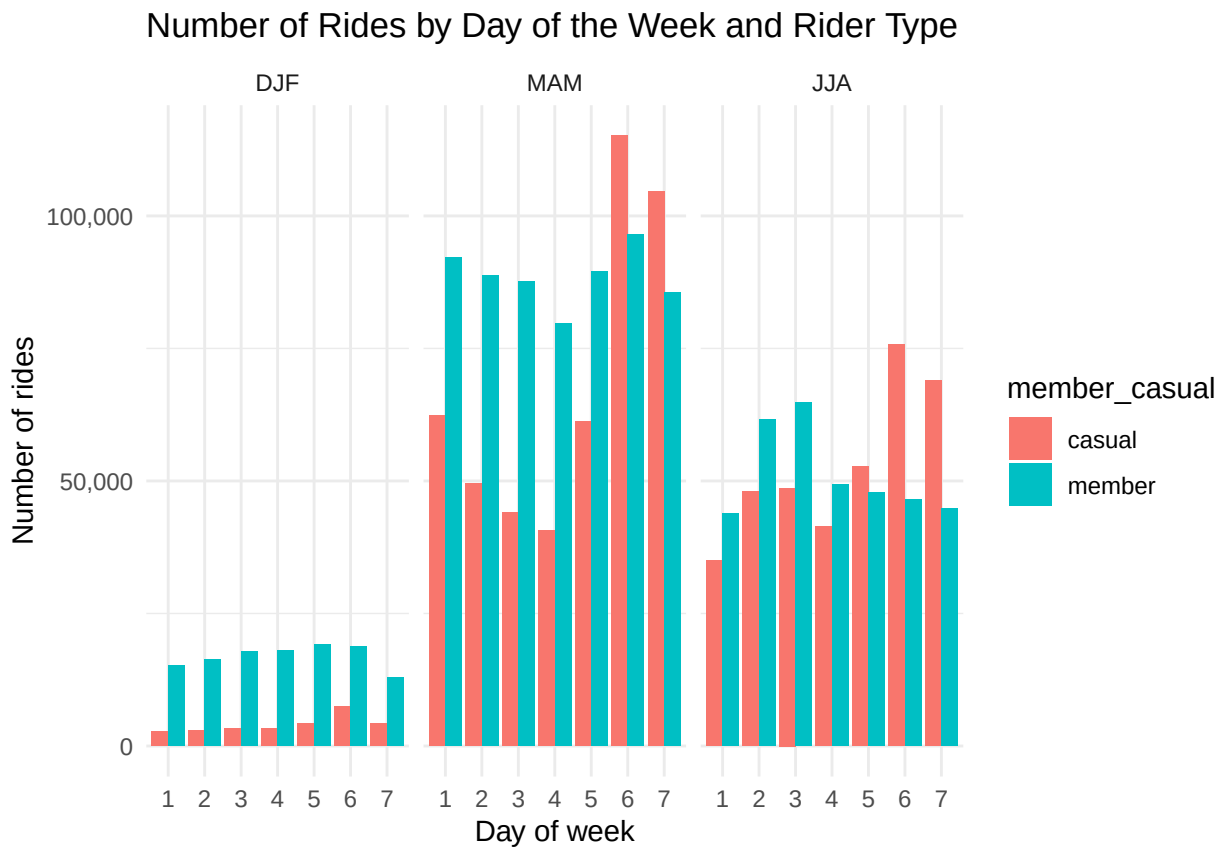
SON:Fall

```
all_trips_v1$season <- season(all_trips_v1$month)
```

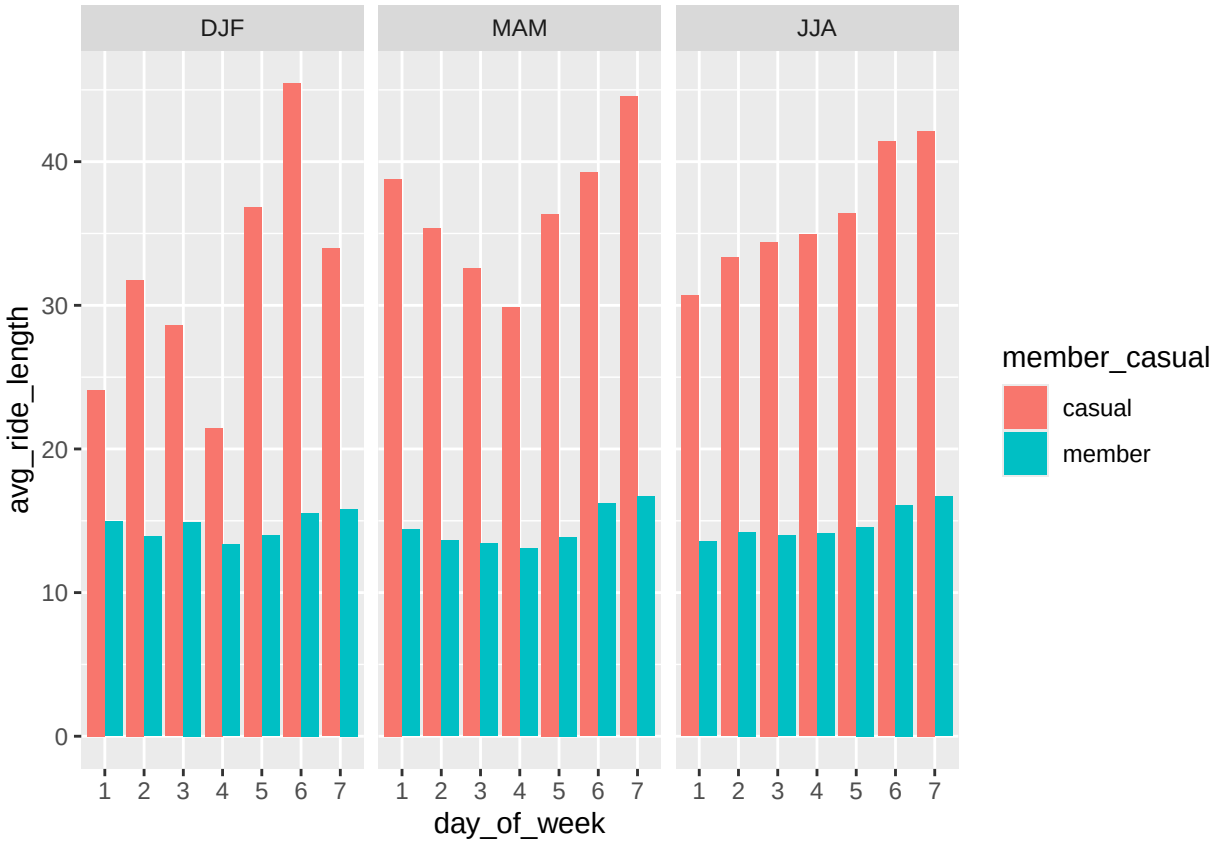
First, let us consider number of rides and ride length by weekday on each season

```
all_trips_v1 %>%
  group_by(season, day_of_week, member_casual) %>%
  summarise(number_of_rides = n(),
            avg_ride_length = mean(ride_length_m, na.rm = TRUE),
            .groups = "drop") %>%  # <- Adicionado para remover o agrupamento
  ggplot() +
  geom_col(mapping = aes(x = day_of_week, y = number_of_rides, fill = member_casual), position = "dodge
  facet_wrap(~season) +
```

```
  scale_y_continuous(labels = scales::comma, breaks = seq(0, 400000, by = 50000)) +
  labs(title = "Number of Rides by Day of the Week and Rider Type",
       x = "Day of week",
       y = "Number of rides") +
  theme_minimal()
```




```
all_trips_v1%>%
  group_by(season, day_of_week, member_casual) %>%
  summarise(number_of_rides = n()
  ,avg_ride_length = mean(ride_length_m),.groups="drop") %>%
ggplot() + geom_col(mapping = aes(x = day_of_week, y = avg_ride_length, fill = member_casual), position
```

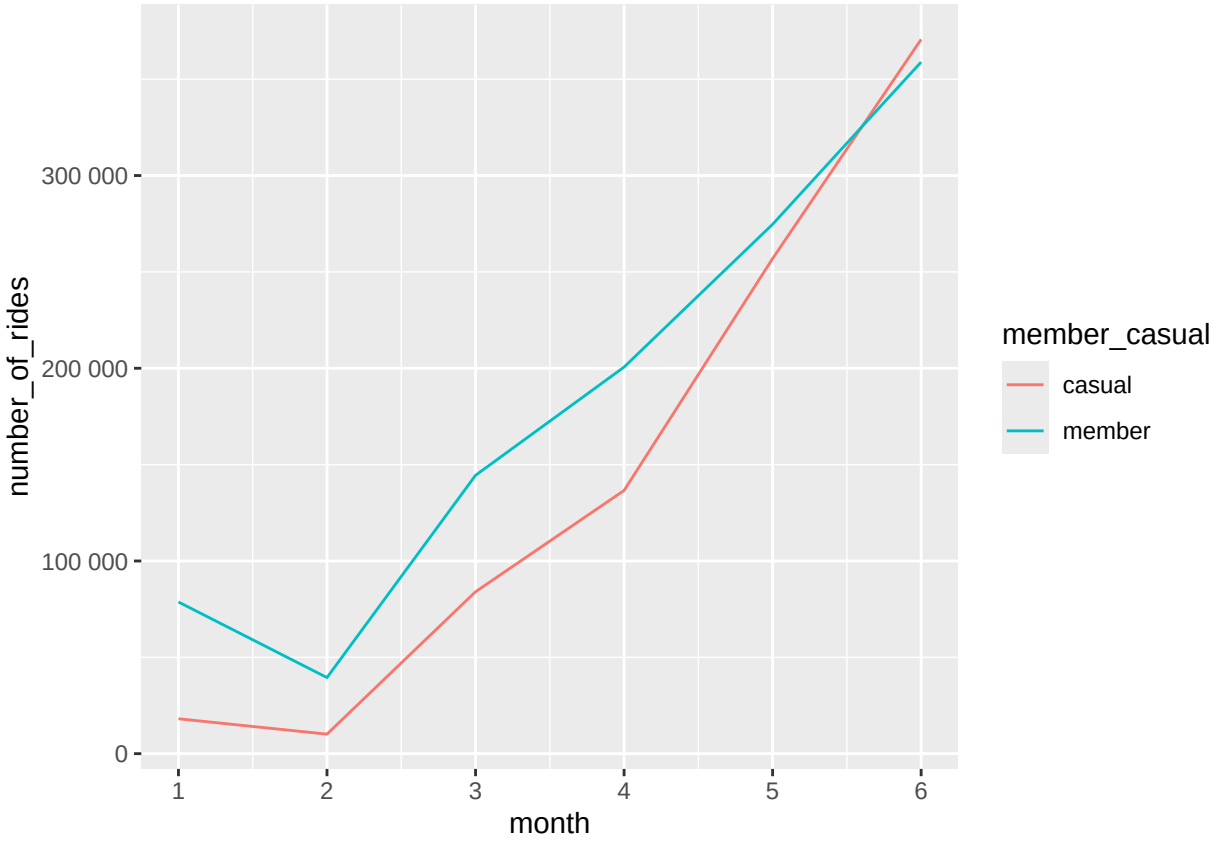


In Chart 6, it is observed that during the spring, members dominate on weekdays, while on weekends, the majority of riders are casual. In the summer, the number of rides per user starts to balance out: from Monday to Thursday, members take the lead, while on Fridays and weekends, casual riders dominate. In the winter, the number of rides drops drastically, and members stand out on all seven days of the week.

In Chart 7, during the spring, the average ride duration for casual users across all seven days of the week is 36.28 minutes, with Sunday standing out as having the highest average within this group, close to 44 minutes. In contrast, members have a much lower average of 12.42 minutes, with Sunday also showing the highest average of the week at 16 minutes. In the summer, the difference remains consistent, with casual riders dominating, and the values are very similar to those in spring. In winter, the average for casual users decreases to 31.28 minutes across all seven days, with Saturdays having the highest average of the week at about 45 minutes. Surprisingly, members have an average of 12.71 minutes across all seven days, which is higher than their average in both spring and summer.

Lastly, let us generate a line plot for continuous change of number of rides along the whole year for the two rider types.

```
all_trips_v1 %>%
  group_by(month, member_casual) %>%
  summarise(number_of_rides = n(),
            avg_ride_length = mean(ride_length_m), .groups = "drop") %>%
  ggplot() +
  geom_line(mapping = aes(x = month, y = number_of_rides, color = member_casual)) +
  scale_x_continuous(breaks = seq(1, 12, by = 1)) +
  scale_y_continuous(labels = label_number(accuracy = 1)) # Remove scientific notation from the Y axis
```



The ascending chart shows that, month by month, until June, the number of users, both casual and member, only increases compared to the start of the year, which is January. Other key details from the chart include the point where the two lines intersect in May, indicating that the number of rides for casual users and members becomes equal in that month. Additionally, it can be observed that, from January to April, the number of rides for members is higher, with the exception of June, when the number of rides for casual riders surpasses that of members.

Share

Conclusion

Based on the analyses from the charts, we can conclude that annual members and casual riders use Cyclistic bikes differently, reflecting their distinct usage patterns.

1. Days and Frequency of Use: Members tend to use the bikes more during weekdays, primarily for daily commuting, while casual riders dominate the weekends, likely for leisure activities. However, weekends are also popular among members, although their usage is more limited compared to casual riders, who tend to rent bikes for longer durations.
2. Duration of Rides: Casual riders tend to use the bikes for longer periods, with an average of 37 minutes per ride, while members typically use bikes for 15 minutes, suggesting that members use them for quicker tasks, such as commuting to work.
3. User Proportions: Members represent a larger proportion of the total number of users, with 10% more participation than casual riders.

4. Seasonal Patterns: During spring and summer, members dominate during weekdays, while casual riders are more prominent on weekends. In winter, the number of rides drops drastically, but members still dominate.

5. Ride Duration by Season: Throughout all seasons, casual riders tend to have longer ride durations compared to members, especially on weekends.

6. Monthly Usage Patterns: Starting in May, the number of rides between members and casual riders equalizes, with casual riders surpassing members in rides in June.

Question to answer: How do annual members and casual riders use Cyclistic bikes differently?

Answer to the Question: Cyclistic annual members primarily use the bikes for daily commuting during weekdays, with shorter and more frequent rides. On the other hand, casual riders use the bikes more for leisure on weekends, with longer and less frequent rides. This difference in behavior can be leveraged to create targeted marketing and loyalty strategies for both user groups.

Recommendations

Based on the conclusions about the behavior of annual members and casual riders, here are some recommendations that can enhance the experience for both user groups and optimize Cyclistic's operations:

1. Offer Flexible Subscription Plans for Casual Riders:

Objective: Leverage the more sporadic usage pattern of casual riders to encourage them to become members.

Recommendation: Create short-term subscription plans or special packages for casual riders, such as a weekend package or monthly offers. This can encourage greater commitment and generate recurring revenue.

2. Weekend Promotions:

Objective: Increase bike usage on weekends when casual riders dominate.

Recommendation: Offer special weekend promotions, such as discounts for long rides or a rewards system for those renting bikes during peak hours, aiming to increase demand and drive more traffic from casual riders.

3. Develop Member Convenience Features:

Objective: Make daily use more convenient for members, who often use bikes during the week.

Recommendation: Implement faster and more comfortable bikes, or even quick maintenance services for members. Offering priority service or personalized assistance can increase member satisfaction and loyalty.

4. Seasonal Marketing Campaigns:

Objective: Leverage seasonal usage patterns to attract more riders.

Recommendation: Create seasonal marketing campaigns targeting each time of year. During spring and summer, focus on promotions for casual riders seeking leisure, such as packages for sightseeing tours. In winter, highlight bike usage as a practical transportation solution for members continuing their daily routines despite colder temperatures.

5. Improve Infrastructure and Support for Casual Riders:

Objective: Make the casual rider experience more attractive and accessible.

Recommendation: Enhance bike availability on weekends, especially in tourist areas and leisure spots, to meet the high demand from casual users. Additionally, offer guidance on bike usage and create tour guides, attracting more casual users to the service.

6. Data Analysis to Identify Usage Trends:

Objective: Adjust operations based on behavioral trends.

Recommendation: Continuously analyze usage data to identify patterns and shifts in user behavior. This could include tracking the rise of members during certain months or adjusting pricing during specific hours to maximize usage and user satisfaction.

7. Offer Personalized Experiences:

Objective: Build a deeper connection with users, both casual and members.

Recommendation: Create personalized experiences based on user profiles, such as route suggestions for casual riders and weekly challenges or rewards for members. These actions can generate higher engagement and increase brand loyalty.