Case Study: How Does a Bike-Share Navigate Speedy Success?

Lingyu Tan

Dec 9, 2021

Contents

1 Background (Ask)

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime.

Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.

Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, Moreno believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, Moreno believes there is a very good chance to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs.

Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the marketing analyst team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

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?

In this report, I will try to answer **the first question** with the following deliverables thus helping Cyclistic design marketing strategies:

1. A clear statement of the business task
2. A description of all data sources used
3. Documentation of any cleaning or manipulation of data
4. A summary of your analysis
5. Supporting visualizations and key findings
6. Your top three recommendations based on your analysis

2 Data Source (Prepare)

The datasets which are used in this case study can be accessed here. The datasets include previous 12 months of historical trip data (2020/12 - 2021/11) of Cyclistic, a fictional company. The data has been made available by Motivate International Inc. under this license.

2.1 Setting up Environment

```
library(tidyverse)
library(janitor)
library(scales)
```

2.2 Loading Datasets

```
### Load datasets
# data <- read_csv("Data/202012-divvy-tripdata.csv")
data_202012 <- read_csv("Data/202012-divvy-tripdata.csv")
data_202101 <- read_csv("Data/202101-divvy-tripdata.csv")
data_202102 <- read_csv("Data/202102-divvy-tripdata.csv")
data_202103 <- read_csv("Data/202103-divvy-tripdata.csv")
data_202104 <- read_csv("Data/202104-divvy-tripdata.csv")
data_202105 <- read_csv("Data/202105-divvy-tripdata.csv")
data_202106 <- read_csv("Data/202106-divvy-tripdata.csv")
data_202107 <- read_csv("Data/202107-divvy-tripdata.csv")
data_202108 <- read_csv("Data/202108-divvy-tripdata.csv")
data_202109 <- read_csv("Data/202109-divvy-tripdata.csv")
```

```
data_202110 <- read_csv("Data/202110-divvy-tripdata.csv")
data_202111 <- read_csv("Data/202111-divvy-tripdata.csv")

### Combine datasets
data <- rbind(data_202012, data_202101, data_202102, data_202103,
              data_202104, data_202105, data_202106, data_202107,
              data_202108, data_202109, data_202110, data_202111)

# rm(data_202012, data_202101, data_202102, data_202103,
#    data_202104, data_202105, data_202106, data_202107,
#    data_202108, data_202109, data_202110, data_202111)
```

2.3 Data Structure

```
glimpse(data)
```

```
## Rows: 5,479,096
## Columns: 13
## $ ride_id            <chr> "70B6A9A437D4C30D", "158A465D4E74C54A", "5262016E0F~
## $ rideable_type      <chr> "classic_bike", "electric_bike", "electric_bike", "~
## $ started_at         <dttm> 2020-12-27 12:44:29, 2020-12-18 17:37:15, 2020-12--~
## $ ended_at           <dttm> 2020-12-27 12:55:06, 2020-12-18 17:44:19, 2020-12--~
## $ start_station_name <chr> "Aberdeen St & Jackson Blvd", NA, NA, NA, NA, NA, N~
## $ start_station_id   <chr> "13157", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ end_station_name   <chr> "Desplaines St & Kinzie St", NA, NA, NA, NA, NA, NA~
## $ end_station_id     <chr> "TA1306000003", NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_lat          <dbl> 41.87773, 41.93000, 41.91000, 41.92000, 41.80000, 4~
## $ start_lng          <dbl> -87.65479, -87.70000, -87.69000, -87.70000, -87.590~
## $ end_lat            <dbl> 41.88872, 41.91000, 41.93000, 41.91000, 41.80000, 4~
## $ end_lng            <dbl> -87.64445, -87.70000, -87.70000, -87.70000, -87.590~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

In this datasets, there are overall 5,479,096 observations and 13 variables, including 7 character, 4 double, 2 date-time. Just to clarify, `start_lat` means the latitude of starting point, and `start_lng` means the longitude of starting point.

3 Cleaning Data (Process)

First we need to check if there is duplicated data in the dataset.

```
sum(duplicated(data$ride_id))
```

```
## [1] 0
```

We see that `ride_id` seems to be a unique identifier without repetition thus there seems to be no duplicates. We can check all the records without `ride_id` to see what happens.

```
### There is no duplicated rows so the sum is zero
# sum(duplicated(data[, -1]))
# = 502

# data_no_id <- data[, -1]
# dup <- duplicated(data_no_id)
# dup_data <- data[dup, ]
# dup_data[order(dup_data$started_at), ]

# nrow(unique(data[, -1]))
# = 5478594
```

There are 502 out of 5,479,096 records are duplicates without considering `ride_id`. We can have a look at one duplicate below:

```
data %>% filter(started_at == "2020-12-01 14:17:37 UTC")
```

```
## # A tibble: 2 x 13
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 0DF824~ docked_bike   2020-12-01 14:17:37 2020-12-01 14:26:31 Theater on the ~
## 2 6C8319~ docked_bike   2020-12-01 14:17:37 2020-12-01 14:26:31 Theater on the ~
## # ... with 8 more variables: 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>
```

We clearly see that all records of these two observations are identical except `ride_id`. It is possible that a couple start their journey at the same time/place and end simultaneously. Therefore, we decide not to remove these potential "duplicates". (If given more information such as `user_id`, we could better decide if it is necessary to remove these "duplicates".)

In this report we are not interested in the information related to the locations (`start_station_name`, `start_station_id`, `end_station_name`, `end_station_id`, `start_lat`, `start_lng`, `end_lat`, `end_lng`) so we will not clean these data. We know for sure that theoretically `ended_at` must happen after `started_at`, we can check this by the following code.

```
outlier <- data %>% filter(started_at > ended_at) %>% arrange(started_at)
nrow(outlier)
```

```
## [1] 581
```

```
print(outlier)
```

```
## # A tibble: 581 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 B42B020CCE2FCCB1 docked_bike   2020-12-01 05:45:22 2020-12-01 05:44:49
##  2 8880EE92E8ED185C docked_bike   2020-12-01 07:14:28 2020-12-01 07:13:38
##  3 033B361CE2A558BE docked_bike   2020-12-01 07:16:21 2020-12-01 07:15:48
##  4 ACA7DA442E8BCB55 docked_bike   2020-12-01 07:17:34 2020-12-01 07:17:21
##  5 0EC60FE7864F9175 docked_bike   2020-12-01 07:19:01 2020-12-01 07:18:12
##  6 A14ED92EA9980118 docked_bike   2020-12-01 07:21:16 2020-12-01 07:20:43
##  7 551145A3F9B70668 docked_bike   2020-12-01 07:28:05 2020-12-01 07:27:46
##  8 FB67CE9A11B8BBE4 docked_bike   2020-12-01 08:15:23 2020-12-01 08:14:52
##  9 F7B46074A99AA1F6 docked_bike   2020-12-01 09:37:26 2020-12-01 09:37:06
## 10 171CCDE1037D7C5C docked_bike   2020-12-01 09:47:03 2020-12-01 09:46:33
## # ... with 571 more rows, and 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>
```

After observing 581 abnormal data observations, we find that from the 57th outlier, there is a large difference between `started_at` and `ended_at`, which was probably due to errors when collecting data.

```
print(outlier[57, ])
```

```
## # A tibble: 1 x 13
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 BE1F69~ docked_bike   2020-12-15 11:34:23 2020-11-25 08:00:15 Kingsbury St & ~
## # ... with 8 more variables: 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>
```

```
outlier_202012 <- data_202012 %>% filter(started_at > ended_at)
nrow(outlier_202012)
```

```
## [1] 434
```

```
print(outlier_202012)
```

```
## # A tibble: 434 x 13
##    ride_id          rideable_type started_at          ended_at
##    <chr>            <chr>         <dttm>              <dttm>
##  1 B7DF480FC433DA00 electric_bike 2020-12-15 11:48:35 2020-11-25 13:21:43
##  2 923B69F36CE1E2D3 electric_bike 2020-12-15 11:50:32 2020-11-25 14:06:37
##  3 4804C00B5CA844EA electric_bike 2020-12-15 12:05:43 2020-11-25 20:20:05
##  4 D343E9446089AAF2 electric_bike 2020-12-15 12:11:41 2020-11-25 20:03:07
##  5 6E67A4AAB95CA751 electric_bike 2020-12-15 12:15:37 2020-11-25 16:38:59
##  6 5577F015C086E5C4 electric_bike 2020-12-15 11:51:59 2020-11-25 14:16:40
##  7 34552B053B1D01C7 electric_bike 2020-12-15 11:37:49 2020-11-25 09:04:08
##  8 5B09680D33DFD894 electric_bike 2020-12-15 11:42:13 2020-11-25 10:50:56
##  9 43802DDD3A14F821 docked_bike   2020-12-02 08:14:17 2020-12-02 08:14:12
## 10 CDA12952E2FC2D99 docked_bike   2020-12-15 11:39:19 2020-11-25 10:16:59
## # ... with 424 more rows, and 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>
```

A large proportion (434/581) of outliers were in Dec, 2020, where somehow `started_at` is later than `ended_at` for around 20 days. We didn't know what was happened when collecting these data so we decided to simply remove the obviously abnormal data where the `started_at` is later than `ended_at` for more than 1 day, otherwise we assume that the records of `started_at` and `ended_at` were misplaced so we swap them. (Extra information is needed to decide how to deal with the abnormal data.)

```r
# Delete observations with `started_at` > `ended_at` for 1 day
data_cleaned <- data %>% filter(started_at - ended_at <= 86400) # Delete > 1 days

# Record id of abnormal observations
swap_id <- which(data_cleaned$started_at > data_cleaned$ended_at)

# Swap reversed values of `started_at` and `ended_at`
temp <- data_cleaned[swap_id, "started_at"]
data_cleaned[swap_id, "started_at"] <- data_cleaned[swap_id, "ended_at"]
data_cleaned[swap_id, "ended_at"] <- temp
```

Now the data is clean, and we transform/trim the data so that it only contains necessary information.

```r
data_cleaned_trimmed <- data_cleaned %>%
  mutate(ride_length = ended_at - started_at, day_of_week_char = weekdays(started_at)) %>%
  mutate(ride_length_2 = as.integer(ride_length)) %>%
  mutate(rideable_type_2 = as.factor(rideable_type)) %>%
  mutate(member_casual_2 = as.factor(member_casual)) %>%
  mutate(day_of_week = factor(day_of_week_char,
                              levels = c("Sunday", "Monday", "Tuesday", "Wednesday",
                                         "Thursday", "Friday", "Saturday"), ordered = TRUE)) %>%
  # select(ride_id, rideable_type = rideable_type_2, member_casual = member_casual_2,
  #         ride_length = ride_length_2, day_of_week)
  select(ride_id, rideable_type = rideable_type_2, member_casual = member_casual_2,
         ride_length = ride_length_2, day_of_week, started_at, ended_at,
         start_lat, start_lng, end_lat, end_lng)
```

4 Data Analysis (Analyze)

4.1 Descriptive Analysis

First we run a descriptive analysis on the trimmed cleaned data:

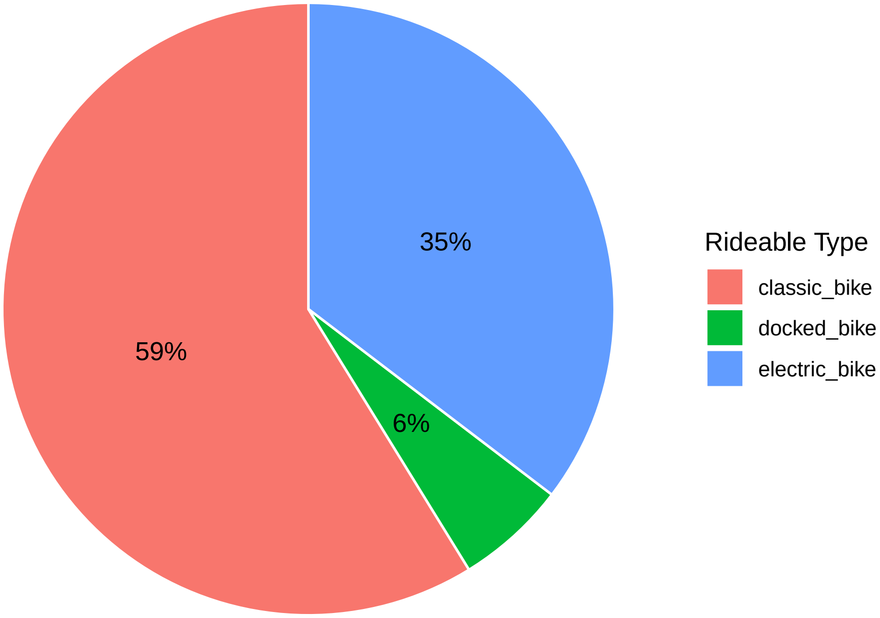
```r
summary(data_cleaned_trimmed)
```

```
##    ride_id                rideable_type     member_casual     ride_length
##  Length:5478718     classic_bike :3221009   casual:2489268   Min.   :      0
##  Class :character   docked_bike  : 320210   member:2989450   1st Qu.:    410
##  Mode  :character   electric_bike:1937499                    Median :    728
##                                                              Mean   :   1328
##                                                              3rd Qu.:   1320
##                                                              Max.   :3356649
##
##     day_of_week       started_at                      ended_at
##  Sunday   :850362   Min.   :2020-12-01 00:01:15   Min.   :2020-12-01 00:06:33
##  Monday   :689813   1st Qu.:2021-05-31 17:22:43   1st Qu.:2021-05-31 17:53:29
##  Tuesday  :732662   Median :2021-07-25 12:21:16   Median :2021-07-25 12:47:35
##  Wednesday:733686   Mean   :2021-07-17 16:04:45   Mean   :2021-07-17 16:26:53
##  Thursday :711303   3rd Qu.:2021-09-15 16:33:03   3rd Qu.:2021-09-15 16:51:48
##  Friday   :783935   Max.   :2021-11-30 23:59:56   Max.   :2021-12-02 06:41:33
##  Saturday :976957
##    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   :4737    NA's   :4737
```

We have 5 variables: unique `ride_id`; three types of `rideable_type`: *classic_bike, docked_bike, electric_bike*; two types of `member_causal`: *member, casual*; numerical `ride_length` where the average length is 1328 seconds (22.13 minutes) and the longest length is 3356649 seconds (38.85 days), which seems acceptable; seven `day_of_week`: Sunday to Saturday.

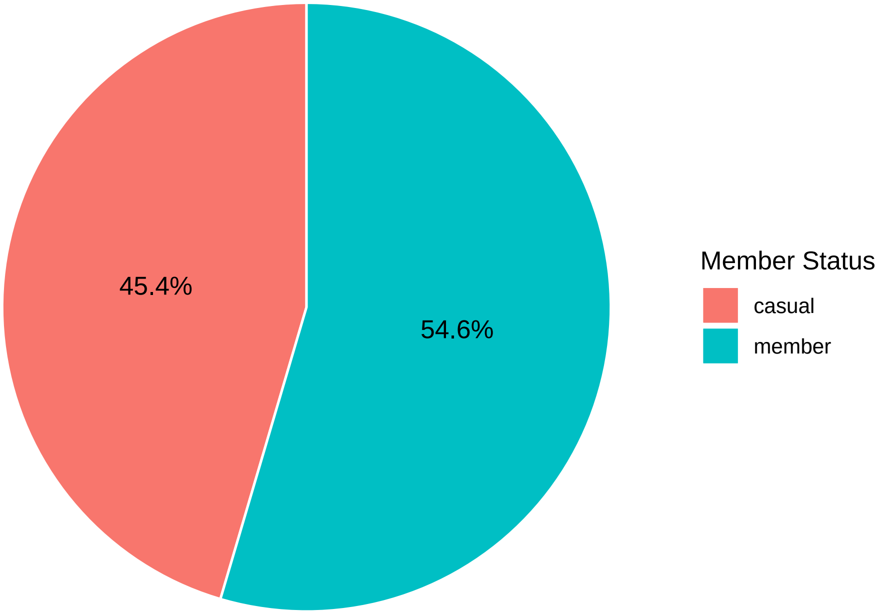
From the below figure we see that 59% of the bikes are classic bikes, and there are 35% electric bikes and only 6% docked bikes.

```
data_cleaned_trimmed %>%
  group_by(rideable_type) %>%
  count() %>%
  ungroup() %>%
  mutate(perc = `n` / sum(`n`)) %>%
  mutate(labels = scales::percent(perc)) %>%

  ggplot(aes(x = "", y = n, fill = rideable_type)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar("y", start = 0) +
  theme_void() +
  geom_text(aes(label = labels), position = position_stack(vjust = 0.5)) +
  guides(fill = guide_legend(title = "Rideable Type"))
```



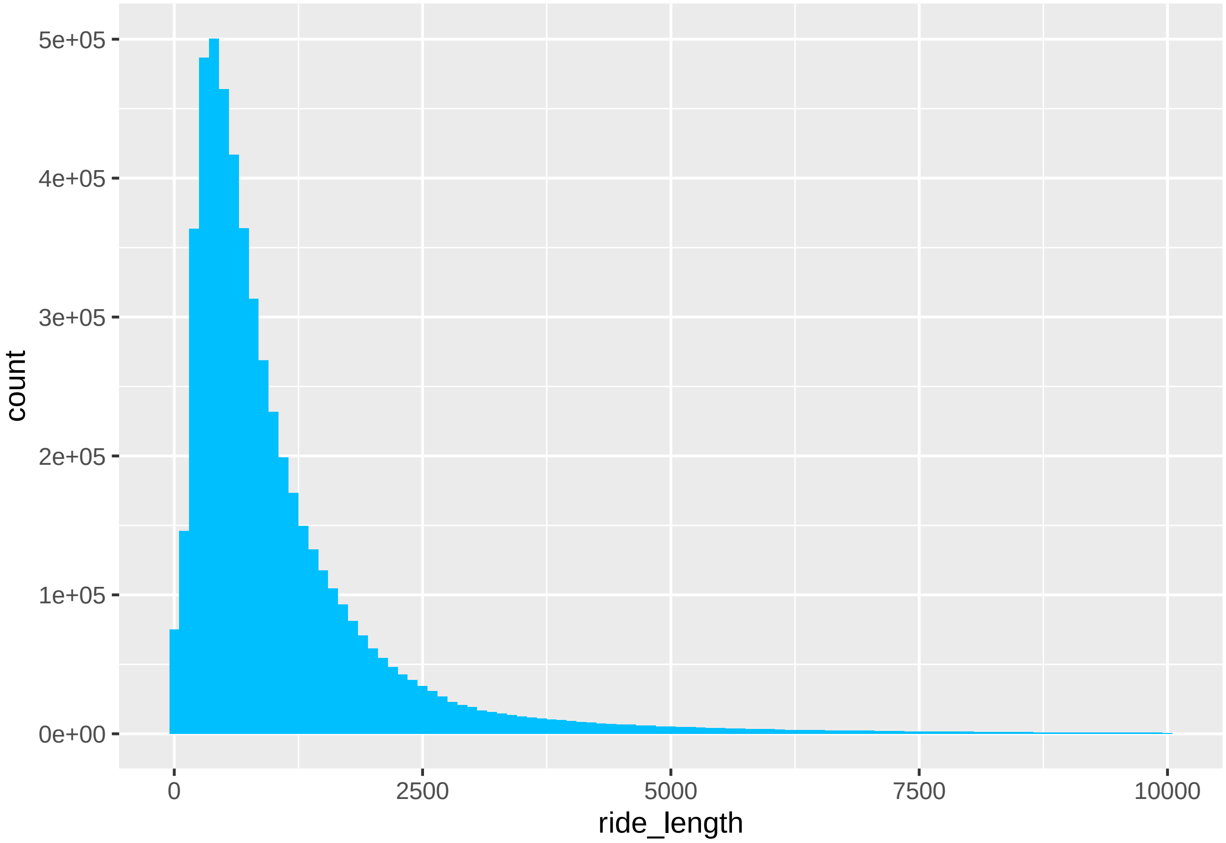
The percentage of members is 54.6%, which is slightly greater than that of causal users (45.4%).

```r
data_cleaned_trimmed %>%
  group_by(member_casual) %>%
  count() %>%
  ungroup() %>%
  mutate(perc = `n` / sum(`n`)) %>%
  mutate(labels = scales::percent(perc)) %>%

  ggplot(aes(x = "", y = n, fill = member_casual)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar("y", start = 0) +
  theme_void() +
  geom_text(aes(label = labels), position = position_stack(vjust = 0.5)) +
  guides(fill = guide_legend(title = "Member Status"))
```



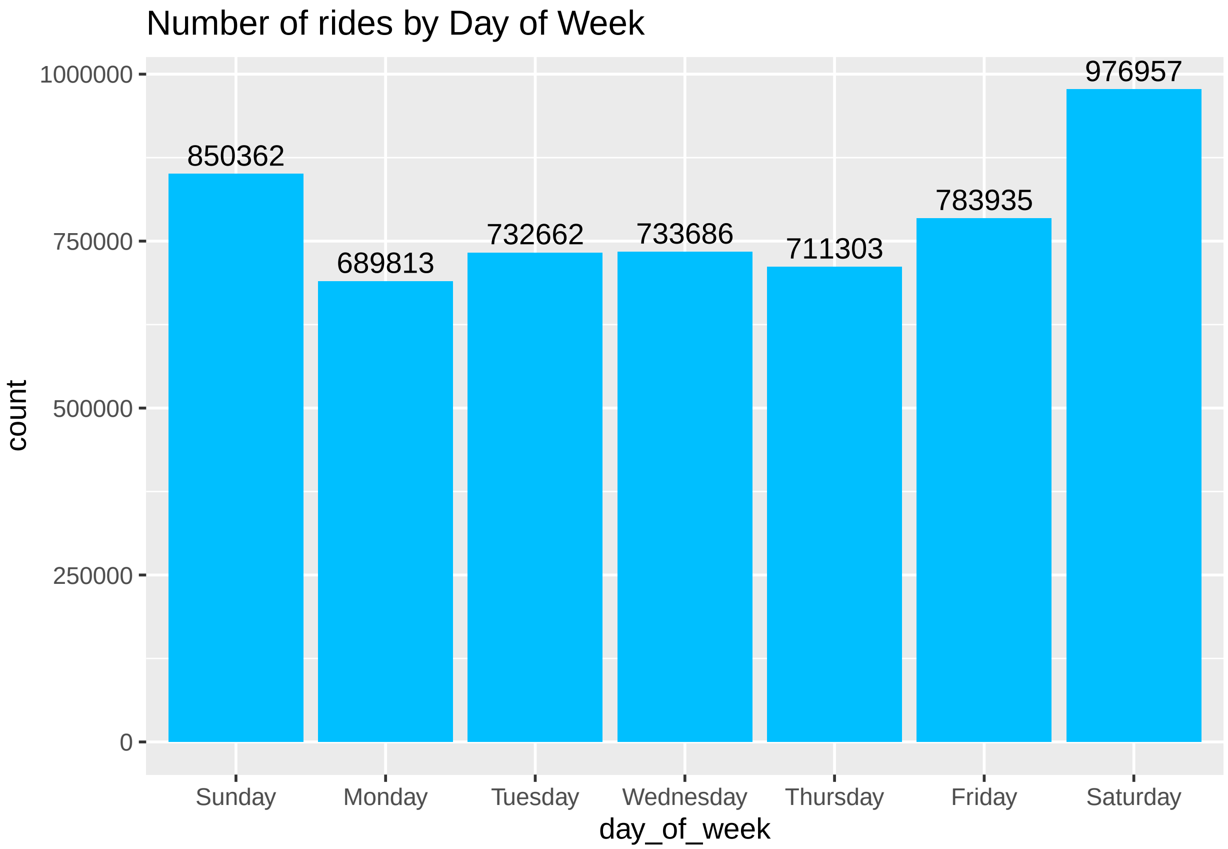
By plotting a simple histogram we see that the distribution of ride lengths is right-skewed with occasional extremely large values. We notice that most of the ride lengths are less than 10,000 seconds so we filter the data and take a closer look at the distribution.By observing the following plot we see that the mode lies at around 500 seconds, after which the number of rides seems to be monotonically decreasing.

```r
data_cleaned_trimmed %>%
  filter(ride_length < 10000) %>%
  ggplot() +
  geom_histogram(aes(ride_length), binwidth = 100, fill = "#00BFFF")
```



Also, we see that during weekdays, the number of rides is at a relatively low level (about 700,000) and shows obvious increase when it comes to weekends especially Saturdays (nearly 1,000,000).

```
data_cleaned_trimmed %>%
  ggplot(aes(x = day_of_week)) +
  geom_bar(fill = "#00BFFF") +
  geom_text(stat='count', aes(label = after_stat(count)), vjust = -0.4) +
  labs(title = "Number of rides by Day of Week")
```



4.2 Comparison between Causal Riders and Annual Members

First, we calculate the average ride length for members and casual rides separately.

```
data_cleaned_trimmed %>%
  group_by(member_casual) %>%
  summarise(`Average ride length` = mean(ride_length))
```

```
## # A tibble: 2 x 2
##   member_casual ‘Average ride length‘
##   <fct>                         <dbl>
## 1 casual                        1931.
## 2 member                         826.
```

We find that causal users are likely to spend more time (18 minutes average) than members in a single ride. It might be because that casual users tend to use our bikes when the journey is relatively long, whilst members just frequently use our bikes regardless of the length of journey.

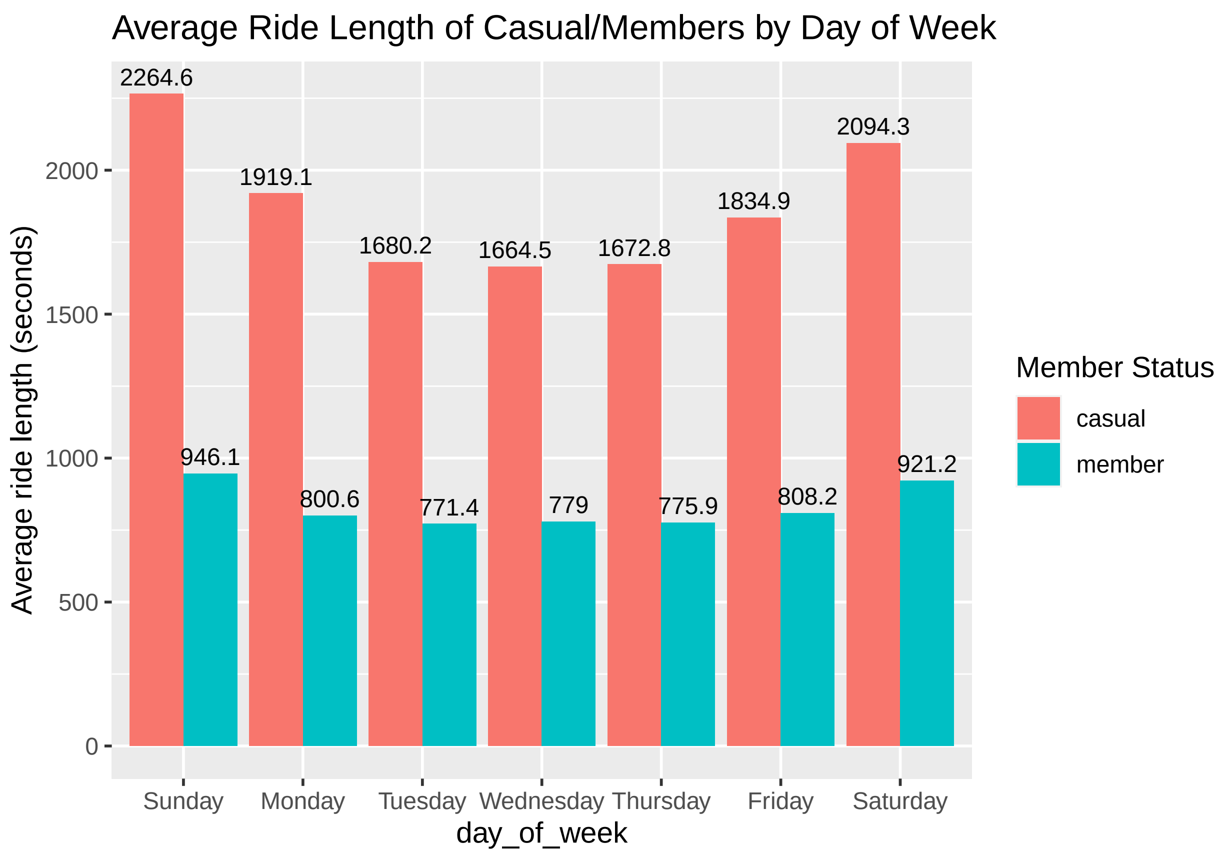
Then we calculate the average ride length for members and casual rides by `day_of_week`:

```
average_mc_dow_summary <- data_cleaned_trimmed %>%
  group_by(member_casual, day_of_week) %>%
  summarise(`Average ride length` = mean(ride_length), .groups = "drop")

print(average_mc_dow_summary, n = nrow(average_mc_dow_summary))
```

```
## # A tibble: 14 x 3
##    member_casual day_of_week ‘Average ride length‘
##    <fct>         <ord>                       <dbl>
##  1 casual        Sunday                      2265.
##  2 casual        Monday                      1919.
##  3 casual        Tuesday                     1680.
##  4 casual        Wednesday                   1665.
##  5 casual        Thursday                    1673.
##  6 casual        Friday                      1835.
##  7 casual        Saturday                    2094.
##  8 member        Sunday                       946.
##  9 member        Monday                       801.
## 10 member        Tuesday                      771.
## 11 member        Wednesday                    779.
## 12 member        Thursday                     776.
## 13 member        Friday                       808.
## 14 member        Saturday                     921.
```

And we use the following visualisation to help us better understand the user behaviour:

```
average_mc_dow_summary %>%
ggplot(aes(fill = member_casual, y = `Average ride length`, x = day_of_week)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = round(`Average ride length`, 1)),
            position = position_dodge(0.9), vjust = -0.5, size = 3) +
  guides(fill = guide_legend(title = "Member Status")) +
  labs(title = "Average Ride Length of Casual/Members by Day of Week",
       y = "Average ride length (seconds)")
```

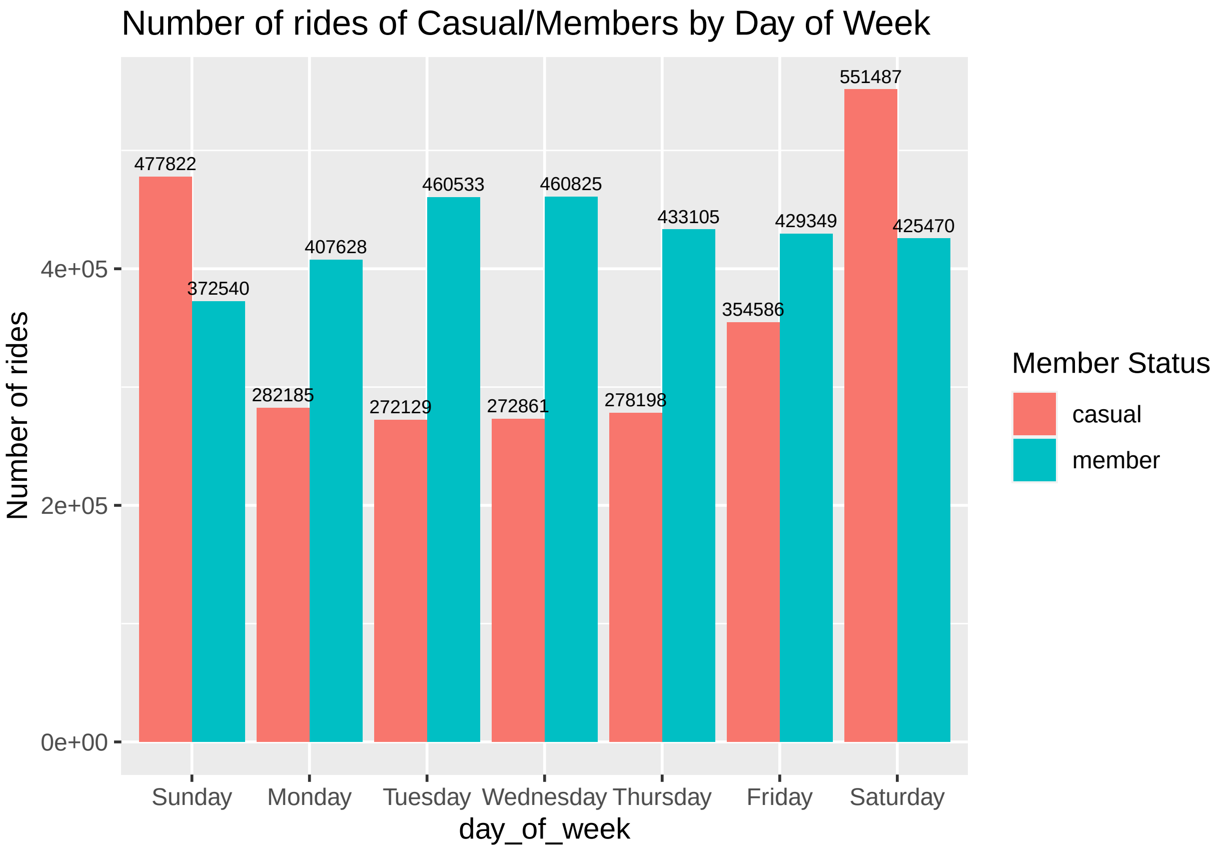


We see that the average ride length of members is relatively stable amongst days in a week, but for casual users the ride length at weekends is obviously longer.

```
data_cleaned_trimmed %>%
  group_by(member_casual) %>%
  count()
```

```
## # A tibble: 2 x 2
## # Groups:   member_casual [2]
##   member_casual       n
##   <fct>           <int>
## 1 casual        2489268
## 2 member        2989450
```

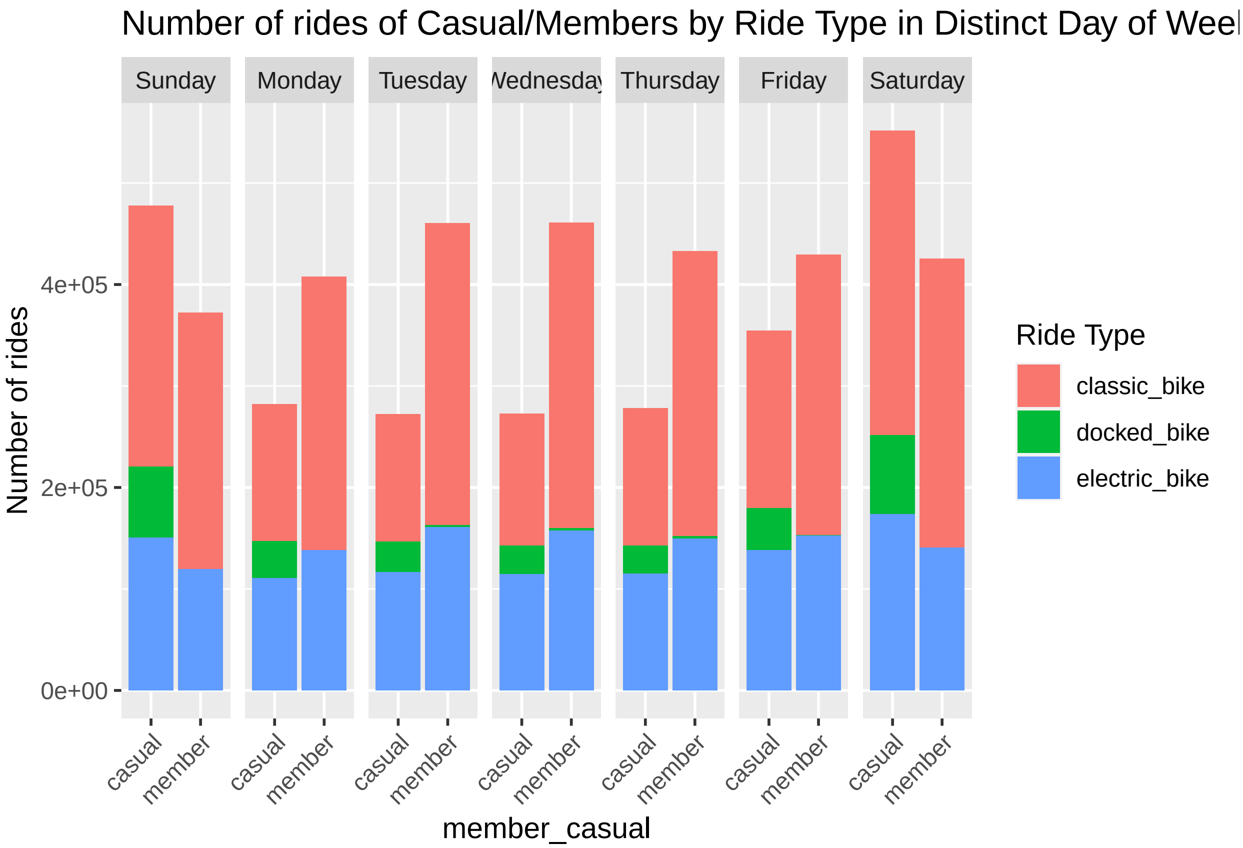
```
data_cleaned_trimmed %>%
  group_by(day_of_week, member_casual) %>%
  count() %>%

  ggplot(aes(fill = member_casual, y = n, x = day_of_week)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = n),
            position = position_dodge(0.9), vjust = -0.5, size = 2.5) +
  guides(fill = guide_legend(title = "Member Status")) +
  ylab("Number of rides") +
  labs(title = "Number of rides of Casual/Members by Day of Week")
```



Also, the number of rides shows similar features as the above average ride length. For annual members the number of rides seems stable where the minimum is on Sunday, but for casual users the numbers of rides on Friday, Saturday and Sunday are much greater than those of others, especially in Saturday. We guess that casual users are likely to use our service in Friday/weekends to travel whilst members usually use our services on weekdays to commute to work.

Next, we can further look at the types of bikes users borrow as follows:

```r
data_cleaned_trimmed %>%
  group_by(day_of_week, member_casual, rideable_type) %>%
  count() %>%

  ggplot(aes(fill = rideable_type, y = n, x = member_casual)) +
  geom_bar(stat = "identity", position = "stack") +
  guides(fill = guide_legend(title = "Ride Type")) +
  ylab("Number of rides") +
  labs(title = "Number of rides of Casual/Members by Ride Type in Distinct Day of Week") +
  facet_wrap(~day_of_week, nrow = 1) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



The major difference is that almost no member use docked bike and quite a few casual users use docked bikes especially in Saturday and Sunday.

5 Conclusion

From the above analysis, we can see the following differences between the behaviors of members and casual users:

1. The average ride length of casual users (32.2 minutes) is longer than annual members (13.8 minutes), and casual users tend to ride a longer length on weekends (average 37.7 minutes on Sunday, 34.9 minutes on Saturday).
2. Overall the number of rides of casual users (2,489,268) is 17% less than members (2,989,450). The number of rides of casual users is less than members on weekdays, whilst it is more than members on weekends.
3. Classic bikes are most frequently used, electric bikes are the second, and docked bikes are the least used. Members usually do not use docked bikes, and quite a few casual users use docked bikes especially on weekends. Members like using classic bikes most and the frequency is about the same level during a week. Casual users also like using classic bikes and much more frequent on weekends. For electric bikes, the number of rides between two groups are approximately the same. It seems that casual users use electric bikes slightly frequently on weekends, whilst members use electric bikes slightly frequently on weekdays.

6 Recommendation

Given the above conclusions and aiming to turn more casual users to members, we would make the following top three recommendations:

1. We can charge casual users the rent based on the ride length. The longer the ride is, the more we charge, and the average charge of longer ride (e.g. more than 30 minutes) should be much more than annual members.
2. Casual users are more likely to use our bikes on weekends so we can slightly increase our price (single-ride pass, full-day pass) on weekends. Also, we can create an annual weekend pass, say, free to use the service only on weekends and the price is slightly lower than normal annual pass, to attract more casual users who usually use our services on weekends to convert into annual members.
3. Replace some docked bikes with electric bikes and classic ones, and slight increase the cost of using docked bikes.

Furthermore, there are lots of other aspects we can work on. For instance, how many times will a all-day pass holder use our bikes. Based on the prices of all-day passes and single-ride passes, should we adjust them? We notice that many members tend to use our bikes to commute to work. Is it possible for us to advertise the benefits of riding to work? Also, to attract more casual users who use our bikes on weekends to travel, can we make a guide of riding routes in the city so people are willing to try them every weekends finding it cheaper to become annual members? (If possible, we can collect user id (unique but encrypted) to analyse individual behaviour to make better strategies.)