Bike Sharing Case Study

Tsun Hei Tai

In this case study, I have been tasked to work for a fictional company, Cyclistic, a bike sharing company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. One of the business tasks was to find out how annual members and casual riders use bikes differently.

The case study was completed as a part of the 'Google Data Analytics Certificate' online course on Coursera.

Business Task

Key question to answer:

How do annual members and casual riders use Cyclistic bikes differently?

Answering this question will help the company design marketing strategies aimed at converting casual riders into annual members.

Setup working environment in R

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

```
library(ggplot2)
```

Upload data ro R

Note: "Divvy_Trips_2020_Q2.csv" was made by taking data from April to June 2020 data and combining into 1 csv file. Data has been sourced from https://divvy-tripdata.s3.amazonaws.com/index.html. The data has been made available by Motivate International Inc. under this license. Data for this case study is appropriate and enables me to answer the business question.

```
q3_2019 <- read_csv("Divvy_Trips_2019_Q3.csv")
```

```
## Rows: 1640718 Columns: 12
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## dttm (2): start_time, end_time
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q4_2019 <- read_csv("Divvy_Trips_2019_Q4.csv")
```

```
## Rows: 704054 Columns: 12
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## dttm (2): start_time, end_time
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q1_2020 <- read_csv("Divvy_Trips_2020_Q1.csv")
```

```
## Rows: 426887 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q2_2020 <- read_csv("Divvy_Trips_2020_Q2.csv")
```

```
## Rows: 526940 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, en...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Wrangle data and combine data sets into one single dataset

Compare column names of each file

```
colnames(q3_2019)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```
colnames(q4_2019)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```
colnames(q1_2020)
```

```
##  [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"
```

```
colnames(q2_2020)
```

```
##  [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"
```

Rename column names so each file has consistent column names as q1_2020

```
q4_2019 <- rename(q4_2019
                  ,ride_id = trip_id
                  ,rideable_type = bikeid
                  ,started_at = start_time
                  ,ended_at = end_time
                  ,start_station_name = from_station_name
                  ,start_station_id = from_station_id
                  ,end_station_name = to_station_name
                  ,end_station_id = to_station_id
                  ,member_casual = usertype)


q3_2019 <- rename(q3_2019
                   ,ride_id = trip_id
                   ,rideable_type = bikeid
                   ,started_at = start_time
                   ,ended_at = end_time
                   ,start_station_name = from_station_name
                   ,start_station_id = from_station_id
                   ,end_station_name = to_station_name
```

```
                   ,end_station_id = to_station_id
                   ,member_casual = usertype)
```

Convert ride_id and rideable_type to character so that they can stack correctly

```
q4_2019 <-  mutate(q4_2019, ride_id = as.character(ride_id)
                   ,rideable_type = as.character(rideable_type))
q3_2019 <-  mutate(q3_2019, ride_id = as.character(ride_id)
                   ,rideable_type = as.character(rideable_type))
q2_2020 <-  mutate(q2_2020, started_at = as.POSIXct(started_at)
                   ,ended_at = as.POSIXct(ended_at))
```

Combine each quarter's data frame together into one large data frame

```
all_trips <- bind_rows(q3_2019, q4_2019, q1_2020, q2_2020)
```

Remove lat, long, birthyear, and gender fields as this data was dropped beginning in 2020

```
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, birthyear, gender, "tripduration"))
```

Clean and prepare data for analysis

Change labels in the member_casual column as "Subscriber" and "Customer" was used in 2019. Change to "member" and "casual" to match the labels used in 2020 data

```
all_trips <-  all_trips %>%
  mutate(member_casual = recode(member_casual
                           ,"Subscriber" = "member"
                           ,"Customer" = "casual"))
```

Check to ensure that member_casual column has only two labels

```
table(all_trips$member_casual)
```

```
##
##  casual  member
##  866035 2432564
```

Add columns for date, month, year and day of each ride

```
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), "%A")
```

Add "ride_length" calculation column to all_trips (in seconds)

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

Convert "ride_length" from to numeric so calculations can be made

```
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
is.numeric(all_trips$ride_length) # Check that ride_length column is numeric
```

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

Remove 'bad' data

The dataframe includes a few hundred entries when bikes were taken out of docks (HQ QR) and checked for quality by Divvy or ride_length was negative

```
all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR" | all_trips$ride_length<0),]
```

Descriptive analysis

Summary stats of ride_length

```
summary(all_trips_v2$ride_length)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##         0       268       576     49779      1118 788918400
```

Compare members and casual users

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                161515.70
## 2                     member                 10180.05
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                      1080
## 2                     member                       511
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                725846400
## 2                     member                788918400
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                         0
## 2                     member                         0
```

Average ride time by each day for members vs casual users

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Friday               118584.893
## 2                      member                   Friday                10425.323
## 3                      casual                   Monday               157033.360
## 4                      member                   Monday                 6263.064
## 5                      casual                 Saturday               139024.143
## 6                      member                 Saturday                21134.866
## 7                      casual                   Sunday               158368.862
## 8                      member                   Sunday                14688.050
## 9                      casual                 Thursday               156249.582
## 10                     member                 Thursday                 8595.832
## 11                     casual                  Tuesday               247242.631
## 12                     member                  Tuesday                 8763.475
## 13                     casual                Wednesday               183604.382
## 14                     member                Wednesday                 7278.830
```

Days of the week are out of order. Run script to fix this.

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "
```

Average ride time by each day for members vs casual users

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Sunday               158368.862
## 2                      member                   Sunday                14688.050
## 3                      casual                   Monday               157033.360
## 4                      member                   Monday                 6263.064
## 5                      casual                  Tuesday               247242.631
## 6                      member                  Tuesday                 8763.475
## 7                      casual                Wednesday               183604.382
## 8                      member                Wednesday                 7278.830
## 9                      casual                 Thursday               156249.582
## 10                     member                 Thursday                 8595.832
## 11                     casual                   Friday               118584.893
## 12                     member                   Friday                10425.323
## 13                     casual                 Saturday               139024.143
## 14                     member                 Saturday                21134.866
```

Analyse ridership data by type and weekday

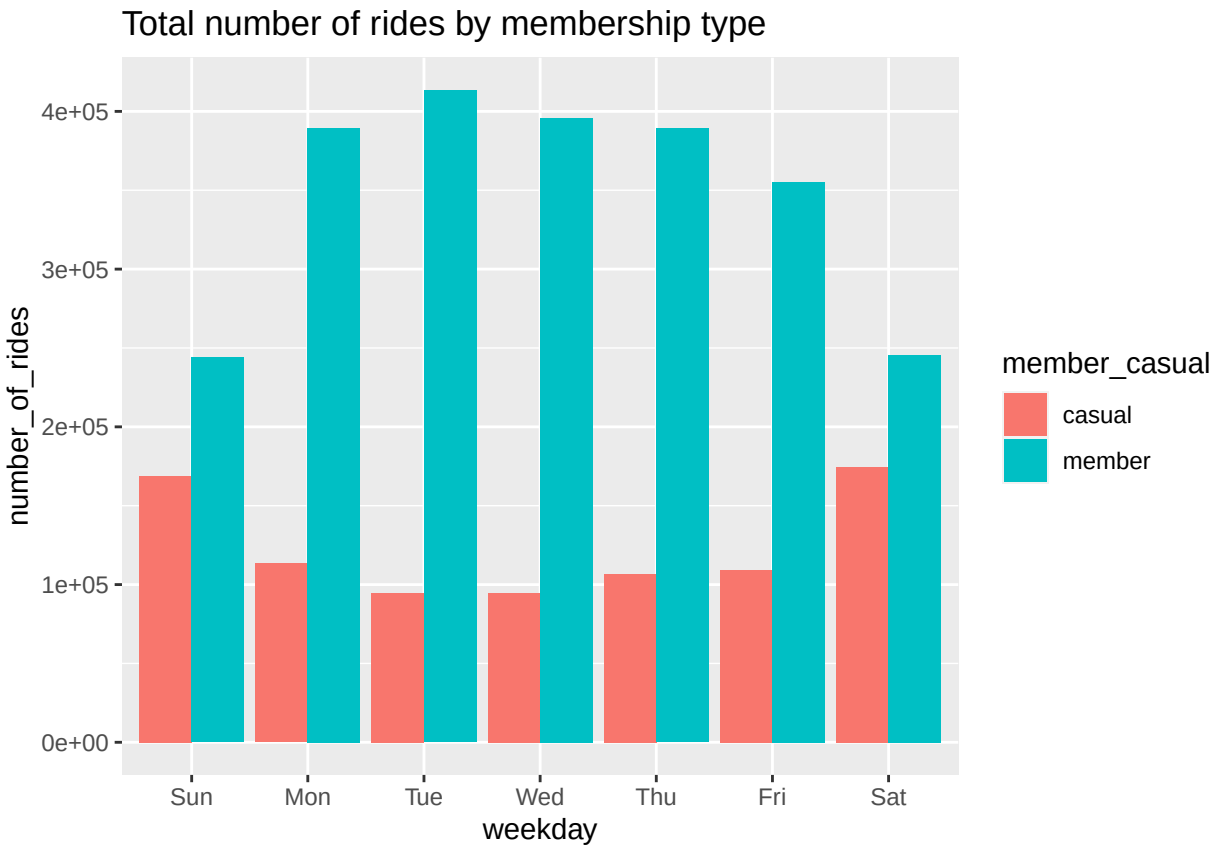
```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                              #calculates the number of rides and average
  ,average_duration = mean(ride_length)) %>%        # calculates the average duration
  arrange(member_casual, weekday)                                  # sorts
```

```
## # A tibble: 14 x 4
```

```
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              168656          158369.
##  2 casual        Mon              113467          157033.
##  3 casual        Tue               94832          247243.
##  4 casual        Wed               94533          183604.
##  5 casual        Thu              106748          156250.
##  6 casual        Fri              109210          118585.
##  7 casual        Sat              174635          139024.
##  8 member        Sun              243926           14688.
##  9 member        Mon              389350            6263.
## 10 member        Tue              413336            8763.
## 11 member        Wed              395933            7279.
## 12 member        Thu              389218            8596.
## 13 member        Fri              355293           10425.
## 14 member        Sat              245475           21135.
```

Visualise number of rides by rider type

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                              #calculates the number of rides and average
            ,average_duration = mean(ride_length)) %>%      # calculates the average duration
  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 by membership type")
```

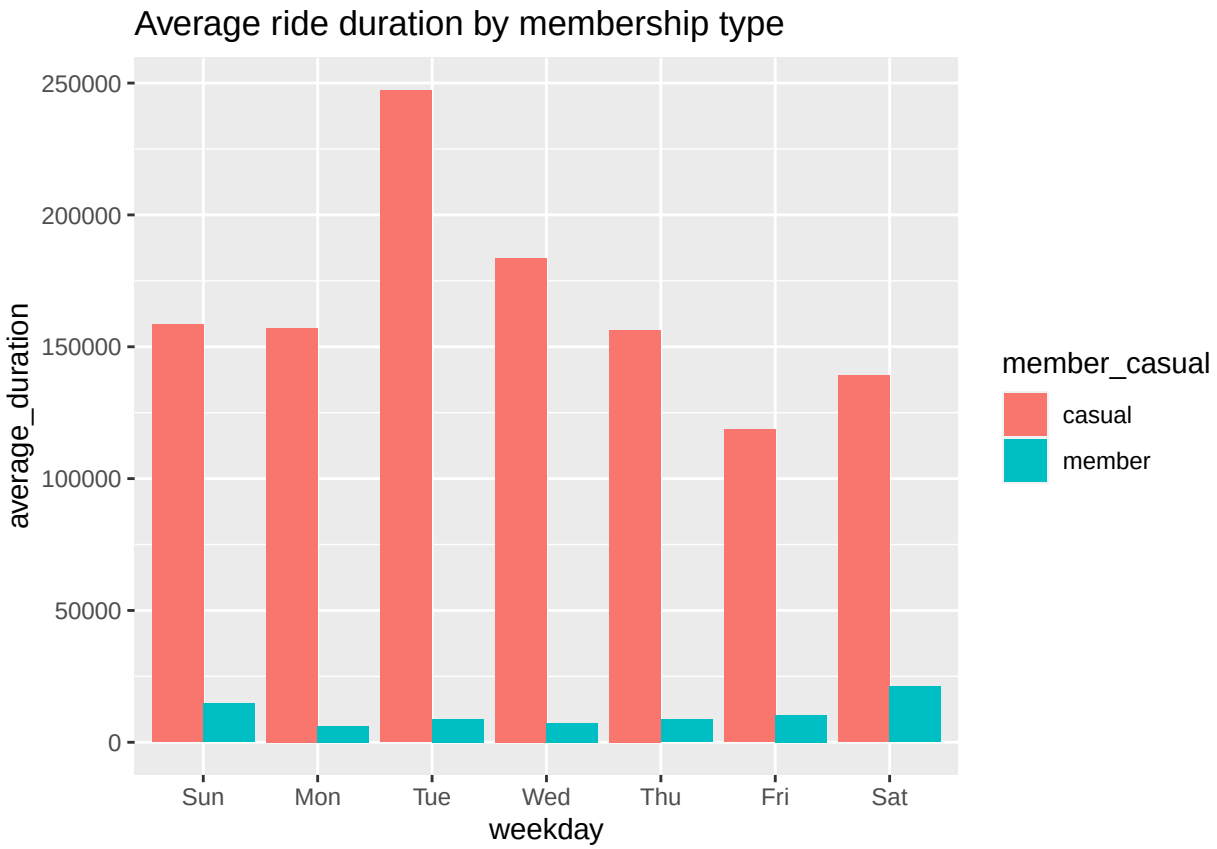


As seen in this graph above, the total number of rides on any weekday for annual members is more than total number of rides for casual riders. The most popular days for casual riders are on Saturdays and Sundays. Whereas for annual members, the most popular days are Monday to Friday.

Average ride duration by membership type

```
all_trips_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 duration by membership type")
```

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



As seen in the above graph, the average ride duration for casual members is far greater than the average ride duration for annual members. The greatest average ride duration for casual members being on a Tuesday.

Top recommendations from these findings

1. Most casual riders ride on the weekends, whereas annual members ride most during the weekdays. Recommend a signing on discount for casual riders to become annual members.
2. Casual riders have a far greater average ride duration than annual members. Compare the cost of casual rider trips vs the benefits/cost of becoming an annual member.
3. Further analysis with additional data could help understand why the average ride duration for casual riders is the most on a Tuesday. This may help with converting more casual riders into becoming annual members.