Google Analytics Capstone : Bike Sharing Case Study

Russhel De Leon

3/10/2022

How Does a Bike-Share Navigate Speedy Success?

Google Analytics Capstone

Scenario 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.

Ask

The business task is to better understand how annual members and casual riders differ. A descriptive analysis will help them to understand the difference between the two types of riders and to develop a new marketing strategy in converting casual riders into annual members.

Prepare

The data source can be found at https://divvy-tripdata.s3.amazonaws.com/index.html. Data is structured and organized in a wide format where each row is an observation and each column is a variable.

The datasets have a different name because Cyclistic is a fictional company. For the purposes of this case study, the datasets are appropriate and will enable us to answer the business questions. The data has been made available by Motivate International Inc. under this license: https://www.divvybikes.com/data-license-agreement.

Process

This step involves examining and cleaning the data. The cleaning involves standardize each column type, making sure that each column is consistent, removing duplicates and etc. The files are large and consist of million rows. RStudio is a flexible tool that will help us to clean and analyze these files.

Step 1: Setting up our environment.

Tidyverse helps us wrangle data.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     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()
```

Lubridate helps us to format and clean date.

```
library(lubridate)
```

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

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

Janitor package helps us to clean column.

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

Step 2: Collect data.

```
trip202112 <- read.csv("raw_data/202112-divvy-tripdata.csv")
trip202111 <- read.csv("raw_data/202111-divvy-tripdata.csv")
trip202110 <- read.csv("raw_data/202110-divvy-tripdata.csv")
trip202109 <- read.csv("raw_data/202109-divvy-tripdata.csv")
trip202108 <- read.csv("raw_data/202108-divvy-tripdata.csv")
trip202107 <- read.csv("raw_data/202107-divvy-tripdata.csv")
trip202106 <- read.csv("raw_data/202106-divvy-tripdata.csv")
trip202105 <- read.csv("raw_data/202105-divvy-tripdata.csv")
trip202104 <- read.csv("raw_data/202104-divvy-tripdata.csv")
trip202103 <- read.csv("raw_data/202103-divvy-tripdata.csv")
trip202102 <- read.csv("raw_data/202102-divvy-tripdata.csv")
trip202101 <- read.csv("raw_data/202101-divvy-tripdata.csv")
```

Step 3: Combine data into single file.

Examine column name and type for each dataframe.

```
compare_df_cols(trip202101, trip202102, trip202103, trip202104, trip202105, trip202106, trip202107, tri
                trip202109, trip202110, trip202111, trip202112)
```

```
##           column_name trip202101 trip202102 trip202103 trip202104 trip202105
## 1            end_lat    numeric    numeric    numeric    numeric    numeric
## 2            end_lng    numeric    numeric    numeric    numeric    numeric
## 3     end_station_id  character  character  character  character  character
## 4   end_station_name  character  character  character  character  character
## 5           ended_at  character  character  character  character  character
## 6      member_casual  character  character  character  character  character
## 7            ride_id  character  character  character  character  character
## 8      rideable_type  character  character  character  character  character
## 9          start_lat    numeric    numeric    numeric    numeric    numeric
```

```
## 10          start_lng    numeric    numeric    numeric    numeric    numeric
## 11   start_station_id  character  character  character  character  character
## 12 start_station_name  character  character  character  character  character
## 13         started_at  character  character  character  character  character
##    trip202106 trip202107 trip202108 trip202109 trip202110 trip202111 trip202112
## 1     numeric    numeric    numeric    numeric    numeric    numeric    numeric
## 2     numeric    numeric    numeric    numeric    numeric    numeric    numeric
## 3   character  character  character  character  character  character  character
## 4   character  character  character  character  character  character  character
## 5   character  character  character  character  character  character  character
## 6   character  character  character  character  character  character  character
## 7   character  character  character  character  character  character  character
## 8   character  character  character  character  character  character  character
## 9     numeric    numeric    numeric    numeric    numeric    numeric    numeric
## 10    numeric    numeric    numeric    numeric    numeric    numeric    numeric
## 11  character  character  character  character  character  character  character
## 12  character  character  character  character  character  character  character
## 13  character  character  character  character  character  character  character
```

Verify if the all file are readily bindable.

```
compare_df_cols_same(trip202101, trip202102, trip202103, trip202104, trip202105, trip202106, trip202107
                     trip202109, trip202110, trip202111, trip202112)
```

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

Stack monthly's dataframes into one big dataframe.

```
all_trip <- bind_rows(trip202101, trip202102, trip202103, trip202104, trip202105, trip202106, trip20210
                      trip202109, trip202110, trip202111, trip202112)
```

Step 4: Inspect new dataframe created.

str () Shows column type and observation per column.

```
str(all_trip)
```

```
## 'data.frame':    5595063 obs. of  13 variables:
##  $ ride_id           : chr  "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA453A75AE377
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2021-01-23 16:14:19" "2021-01-27 18:43:08" "2021-01-21 22:35:54" "2021-
##  $ ended_at          : chr  "2021-01-23 16:24:44" "2021-01-27 18:47:12" "2021-01-21 22:37:14" "2021-
##  $ start_station_name: chr  "California Ave & Cortez St" "California Ave & Cortez St" "California Av
##  $ start_station_id  : chr  "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

Inspect the first 6 rows of the table

```
head(all_trip)
```

```
##            ride_id rideable_type          started_at            ended_at
## 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
##           start_station_name start_station_id end_station_name end_station_id
## 1 California Ave & Cortez St            17660
## 2 California Ave & Cortez St            17660
## 3 California Ave & Cortez St            17660
## 4 California Ave & Cortez St            17660
## 5 California Ave & Cortez St            17660
## 6 California Ave & Cortez St            17660
##   start_lat start_lng end_lat end_lng member_casual
## 1  41.90034 -87.69674   41.89  -87.72        member
## 2  41.90033 -87.69671   41.90  -87.69        member
## 3  41.90031 -87.69664   41.90  -87.70        member
## 4  41.90040 -87.69666   41.92  -87.69        member
## 5  41.90033 -87.69670   41.90  -87.70        casual
## 6  41.90041 -87.69676   41.94  -87.71        casual
```

Inspect the last 6 rows of the table

```
tail(all_trip)
```

```
##                  ride_id rideable_type          started_at            ended_at
## 5595058 92BBAB97D1683D69 electric_bike 2021-12-24 15:42:09 2021-12-24 19:29:35
## 5595059 847431F3D5353AB7 electric_bike 2021-12-12 13:36:55 2021-12-12 13:56:08
## 5595060 CF407BBC3B9FAD63 electric_bike 2021-12-06 19:37:50 2021-12-06 19:44:51
## 5595061 60BB69EBF5440E92 electric_bike 2021-12-02 08:57:04 2021-12-02 09:05:21
## 5595062 C414F654A28635B8 electric_bike 2021-12-13 09:00:26 2021-12-13 09:14:39
## 5595063 37AC57E34B2E7E97  classic_bike 2021-12-13 08:45:32 2021-12-13 08:49:09
##                   start_station_name start_station_id         end_station_name
## 5595058       Canal St & Madison St            13341
## 5595059       Canal St & Madison St            13341
## 5595060       Canal St & Madison St            13341 Kingsbury St & Kinzie St
## 5595061       Canal St & Madison St            13341  Dearborn St & Monroe St
## 5595062      Lawndale Ave & 16th St            362.0
## 5595063 Michigan Ave & Jackson Blvd     TA1309000002  Dearborn St & Monroe St
##         end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 5595058                 41.88180 -87.63997 41.88000 -87.64000        casual
## 5595059                 41.88229 -87.63975 41.89000 -87.61000        casual
## 5595060   KA1503000043  41.88212 -87.64005 41.88911 -87.63886        member
## 5595061   TA1305000006  41.88196 -87.63995 41.88025 -87.62960        member
## 5595062                 41.86000 -87.72000 41.85000 -87.71000        member
## 5595063   TA1305000006  41.87785 -87.62408 41.88132 -87.62952        member
```

Check the summary of each column

```
summary(all_trip)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:5595063     Length:5595063     Length:5595063     Length:5595063
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##
##  start_station_name start_station_id   end_station_name   end_station_id
##  Length:5595063     Length:5595063     Length:5595063     Length:5595063
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    start_lat       start_lng         end_lat         end_lng
##  Min.   :41.64   Min.   :-87.84   Min.   :41.39   Min.   :-88.97
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.07   Max.   :-87.52   Max.   :42.17   Max.   :-87.49
##                                   NA's   :4771    NA's   :4771
##  member_casual
##  Length:5595063
##  Class :character
##  Mode  :character
##
##
##
##
```

Step 5: Clean the data and prepare for analysis.

Upon inspecting, we identify some workarounds:

a. Verify if categorical values in rideable_type, and member_casual were consistent.
b. Convert started_at and ended_at columns into datetime columns.
c. Separate datetime columns in to two column.

Verify if categorical values in rideable_type were consistent.

```
table(all_trip$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##       3251028        312343       2031692
```

Verify if categorical values in member_casual were consistent.

```
table(all_trip$member_casual)
```

```
##
##  casual  member
## 2529005 3066058
```

Note: If not consistent, standardize them.

Convert started_at and ended_at columns into datetime columns.

```
all_trip <- all_trip %>%
  mutate(
    started_at = ymd_hms(started_at),
```

```
    ended_at = ymd_hms(ended_at)
    )
```

create new columns for time of the day, day, month, year, day of week and hour.

```
all_trip <- all_trip %>%
  mutate(
    time = format(as.POSIXct(started_at), format = "%H:%M:%S"),
    day = format(started_at, "%d"),
    month = format(format(started_at, "%m")),
    year = format(started_at, "%Y"),
    day_of_week = format(started_at, "%A"),
    hour = hour(started_at)
    )
```

calculate the duration of the trip

```
all_trip$ride_length <- difftime(all_trip$ended_at, all_trip $started_at, units = "secs")
```

Convert ride_length from 'difftime num' to numeric so that we can run calculations on the data.

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

Verify if the column was successfully converted.

```
is.numeric(all_trip$ride_length)
```

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

Remove "bad" data. The dataframe includes a few hundred entries when bikes were taken out of docks and checked for quality by Divvy or ride_length was negative.

```
all_trip_v2 <- all_trip %>%
  distinct(ride_id, .keep_all = TRUE) %>%
  filter(ride_length > 60)
```

Analyze

Descriptive analysis on ride_length (all figures in seconds).

Average total ride length in seconds.

```
mean(all_trip_v2$ride_length)
```

```
## [1] 1336.336
```

Midpoint number of ride length.

```
median(all_trip_v2$ride_length)
```

```
## [1] 732
```

Longest ride.

```
max(all_trip_v2$ride_length)
```

```
## [1] 3356649
```

Shortest ride.

```
min(all_trip_v2$ride_length)
```

```
## [1] 61
```

Compare members and casual users.

Average total ride length.

```
aggregate(all_trip_v2$ride_length ~ all_trip_v2$member_casual, FUN = mean)
```

```
##   all_trip_v2$member_casual all_trip_v2$ride_length
## 1                    casual               1946.2553
## 2                    member                831.5963
```

Midpoint number of ride length.

```
aggregate(all_trip_v2$ride_length ~ all_trip_v2$member_casual, FUN = median)
```

```
##   all_trip_v2$member_casual all_trip_v2$ride_length
## 1                    casual                     973
## 2                    member                     586
```

Shortest ride.

```
aggregate(all_trip_v2$ride_length ~ all_trip_v2$member_casual, FUN = min)
```

```
##   all_trip_v2$member_casual all_trip_v2$ride_length
## 1                    casual                      61
## 2                    member                      61
```

Longest ride.

```
aggregate(all_trip_v2$ride_length ~ all_trip_v2$member_casual, FUN = max)
```

```
##   all_trip_v2$member_casual all_trip_v2$ride_length
## 1                    casual                3356649
## 2                    member                  93596
```

See the average ride time by each day for members vs casual users

```
aggregate(all_trip_v2$ride_length ~ all_trip_v2$member_casual + all_trip_v2$day_of_week, FUN = mean)
```

```
##    all_trip_v2$member_casual all_trip_v2$day_of_week all_trip_v2$ride_length
## 1                     casual                  Friday               1845.9868
## 2                     member                  Friday                812.7765
## 3                     casual                  Monday               1938.3704
## 4                     member                  Monday                807.8854
## 5                     casual                Saturday               2110.6803
## 6                     member                Saturday                932.5526
## 7                     casual                  Sunday               2285.1582
## 8                     member                  Sunday                956.7413
## 9                     casual                Thursday               1684.1645
## 10                    member                Thursday                778.5100
## 11                    casual                 Tuesday               1701.0276
## 12                    member                 Tuesday                779.3520
## 13                    casual               Wednesday               1681.9952
## 14                    member               Wednesday                781.2748
```

Arrange days of the week in chronological order.

```
all_trip_v2$day_of_week <- ordered(all_trip_v2$day_of_week, levels = c("Sunday", "Monday", "Tuesday",
                                                                       "Wednesday", "Thursday", "Friday
                                                                       "Saturday" ))
```

Rerun the average ride time by each day for members vs casual users

```
aggregate(all_trip_v2$ride_length ~ all_trip_v2$member_casual + all_trip_v2$day_of_week, FUN = mean)
```

```
##    all_trip_v2$member_casual all_trip_v2$day_of_week all_trip_v2$ride_length
## 1                     casual                  Sunday               2285.1582
## 2                     member                  Sunday                956.7413
## 3                     casual                  Monday               1938.3704
## 4                     member                  Monday                807.8854
## 5                     casual                 Tuesday               1701.0276
## 6                     member                 Tuesday                779.3520
## 7                     casual               Wednesday               1681.9952
## 8                     member               Wednesday                781.2748
## 9                     casual                Thursday               1684.1645
## 10                    member                Thursday                778.5100
## 11                    casual                  Friday               1845.9868
## 12                    member                  Friday                812.7765
## 13                    casual                Saturday               2110.6803
## 14                    member                Saturday                932.5526
```

Analyze ridership data by usertype and weekday.

```
all_trip_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)
```

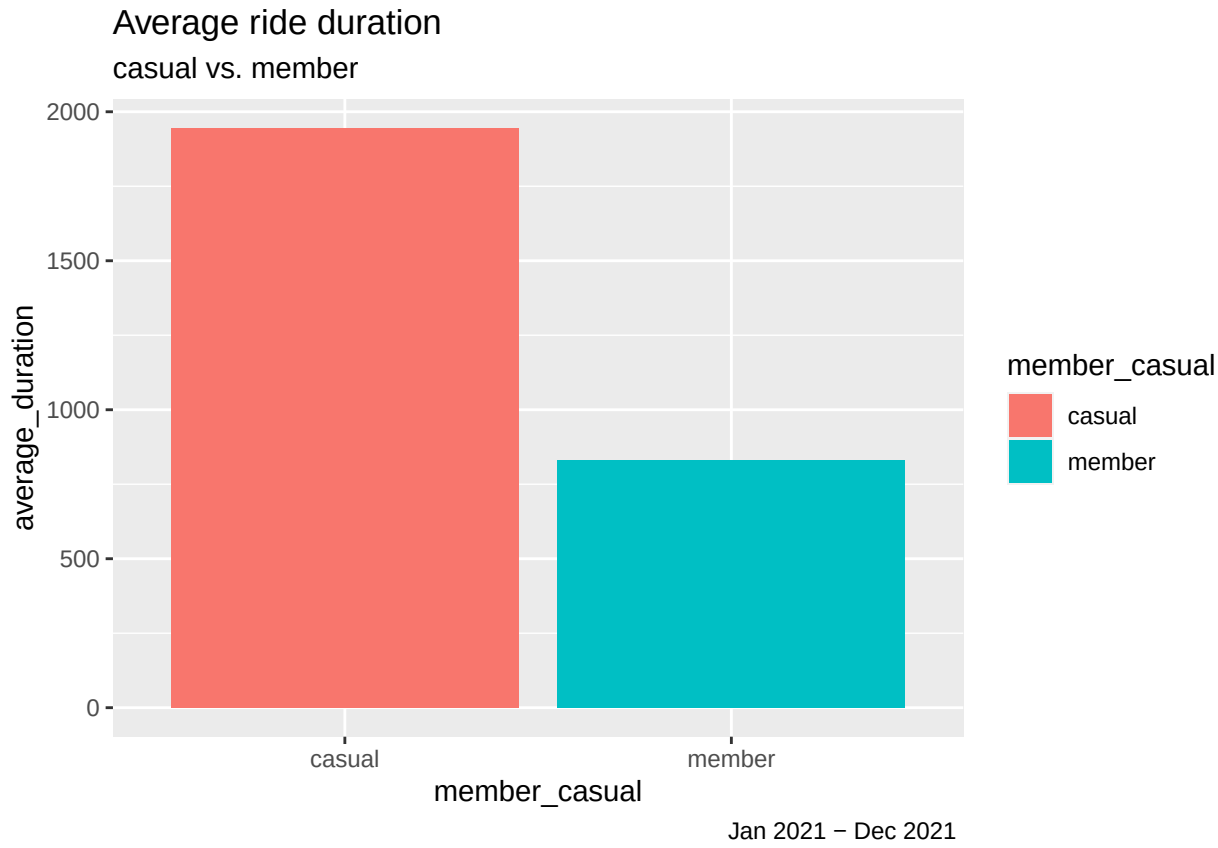
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              474469            2285.
##  2 casual        Mon              282505            1938.
##  3 casual        Tue              270675            1701.
##  4 casual        Wed              275152            1682.
##  5 casual        Thu              282279            1684.
##  6 casual        Fri              359064            1846.
##  7 casual        Sat              550420            2111.
##  8 member        Sun              369152             957.
##  9 member        Mon              409282             808.
## 10 member        Tue              458068             779.
## 11 member        Wed              469517             781.
## 12 member        Thu              444375             779.
## 13 member        Fri              438905             813.
## 14 member        Sat              425089             933.
```

Share

Create the Average ride duration per user.

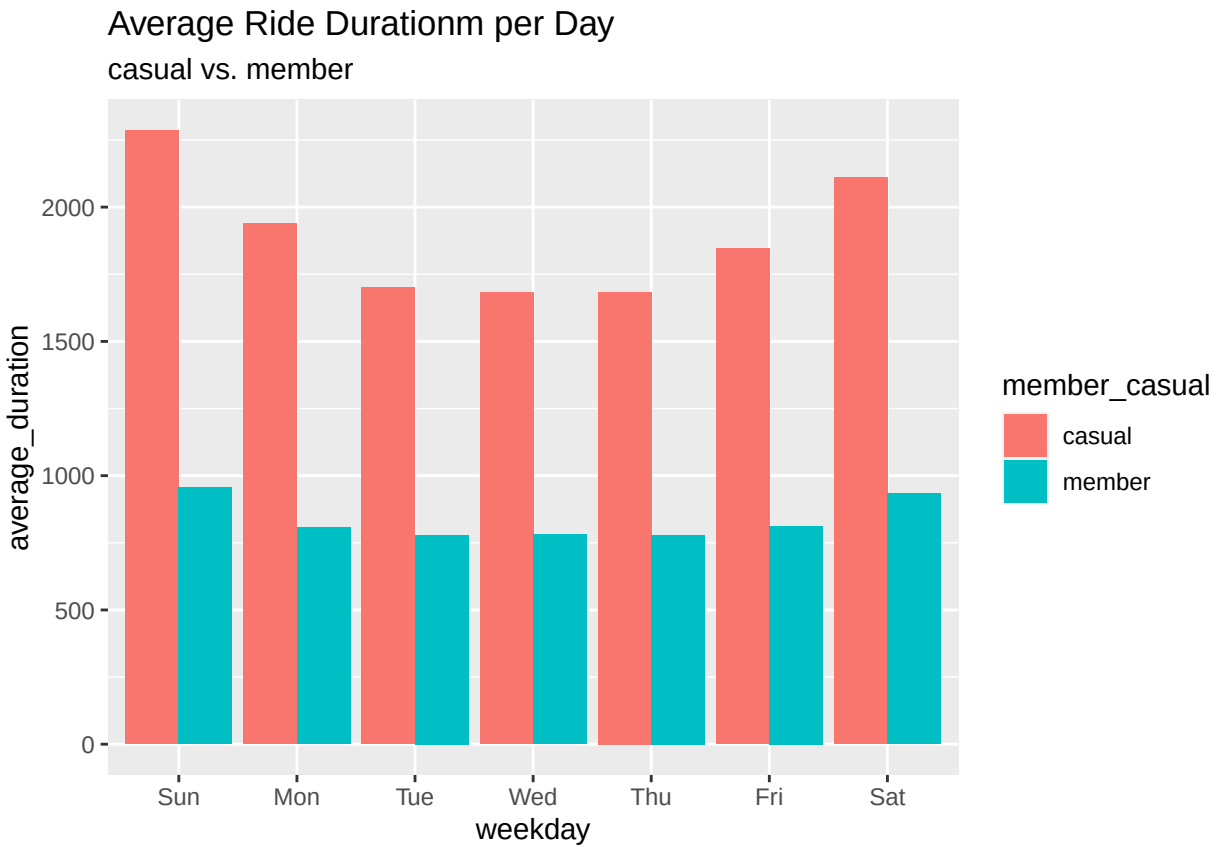
```
all_trip_v2 %>%
  group_by(member_casual) %>%
  summarise(average_duration = mean(ride_length)) %>%
  ggplot(aes(x = member_casual, y = average_duration, fill = member_casual)) +
    geom_col(position = "dodge") +
    labs(title = "Average ride duration", subtitle = "casual vs. member", caption = "Jan 2021 - Dec 202
```



Create a visualization for daily average duration

```
all_trip_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
    geom_col(position = "dodge") +
    labs(title = "Average Ride Durationm per Day", subtitle = "casual vs. member")
```

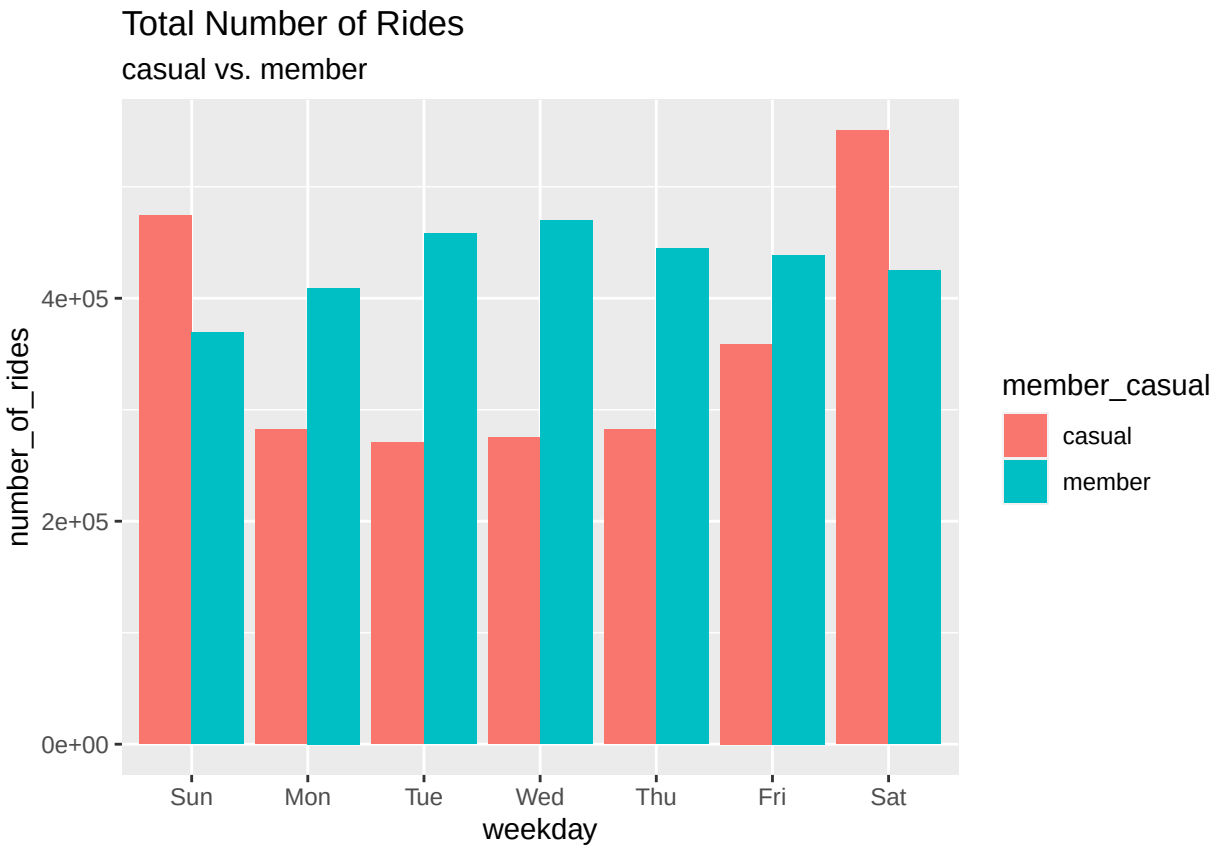
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```



Visualize the number of rides by rider type

```
all_trip_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Total Number of Rides", subtitle = "casual vs. member")
```

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



Key Fidings:

1. Casual rider's average ride duration is higher than members.
2. Casual rider's average ride duration increases during weekend.
3. No. of rides for casual riders surged during saturday while the members ride peak at the middle of the week.

Act

Conclusion

1. It is possible that casual riders are sightseeing persons based on their ride duration and days of trip.
2. The consistency of bike usage and ride durations throughout the week suggest that annual members used Cyclistic bikes for essential purposes.

Next steps

1. **Develop a semi-membership program.** Casual riders may enjoy the same benefit as annual members but are limited to every Friday, Saturday, and Sunday only.
2. **Analyze outliers for very long-duration trips**. Review and investigate if these trips are acceptable to the company policies.
3. **Explore the findings**. Gather more facts that casual riders are 'sightseeing persons'. An online survey may conduct to get the additional data needed.

Appendix

Data source: https://divvy-tripdata.s3.amazonaws.com/index.html
Licence: https://www.divvybikes.com/data-license-agreement