Cyclistic Case Study

Cecilia Rocco Grandal

2023/03/10

Business Task

The objective of this project is to develop marketing strategies that aim to convert casual riders into annual members based on the last three months of customer behaviour data. The project seeks to answer the following questions:

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

Data sources

Cyclistic's trip data

Analysis

Step 1

Collect the data:

```
library(tidyverse)
library(lubridate)
library(ggplot2)
library(readr)
library(dplyr)
```

```
dec_2022 <- read_csv("202212_trip_data.csv")
```

```
## Rows: 181806 Columns: 13
## -- Column specification ------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, s...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
jan_2023 <- read_csv("202301_trip_data.csv")
```

```
## Rows: 190301 Columns: 13
## -- Column specification ------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, s...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
feb_2023 <- read_csv("202302_trip_data.csv")
```

```
## Rows: 190445 Columns: 13
## -- Column specification ----------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, s...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

#Step 2

Visualize data and evaluate it:

```
str(dec_2022)
```

```
## spc_tbl_ [181,806 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:181806] "65DBD2F447EC51C2" "0C201AA7EA0EA1AD" "E0B148CCB358A49D" "54C57
##  $ rideable_type     : chr [1:181806] "electric_bike" "classic_bike" "electric_bike" "classic_bike"
##  $ started_at        : POSIXct[1:181806], format: "2022-12-05 10:47:18" ...
##  $ ended_at          : POSIXct[1:181806], format: "2022-12-05 10:56:34" ...
##  $ start_station_name: chr [1:181806] "Clifton Ave & Armitage Ave" "Broadway & Belmont Ave" "Sangamon
##  $ start_station_id  : chr [1:181806] "TA1307000163" "13277" "TA1306000015" "KA1503000038" ...
##  $ end_station_name  : chr [1:181806] "Sedgwick St & Webster Ave" "Sedgwick St & Webster Ave" "St. Cl
##  $ end_station_id    : chr [1:181806] "13191" "13191" "13016" "13134" ...
##  $ start_lat         : num [1:181806] 41.9 41.9 41.9 41.8 41.9 ...
##  $ start_lng         : num [1:181806] -87.7 -87.6 -87.7 -87.6 -87.7 ...
##  $ end_lat           : num [1:181806] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:181806] -87.6 -87.6 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr [1:181806] "member" "casual" "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>
```

```
str(jan_2023)
```

```
## spc_tbl_ [190,301 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:190301] "F96D5A74A3E41399" "13CB7EB698CEDB88" "BD88A2E670661CE5" "C907
##  $ rideable_type     : chr [1:190301] "electric_bike" "classic_bike" "electric_bike" "classic_bike"
##  $ started_at        : POSIXct[1:190301], format: "2023-01-21 20:05:42" ...
##  $ ended_at          : POSIXct[1:190301], format: "2023-01-21 20:16:33" ...
##  $ start_station_name: chr [1:190301] "Lincoln Ave & Fullerton Ave" "Kimbark Ave & 53rd St" "Western
##  $ start_station_id  : chr [1:190301] "TA1309000058" "TA1309000037" "RP-005" "TA1309000037" ...
##  $ end_station_name  : chr [1:190301] "Hampden Ct & Diversey Ave" "Greenwood Ave & 47th St" "Valli P
##  $ end_station_id    : chr [1:190301] "202480.0" "TA1308000002" "599" "TA1308000002" ...
##  $ start_lat         : num [1:190301] 41.9 41.8 42 41.8 41.8 ...
##  $ start_lng         : num [1:190301] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ end_lat           : num [1:190301] 41.9 41.8 42 41.8 41.8 ...
##  $ end_lng           : num [1:190301] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:190301] "member" "member" "casual" "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>
```

```
str(feb_2023)
```

```
## spc_tbl_ [190,445 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:190445] "CBCD0D7777F0E45F" "F3EC5FCE5FF39DE9" "E54C1F27FA9354FF" "3D56
##  $ rideable_type     : chr [1:190445] "classic_bike" "electric_bike" "classic_bike" "electric_bike"
##  $ started_at        : POSIXct[1:190445], format: "2023-02-14 11:59:42" ...
##  $ ended_at          : POSIXct[1:190445], format: "2023-02-14 12:13:38" ...
##  $ start_station_name: chr [1:190445] "Southport Ave & Clybourn Ave" "Clarendon Ave & Gordon Ter" "S
##  $ start_station_id  : chr [1:190445] "TA1309000030" "13379" "TA1309000030" "TA1309000030" ...
##  $ end_station_name  : chr [1:190445] "Clark St & Schiller St" "Sheridan Rd & Lawrence Ave" "Aberdee
##  $ end_station_id    : chr [1:190445] "TA1309000024" "TA1309000041" "13156" "TA1309000008" ...
##  $ start_lat         : num [1:190445] 41.9 42 41.9 41.9 41.8 ...
##  $ start_lng         : num [1:190445] -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:190445] 41.9 42 41.9 41.9 41.8 ...
##  $ end_lng           : num [1:190445] -87.6 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:190445] "casual" "casual" "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>
```

Afterwards, merge it:

```
last_quart <- bind_rows(dec_2022, jan_2023, feb_2023)
```

#Step 3

To begin the analysis process, the data needs to be cleaned and sorted. There are inconsistencies in the data, such as:

1. Different customer types being named in the member_casual column. To fix this, values are adjusted to range between only two instead of four:

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

```
##
## casual member
## 127918 434634
```

2. Date and time values are separated and added to newly created columns:

```
last_quart$date <- as.Date(last_quart$started_at) #The default format is yyyy-mm-dd
last_quart$month <- format(as.Date(last_quart$date), "%m")
last_quart$day <- format(as.Date(last_quart$date), "%d")
last_quart$year <- format(as.Date(last_quart$date), "%Y")
last_quart$day_of_week <- format(as.Date(last_quart$date), "%A")
```

3. A ride_length column is added by subtracting the starting time from the ending time:

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

Values in the ride_length column are converted from factor to numeric type for calculation:

```
is.factor(last_quart$ride_length)
```

```
## [1] FALSE
```

```
last_quart$ride_length <- as.numeric(as.character(last_quart$ride_length))
is.numeric(last_quart$ride_length)
```

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

4. Some data is removed because it represents events when bikes were out of service.

```
last_quart_v2 <- last_quart[!(last_quart$start_station_name == "HQ QR" | last_quart$ride_length<0),]
```

Step 4

Starting from now, the analysis step will be carried out.

The analysis starts by answering the first business question of how annual members and casual riders use Cyclistic bikes differently. The analysis includes average ride length, median ride length, maximum and minimum ride length:

```
aggregate(last_quart_v2$ride_length ~ last_quart_v2$member_casual, FUN = mean)
```

```
##   last_quart_v2$member_casual last_quart_v2$ride_length
## 1                      casual                 1527.2820
## 2                      member                  630.4394
```

```
aggregate(last_quart_v2$ride_length ~ last_quart_v2$member_casual, FUN = median)
```

```
##   last_quart_v2$member_casual last_quart_v2$ride_length
## 1                      casual                       511
## 2                      member                       429
```

```
aggregate(last_quart_v2$ride_length ~ last_quart_v2$member_casual, FUN = max)
```

```
##   last_quart_v2$member_casual last_quart_v2$ride_length
## 1                      casual                   2016224
## 2                      member                     89996
```

```
aggregate(last_quart_v2$ride_length ~ last_quart_v2$member_casual, FUN = min)
```

```
##   last_quart_v2$member_casual last_quart_v2$ride_length
## 1                      casual                         0
## 2                      member                         0
```

Additionally, the analysis includes comparing average ride time based on the day of the week for each user type.

```
aggregate(last_quart_v2$ride_length ~ last_quart_v2$member_casual + last_quart_v2$day_of_week, FUN = mea
```

```
##    last_quart_v2$member_casual last_quart_v2$day_of_week
## 1                       casual                    Friday
## 2                       member                    Friday
## 3                       casual                    Monday
## 4                       member                    Monday
## 5                       casual                  Saturday
## 6                       member                  Saturday
## 7                       casual                    Sunday
## 8                       member                    Sunday
## 9                       casual                  Thursday
## 10                      member                  Thursday
## 11                      casual                   Tuesday
## 12                      member                   Tuesday
## 13                      casual                 Wednesday
## 14                      member                 Wednesday
##    last_quart_v2$ride_length
## 1                  1615.1775
## 2                   628.6931
## 3                  1355.2468
## 4                   626.2915
## 5                  1896.2738
## 6                   680.8529
## 7                   1873.4363
## 8                    708.4623
## 9                   1300.2877
## 10                   602.0274
## 11                  1197.5667
## 12                   602.9122
## 13                  1335.7962
## 14                   606.5257
```

The days of the week are ordered:

```
last_quart_v2$day_of_week <- ordered(last_quart_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday",
aggregate(last_quart_v2$ride_length ~ last_quart_v2$member_casual + last_quart_v2$day_of_week, FUN = mea
```

```
##    last_quart_v2$member_casual last_quart_v2$day_of_week
## 1                       casual                    Sunday
## 2                       member                    Sunday
## 3                       casual                    Monday
## 4                       member                    Monday
## 5                       casual                   Tuesday
## 6                       member                   Tuesday
## 7                       casual                 Wednesday
## 8                       member                 Wednesday
## 9                       casual                  Thursday
## 10                      member                  Thursday
## 11                      casual                    Friday
## 12                      member                    Friday
## 13                      casual                  Saturday
## 14                      member                  Saturday
##    last_quart_v2$ride_length
## 1                  1873.4363
## 2                   708.4623
## 3                  1355.2468
## 4                   626.2915
## 5                  1197.5667
## 6                   602.9122
## 7                  1335.7962
## 8                   606.5257
## 9                  1300.2877
## 10                  602.0274
## 11                 1615.1775
## 12                  628.6931
## 13                 1896.2738
## 14                  680.8529
```

Then, ridership behaviour per day and user type are compared:

```
library(lubridate)
last_quart_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday() (which come
  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: 15 x 4
## # Groups:   member_casual [3]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun               18445            1873.
##  2 casual        Mon               14520            1355.
##  3 casual        Tue               16311            1198.
##  4 casual        Wed               13493            1336.
##  5 casual        Thu               14007            1300.
##  6 casual        Fri               13488            1615.
##  7 casual        Sat               16447            1896.
##  8 member        Sun               41105             708.
##  9 member        Mon               57021             626.
## 10 member        Tue               69944             603.
## 11 member        Wed               57471             607.
## 12 member        Thu               58645             602.
## 13 member        Fri               48905             629.
## 14 member        Sat               41273             681.
## 15 <NA>          <NA>              81476              NA
```

The analysis shows that, on average, casual members have longer rides every day of the week compared to annual members. However, the number of rides is higher for annual members.

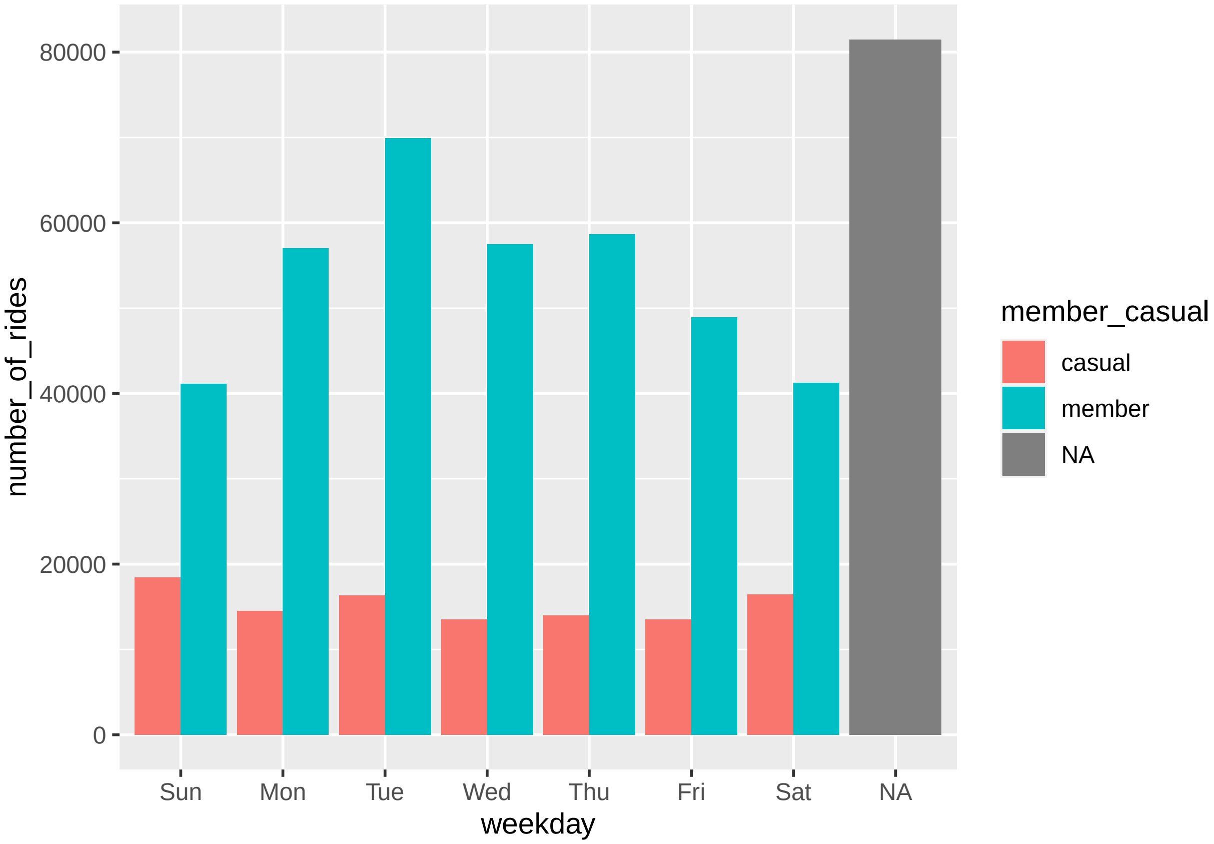
Step 5

To visualize the data, two plots will be created to have a different view of the data to evaluate it.

The first plot describes the number of rides per day, per user type.

```
library(ggplot2)
last_quart_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")
```

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

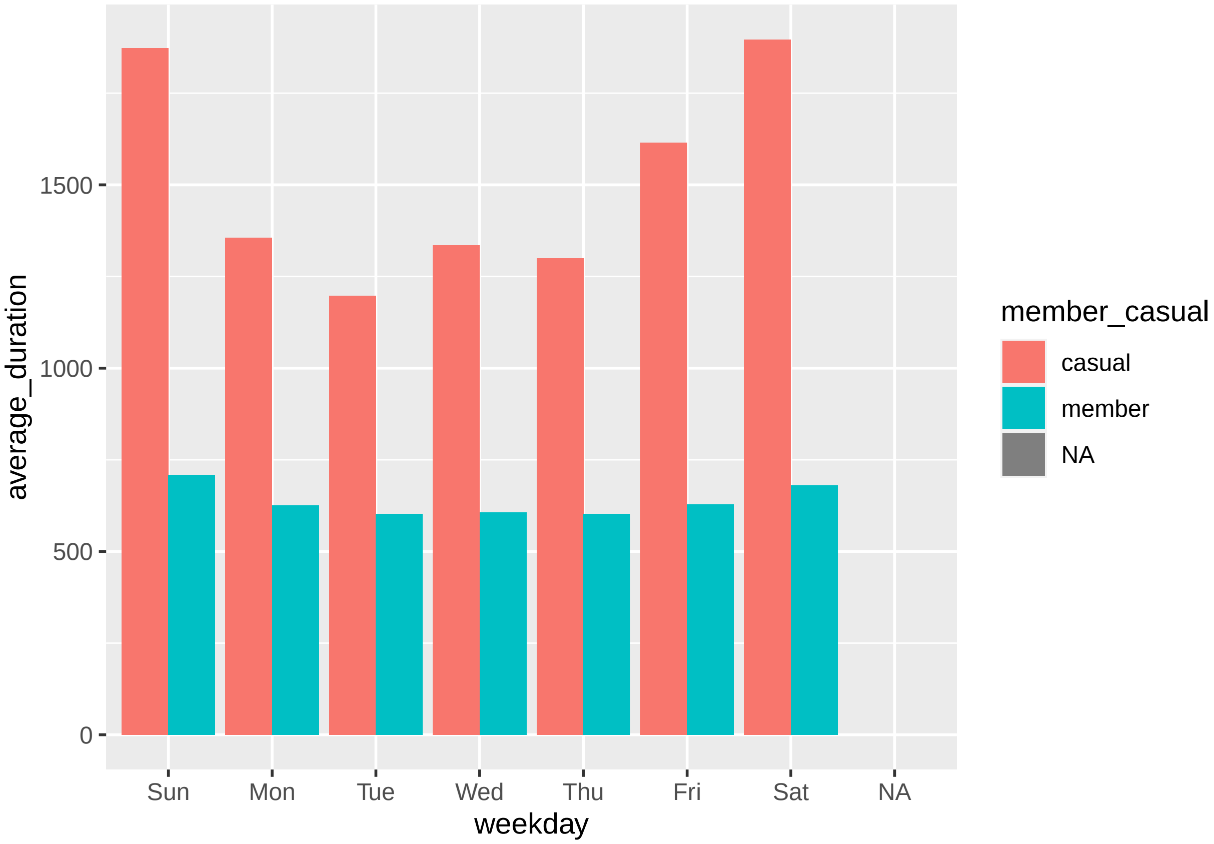


The second plot describes the average duration of the rides per day, per user type.

```
last_quart_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")
```

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

## Warning: Removed 1 rows containing missing values
## (`geom_col()`).
```



#Conclusions After analyzing, comparing, operating on, and visualizing the data, two questions remain unanswered: why would casual riders buy a membership, and how can Cyclistic use digital media to influence casual riders to become members?

To answer these questions, it is important to consider that casual users perform longer rides on average. One suggestion is to offer discounts or incentives to these users to encourage them to purchase an annual membership. Using digital media to target these casual users and apply different marketing strategies could be an effective approach to encouraging membership purchases.