Divvy bikes Case Study

FJ

2023-08-31

```
knitr::opts_knit$set(root.dir = "C:/Users/Felip/OneDrive/Escritorio/Freelancer/Portfolio/7. Case study
```

Cyclistic_Exercise_Full_Year_Analysis

This case study is about a bike-share company in Chicago called ***Cyclistic***. The director of Marketing believes that the company's futures success depends on maximizing the number of annual memberships. So the team want to understand how casual riders and annual members use **Cyclistic** bikes differently. From these insights, the marketing department would design a new strategy to convert casual riders into annual members.

Context

Cyclistic offer the next options to their customers:

- Single-ride passes
- Full-day passes
- Annual memberships

Customer who purchase single-ride or full-day passes are referred as *casual riders*. Customers who purchase annual membership are *Cyclsitic members.*The finance department conclude that annual members are much more profitable than casual riders. So the marketing's director believes that is easier to convert casual members into members, because casual riders are already aware of the company program and have chosen Cyclistic for their mobility needs.

Analysis

This analysis look for answer the next questions:

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?

This information would help to marketing department planning the best strategy to

STEP 1: INSTALL REQUIRED PACKAGES & DATA

For this analysis, we'll be using data from 2019 entire year, that previously was cleaned and combined using SQL Server. You can check this process here SQL__ script

```
library(tidyverse)  #helps wrangle data
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(lubridate)  #helps wrangle date attributes
library(ggplot2)  #helps visualize data
library(dplyr)
```

STEP 2: LOADING DATA

To start with the analysis, we have to set our working directory and read the CSV file with the data

```
setwd("C:/Users/Felip/OneDrive/Escritorio/Freelancer/Portfolio/7. Case study Bike - Sahere navigate spe
all_trips <- read_csv("C:/Users/Felip/OneDrive/Escritorio/Freelancer/Portfolio/7. Case study Bike - Sah
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 3816317 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, user_type, gender
## dbl  (7): trip_id, day_of_week, bike_id, trip_duration, from_station_id, to_...
## dttm (2): start_time, end_time
## time (1): ride_length
##
## 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 3: PREPARE DATA FOR ANALYSIS

3.1 Inspecting the dataset: Column names

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

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "ride_length"       "day_of_week"       "bike_id"
##  [7] "trip_duration"     "from_station_id"   "from_station_name"
## [10] "to_station_id"     "to_station_name"   "user_type"
## [13] "gender"            "birth_year"
```

We have a total of 19 columns in our data set.

Number of rows

```r
#How many rows are in data frame?
nrow(all_trips)
```

```
## [1] 3816317
```

The data set have 3816317 corresponding to company trips during 2019.

First 6 rows

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

```
## # A tibble: 6 x 14
##    trip_id start_time          end_time            ride_length day_of_week
##      <dbl> <dttm>              <dttm>              <time>            <dbl>
## 1 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13 16'47"                2
## 2 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56 04'17"                2
## 3 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41 09'08"                2
## 4 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11 06'23"                2
## 5 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44 35'37"                2
## 6 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39 35'20"                2
## # i 9 more variables: bike_id <dbl>, trip_duration <dbl>,
## #   from_station_id <dbl>, from_station_name <chr>, to_station_id <dbl>,
## #   to_station_name <chr>, user_type <chr>, gender <chr>, birth_year <dbl>
```

Columns and data types

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

```
## spc_tbl_ [3,816,317 x 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ trip_id          : num [1:3816317] 22178533 22178534 22178535 22178536 22178537 ...
##  $ start_time       : POSIXct[1:3816317], format: "2019-04-01 00:19:26" "2019-04-01 00:19:39" ...
##  $ end_time         : POSIXct[1:3816317], format: "2019-04-01 00:36:13" "2019-04-01 00:23:56" ...
##  $ ride_length      : 'hms' num [1:3816317] 00:16:47 00:04:17 00:09:08 00:06:23 ...
##   ..- attr(*, "units")= chr "secs"
##  $ day_of_week      : num [1:3816317] 2 2 2 2 2 2 2 2 2 2 ...
##  $ bike_id          : num [1:3816317] 3270 3123 6418 4513 3280 ...
##  $ trip_duration    : num [1:3816317] 1007 257 548 383 2137 ...
##  $ from_station_id  : num [1:3816317] 202 420 503 260 211 211 304 37 75 334 ...
##  $ from_station_name: chr [1:3816317] "Halsted St & 18th St" "Ellis Ave & 55th St" "Drake Ave & Fulle
##  $ to_station_id    : num [1:3816317] 129 426 500 499 211 211 232 337 36 256 ...
##  $ to_station_name  : chr [1:3816317] "Blue Island Ave & 18th St" "Ellis Ave & 60th St" "Central Par
##  $ user_type        : chr [1:3816317] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender           : chr [1:3816317] "Male" "Male" "Male" "Male" ...
##  $ birth_year       : num [1:3816317] 1992 1999 1969 1991 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   trip_id = col_double(),
##   ..   start_time = col_datetime(format = ""),
##   ..   end_time = col_datetime(format = ""),
##   ..   ride_length = col_time(format = ""),
##   ..   day_of_week = col_double(),
##   ..   bike_id = col_double(),
##   ..   trip_duration = col_double(),
##   ..   from_station_id = col_double(),
##   ..   from_station_name = col_character(),
##   ..   to_station_id = col_double(),
##   ..   to_station_name = col_character(),
##   ..   user_type = col_character(),
##   ..   gender = col_character(),
##   ..   birth_year = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Statistic summary for each column

```
#Statistical summary of data. Mainly for numerics
summary(all_trips)
```

```
##     trip_id            start_time
##  Min.   :21742443   Min.   :2019-01-01 00:04:00.00
##  1st Qu.:22873739   1st Qu.:2019-05-29 15:46:24.00
##  Median :23962217   Median :2019-07-25 17:48:36.00
##  Mean   :23915575   Mean   :2019-07-19 21:43:05.52
##  3rd Qu.:24963661   3rd Qu.:2019-09-15 04:13:05.00
##  Max.   :25962904   Max.   :2019-12-31 23:57:17.00
##     end_time                      ride_length        day_of_week
##  Min.   :2019-01-01 00:11:00.00   Length:3816317    Min.   :1.000
##  1st Qu.:2019-05-29 16:05:57.00   Class1:hms        1st Qu.:2.000
##  Median :2019-07-25 18:07:38.00   Class2:difftime   Median :4.000
##  Mean   :2019-07-19 22:02:11.47   Mode  :numeric    Mean   :4.064
##  3rd Qu.:2019-09-15 06:12:53.00                     3rd Qu.:6.000
##  Max.   :2020-01-01 17:25:25.00                     Max.   :7.000
##     bike_id      trip_duration   from_station_id from_station_name
##  Min.   :   1   Min.   :   61   Min.   :  1.0   Length:3816317
##  1st Qu.:1727   1st Qu.:  411   1st Qu.: 77.0   Class :character
##  Median :3453   Median :  709   Median :174.0   Mode  :character
##  Mean   :3380   Mean   : 1146   Mean   :201.6
##  3rd Qu.:5046   3rd Qu.: 1282   3rd Qu.:289.0
##  Max.   :6946   Max.   :90996   Max.   :673.0
##  to_station_id   to_station_name     user_type            gender
##  Min.   :  1.0   Length:3816317     Length:3816317     Length:3816317
##  1st Qu.: 77.0   Class :character   Class :character   Class :character
##  Median :174.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :202.6
##  3rd Qu.:291.0
##  Max.   :673.0
##    birth_year
##  Min.   :   0
##  1st Qu.:1969
##  Median :1985
##  Mean   :1694
##  3rd Qu.:1991
##  Max.   :2014
```

here we can see some statistics for each cokumn such as: Min, max, mean and mode values for each column.

3.2 Adding columns: date, month, day and year We will want to add some additional columns of data such as day, month , year. That provide additional opportunities to further analysis.

```
all_trips$date <- as.Date(all_trips$start_time) #deafult format date is yyy-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$day_week <- format(as.Date(all_trips$date),"%A")
```

```


all_trips$ride_length <- difftime(all_trips$end_time,all_trips$start_time)
# Convert "ride_length" from Factor to numeric so we can run calculations on the data
is.factor(all_trips$ride_length)
```

3.3 Add a “ride_length” calculation to all_trips (in seconds)

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

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

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

3.4 Remove “bad” data The data frame includes a few hundred entries when bikes were taken out of docks and checked for quality by Divvy or ride_length was negative # We will create a new version of the data frame (v2) since data is being removed

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

```r
summary(all_trips_v2)
```

3.5 Inspecting new dataset

```
##     trip_id           start_time
##  Min.   :21742443   Min.   :2019-01-01 00:04:00.00
##  1st Qu.:22873736   1st Qu.:2019-05-29 15:46:09.75
##  Median :23962210   Median :2019-07-25 17:48:27.00
##  Mean   :23915569   Mean   :2019-07-19 21:42:34.27
##  3rd Qu.:24963651   3rd Qu.:2019-09-15 03:45:54.75
##  Max.   :25962904   Max.   :2019-12-31 23:57:17.00
##     end_time                       ride_length       day_of_week
##  Min.   :2019-01-01 00:11:00.00   Min.   :   1.00   Min.   :1.000
##  1st Qu.:2019-05-29 16:05:54.50   1st Qu.:   6.90   1st Qu.:2.000
##  Median :2019-07-25 18:07:13.00   Median :  11.85   Median :4.000
##  Mean   :2019-07-19 22:01:40.24   Mean   :  19.10   Mean   :4.064
##  3rd Qu.:2019-09-15 05:26:20.00   3rd Qu.:  21.37   3rd Qu.:6.000
##  Max.   :2020-01-01 17:25:25.00   Max.   :1516.62   Max.   :7.000
##     bike_id     trip_duration   from_station_id from_station_name
##  Min.   :   1   Min.   :   61   Min.   :  1.0   Length:3816304
##  1st Qu.:1727   1st Qu.:  411   1st Qu.: 77.0   Class :character
##  Median :3453   Median :  709   Median :174.0   Mode  :character
##  Mean   :3380   Mean   : 1146   Mean   :201.6
##  3rd Qu.:5046   3rd Qu.: 1282   3rd Qu.:289.0
##  Max.   :6946   Max.   :90996   Max.   :673.0
##  to_station_id   to_station_name     user_type           gender
##  Min.   :  1.0   Length:3816304     Length:3816304     Length:3816304
##  1st Qu.: 77.0   Class :character   Class :character   Class :character
##  Median :174.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :202.6
##  3rd Qu.:291.0
##  Max.   :673.0
##    birth_year        date               month               day
##  Min.   :   0   Min.   :2019-01-01   Length:3816304     Length:3816304
##  1st Qu.:1969   1st Qu.:2019-05-29   Class :character   Class :character
##  Median :1985   Median :2019-07-25   Mode  :character   Mode  :character
##  Mean   :1694   Mean   :2019-07-19
##  3rd Qu.:1991   3rd Qu.:2019-09-15
##  Max.   :2014   Max.   :2019-12-31
##      year              day_week
##  Length:3816304     Length:3816304
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

Now , the negative values fror ride_length column are deleted

```r
head(all_trips_v2)
```

```
## # A tibble: 6 x 19
##    trip_id start_time          end_time            ride_length day_of_week
##      <dbl> <dttm>              <dttm>                    <dbl>       <dbl>
## 1 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13       16.8            2
## 2 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56        4.28           2
## 3 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41        9.13           2
## 4 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11        6.38           2
## 5 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44       35.6            2
## 6 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39       35.3            2
## # i 14 more variables: bike_id <dbl>, trip_duration <dbl>,
## #   from_station_id <dbl>, from_station_name <chr>, to_station_id <dbl>,
## #   to_station_name <chr>, user_type <chr>, gender <chr>, birth_year <dbl>,
## #   date <date>, month <chr>, day <chr>, year <chr>, day_week <chr>
```

STEP 4: CONDUCT DESCRIPTIVE ANALYSIS

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

4.1 Descriptive analysis on ride_length

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    6.90   11.85   19.10   21.37 1516.62
```

4.2 Compare members and casual users Members appears in our data as *Subscriber* and casual users as *Customer*

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

```
##   all_trips_v2$user_type all_trips_v2$ride_length
## 1               Customer                 39.57790
## 2             Subscriber                 12.96758
```

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

```
##   all_trips_v2$user_type all_trips_v2$ride_length
## 1               Customer                    25.80
## 2             Subscriber                     9.85
```

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

```
##   all_trips_v2$user_type all_trips_v2$ride_length
## 1               Customer                 1516.617
## 2             Subscriber                 1512.750
```

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

```
##   all_trips_v2$user_type all_trips_v2$ride_length
## 1               Customer                        1
## 2             Subscriber                        1
```

- The average from casual members is higher than annual members. But let's see how many suscribers and how many casual members we have.

```
# Calculate the count of records for each user type
user_type_counts <- table(all_trips_v2$user_type)

# Print the counts
print(user_type_counts)
```

```
##
##   Customer Subscriber
##     879374    2936930
```

- We have **2936930** annual members and **879374** casual members.
- The amount of annual members is 3.3 higher than casual members, but the avg ride length from casual members is 3.05 higher than annual members. For any reason casual members have longer trips than annual members.

```


aggregate(all_trips_v2$ride_length ~ all_trips_v2$user_type + all_trips_v2$day_week, FUN = mean)
```

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

```
##    all_trips_v2$user_type all_trips_v2$day_week all_trips_v2$ride_length
## 1                Customer                Friday                 38.48429
## 2              Subscriber                Friday                 12.56572
## 3                Customer                Monday                 39.47310
## 4              Subscriber                Monday                 12.68410
## 5                Customer              Saturday                 41.68049
## 6              Subscriber              Saturday                 14.49981
## 7                Customer                Sunday                 41.51363
## 8              Subscriber                Sunday                 14.27914
## 9                Customer              Thursday                 37.46438
## 10             Subscriber              Thursday                 12.67308
## 11               Customer               Tuesday                 37.80467
## 12             Subscriber               Tuesday                 12.62597
## 13               Customer             Wednesday                 36.76430
## 14             Subscriber             Wednesday                 12.66589
```

The days of the week are out of order. let's fix that

```
all_trips_v2$day_week <- ordered(all_trips_v2$day_week, levels=c("Sunday", "Monday", "Tuesday", "Wednes
aggregate(all_trips_v2$ride_length ~ all_trips_v2$user_type + all_trips_v2$day_week, FUN = mean)
```

```
##    all_trips_v2$user_type all_trips_v2$day_week all_trips_v2$ride_length
## 1                Customer                Sunday                 41.51363
## 2              Subscriber                Sunday                 14.27914
## 3                Customer                Monday                 39.47310
## 4              Subscriber                Monday                 12.68410
## 5                Customer               Tuesday                 37.80467
## 6              Subscriber               Tuesday                 12.62597
## 7                Customer             Wednesday                 36.76430
## 8              Subscriber             Wednesday                 12.66589
## 9                Customer              Thursday                 37.46438
## 10             Subscriber              Thursday                 12.67308
## 11               Customer                Friday                 38.48429
## 12             Subscriber                Friday                 12.56572
## 13               Customer              Saturday                 41.68049
## 14             Subscriber              Saturday                 14.49981
```

- The avg of ride length for day also shows that the ride length of casual members is higher everyday of week.

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

4.4 Analyze ridership data by type and weekday

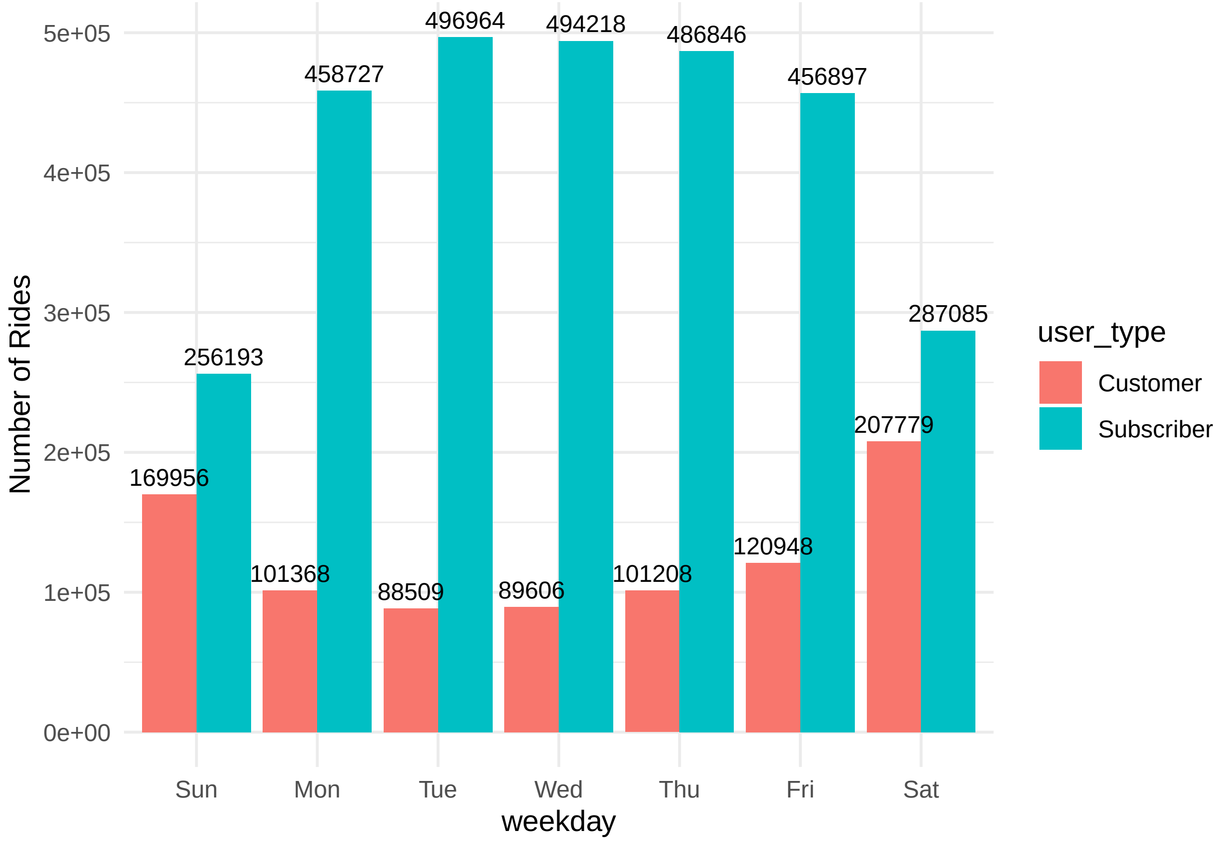
```
## `summarise()` has grouped output by 'user_type'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   user_type [2]
##    user_type  weekday number_of_rides average_duration
##    <chr>      <ord>             <int>            <dbl>
##  1 Customer   Sun              169956             41.5
##  2 Customer   Mon              101368             39.5
##  3 Customer   Tue               88509             37.8
##  4 Customer   Wed               89606             36.8
##  5 Customer   Thu              101208             37.5
##  6 Customer   Fri              120948             38.5
##  7 Customer   Sat              207779             41.7
##  8 Subscriber Sun              256193             14.3
##  9 Subscriber Mon              458727             12.7
## 10 Subscriber Tue              496964             12.6
## 11 Subscriber Wed              494218             12.7
## 12 Subscriber Thu              486846             12.7
## 13 Subscriber Fri              456897             12.6
## 14 Subscriber Sat              287085             14.5
```

```
all_trips_v2 %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(user_type, weekday) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(user_type, weekday) %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = user_type)) +
  geom_col(position = "dodge") +
  geom_text(aes(label = number_of_rides), position = position_dodge(width = 0.9), vjust = -0.5, size=3)
  labs(y = "Number of Rides") +
  theme_minimal()
```

4.5 Let’s visualize the number of rides by rider type

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

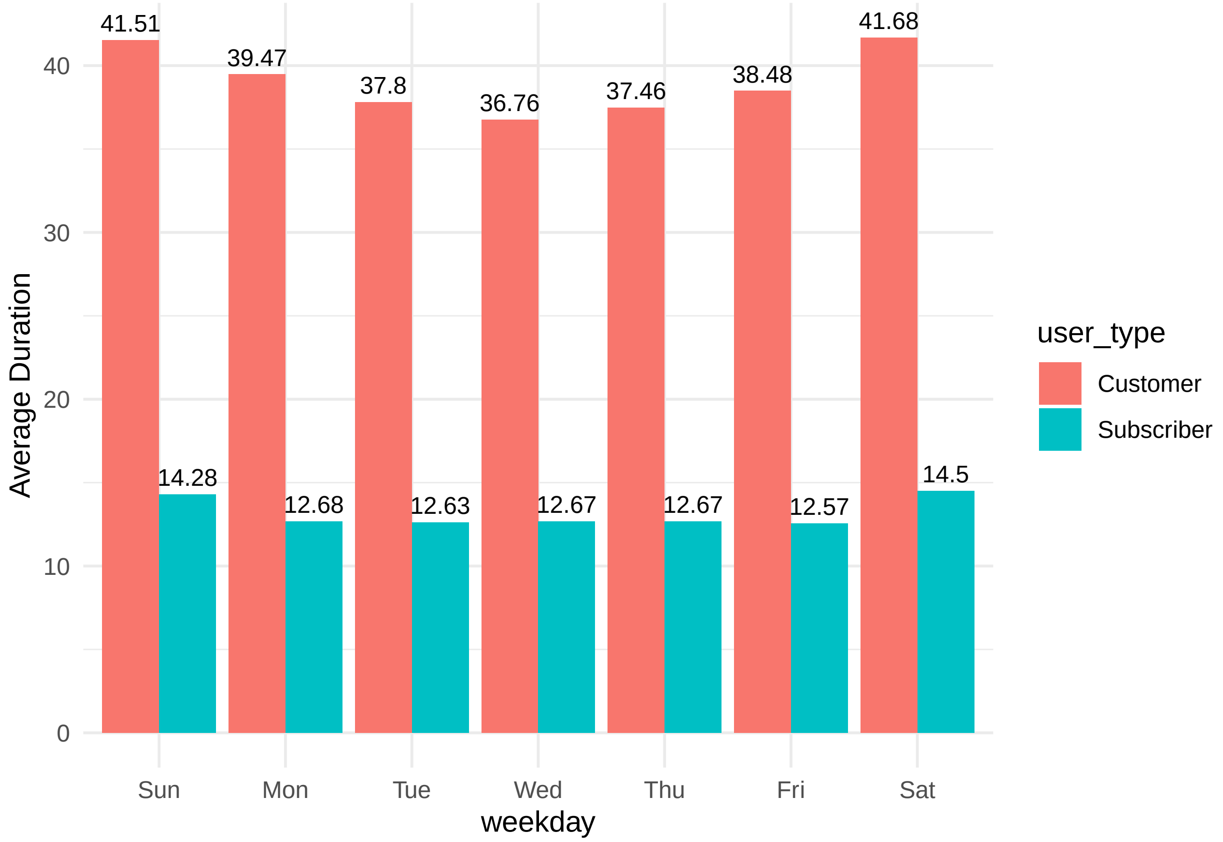


- The number of subscribers’ trip is much larger than those who members.
- The number of rides for **Subscribers** increase for MON-FRI days and is lower in weekends.
- Casual members have the higher number of rides in weekends and this number decrease during MON-FRI days.

```
all_trips_v2 %>%
  mutate(weekday = wday(start_time, label = TRUE)) %>%
  group_by(user_type, weekday) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(user_type, weekday) %>%
  ggplot(aes(x = weekday, y = average_duration, fill = user_type)) +
  geom_col(position = "dodge") +
  geom_text(aes(label = round(average_duration, 2)), position = position_dodge(width = 0.9), vjust = -0
  labs(y = "Average Duration") +
  theme_minimal()
```

Let's create a visualization for average duration

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



- Subscriber average trip duration remain nearly constant throughout the week, with a slight increase on weekends.
- Casual members average trip duration is higher in weekend days.
- The average trip duration of casual members is higher than subscribers.

Conclusions

1. Membership Disparity: Cyclistic has a significantly larger number of annual members (2,936,930) compared to casual riders (879,374). However, the average ride length for casual riders is substantially higher

(3.05 times) than that of annual members. This suggests that casual riders tend to take longer trips when they use Cyclistic bikes.

2. Weekly Ride Patterns: There are clear differences in the riding patterns between annual members and casual riders. Annual members tend to use the service more on weekdays, with a decrease in rides on weekends. Probably the annual members use as their primary mode of transportation to get to work

In contrast, casual riders have higher ride numbers on weekends and fewer rides on weekdays.

3. Ride Duration: The average trip duration of casual members is higher than subscribers. The average trip duration for casual riders is consistently higher on weekends compared to weekdays, indicating that they may use the bikes for leisure activities during weekends. Annual members, on the other hand, have a relatively constant average trip duration throughout the week.

Recommendations

1. Targeted Weekend Campaigns: Given that casual riders have a strong presence on weekends and their rides tend to be longer, the marketing department could run targeted weekend campaigns to encourage more weekend riders to become annual members. Promotions and discounts specifically designed for weekend riders may be effective.

2. Member Conversion Strategy: To convert casual riders into annual members, Cyclistic can highlight the benefits of becoming a member, such as cost savings for frequent riders, access to exclusive features, and priority bike availability during peak hours. These benefits should be communicated clearly through digital media and other marketing channels.

3. Customer Engagement: Cyclistic should focus on engaging casual riders through digital media. This engagement can include providing them with valuable content, such as biking tips, local biking routes, and information about Cyclistic's services. Engaging content can create a sense of community and loyalty among casual riders, making them more likely to consider an annual membership.

4. Weekday Promotions for Annual Members: To maintain and attract more annual members, Cyclistic can offer weekday promotions or incentives, such as discounts on annual memberships, for those who frequently use the service during weekdays. This strategy can encourage annual members to use the service more during weekdays.

5. Data-Driven Decision Making: Continuously monitor rider data to identify trends and adjust marketing strategies accordingly. A data-driven approach will help Cyclistic stay responsive to changing rider behavior and preferences.

6. User Experience Enhancement: Ensure a seamless and user-friendly experience for both annual members and casual riders through the Cyclistic app or website. Make it easy for casual riders to explore the benefits of annual membership and facilitate the conversion process.

7. Feedback Mechanism: Implement a feedback mechanism to gather insights from both annual members and casual riders. Understand their specific needs, preferences, and pain points to tailor marketing strategies and improve the overall customer experience.