Cyclistic Project

Hasan Sezer

2024-08-06

The aim of the project

Cyclistic is a fictionary bike-share company. Stakeholders aim to maximize the number of annual membership. Thus, my objective is to identfy differences between annual members and casual riders, which can turn into a data-driven marketing strategy to convert casual riders to annual members. At the end of the analysis/report, I will offer tentative campaign ideas to convert subscribers into customers.

Data: Trips_2019_Q1.csv

Data Source: https://divvy-tripdata.s3.amazonaws.com/index.html

Project Questions:

1. How do causal riders and annual members/customers use Cyclistic differently?
2. How can subscribers/casual riders become customers?

Road Map

I will focus on **user type** and its relation with different variables such as **trip duration**, **trip days**, **gender of the user**, and **age of the user** and their interaction with each other. I will do data cleaning and formatting, then conduct EDA. Finally, I will build a logistic regression model to find which variables are significant on the **user type.**

Load data

```
trips_2019<-read.csv("Trips_2019_Q1.csv")
```

Load necessary packages

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.5.1     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- 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 errors
```

A. Data inspection and preprocessing

Inspect the data set quickly; look at variables and their types. Do wrangling if needed.

```
str(trips_2019)
```

```
## 'data.frame':    365069 obs. of  12 variables:
##  $ trip_id          : int  21742443 21742444 21742445 21742446 21742447 21742448 21742449 21742450 2
##  $ start_time       : chr  "2019-01-01 00:04:37" "2019-01-01 00:08:13" "2019-01-01 00:13:23" "2019-0
##  $ end_time         : chr  "2019-01-01 00:11:07" "2019-01-01 00:15:34" "2019-01-01 00:27:12" "2019-0
##  $ bikeid           : int  2167 4386 1524 252 1170 2437 2708 2796 6205 3939 ...
##  $ tripduration     : chr  "390.0" "441.0" "829.0" "1,783.0" ...
##  $ from_station_id  : int  199 44 15 123 173 98 98 211 150 268 ...
##  $ from_station_name: chr  "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave & 18th St"
##  $ to_station_id    : int  84 624 644 176 35 49 49 142 148 141 ...
##  $ to_station_name  : chr  "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "Western Ave
##  $ usertype         : chr  "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender           : chr  "Male" "Female" "Female" "Male" ...
##  $ birthyear        : int  1989 1990 1994 1993 1994 1983 1984 1990 1995 1996 ...
```

- There are 14 variables and 365069 rows/observations.
- Variable names make sense and error-free.
- **trip_duration**, **start_time**, and **end_time** are in character/string format. Change the data type.
- In **tripduration**, the commas in certain rows such as "1,783.0" cannot be handled by as.numeric function, hence NAs emerged. I need to handle comma, then convert it to a numeric data. Then, find the trip duration in minutes.

```
trips_2019$tripduration <- gsub(",", "", trips_2019$tripduration)
```

```
trips_2019$tripduration<-as.numeric(gsub(","," ", trips_2019$tripduration))
```

```
trips_2019$tripduration<-trips_2019$tripduration/60
```

Finally, start and end_time need a format change.

```
trips_2019$start_time <- as.POSIXct(trips_2019$start_time, format = "%Y-%m-%d %H:%M:%S")
trips_2019$end_time <- as.POSIXct(trips_2019$end_time, format = "%Y-%m-%d %H:%M:%S")
```

- Missing values: Only gender has NAs, which is 18023. This is around %20 of the sample size.

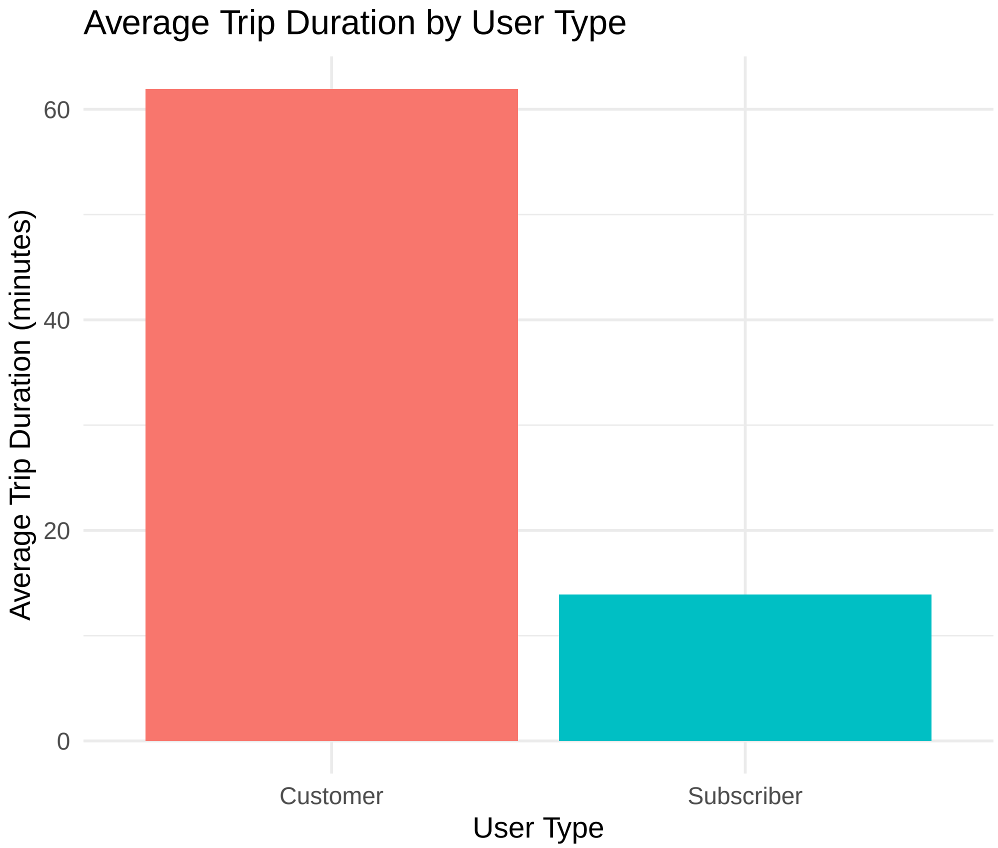
```
missing_summary <- trips_2019%>%
  summarise(across(everything(), ~ sum(is.na(.))))%>%
  pivot_longer(everything(), names_to = "Variable", values_to = "MissingValues")

print(missing_summary)
```

```
## # A tibble: 12 x 2
##    Variable          MissingValues
##    <chr>                     <int>
##  1 trip_id                       0
##  2 start_time               0
##  3 end_time                 0
##  4 bikeid                   0
##  5 tripduration             0
##  6 from_station_id          0
##  7 from_station_name        0
##  8 to_station_id            0
##  9 to_station_name          0
## 10 usertype                 0
## 11 gender                   0
## 12 birthyear            18023
```

B. Exploratory Data Analysis (EDA)

```
duration_by_usertype <- trips_2019 %>%
  group_by(usertype) %>%
  summarize(avg_duration = mean(tripduration, na.rm = TRUE))
```

```
ggplot(data=duration_by_usertype)+
  geom_bar(stat="identity", mapping = aes(x=usertype, y=avg_duration, fill=usertype))+
  labs(x = "User Type", y = "Average Trip Duration (minutes)", title = "Average Trip Duration by User Ty
  theme_minimal()
```



1. Trip Duration by User Type

Interpretation:

Customers' trip duration time is significantly higher than subscribers, or causal riders. If revenue is correlated with trip duration, then it is very reasonable to increase the customer size.

1.1 Trip Duration by User Type and Gender

Gender has a lot of missing values, or empty cells to be more precisely. Handle it first.

```r
table(trips_2019$gender)
```

```
##
##         Female   Male
##  19711  66918 278440
```

```r
trips_2019 <- trips_2019 %>%
  mutate(gender = na_if(gender, "")) # converting empty cells to NAs
```

```r
trips_2019 <- trips_2019 %>%
  filter(!is.na(gender)) # then filter our NAs introduced above
```

```r
table(trips_2019$gender) # All good!
```

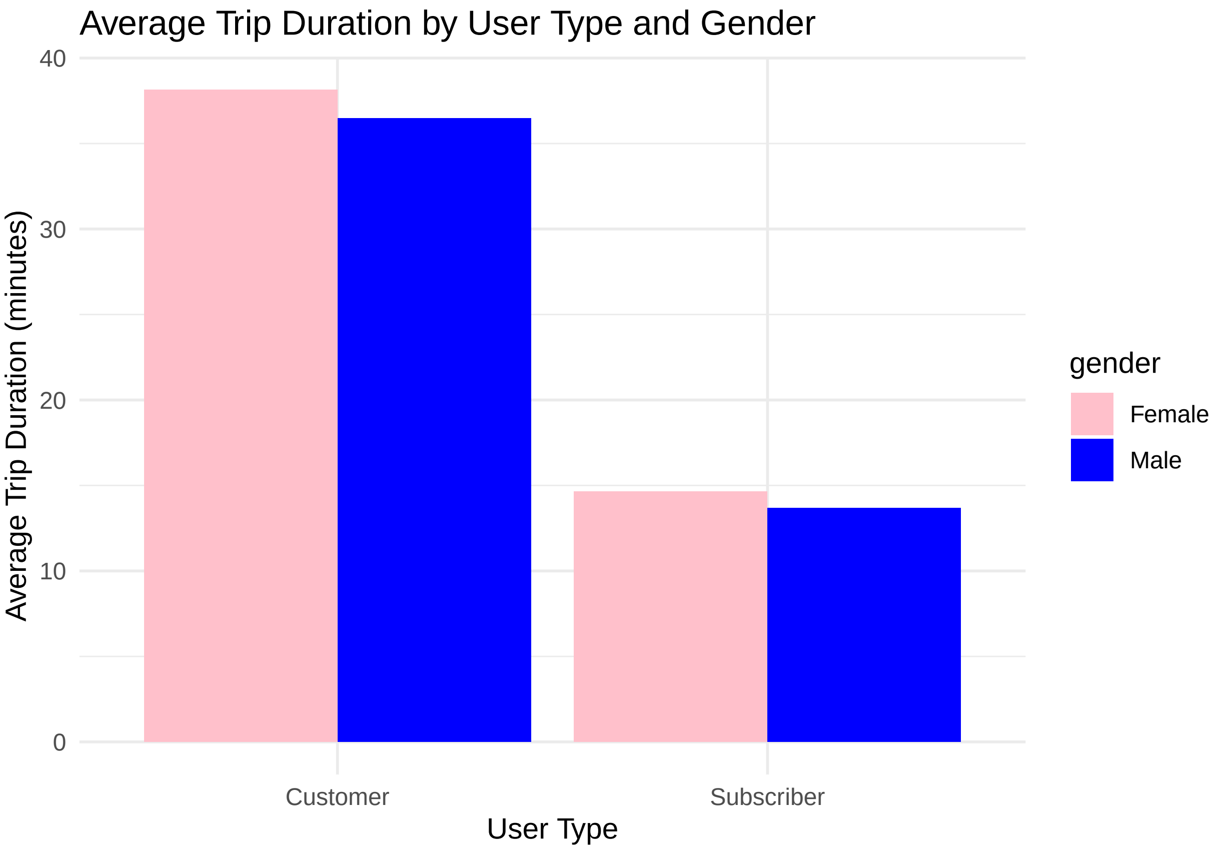
```
##
## Female   Male
##  66918 278440
```

```r
duration_by_usertype_gender <- trips_2019 %>%
  group_by(usertype, gender) %>%
  summarize(avg_duration = mean(tripduration, na.rm = TRUE), .groups = 'drop')
```

```r
duration_by_usertype_gender
```

```
## # A tibble: 4 x 3
##   usertype   gender avg_duration
##   <chr>      <chr>         <dbl>
## 1 Customer   Female         38.2
## 2 Customer   Male           36.5
## 3 Subscriber Female         14.7
## 4 Subscriber Male           13.7
```

```r
ggplot(duration_by_usertype_gender, aes(x = usertype, y = avg_duration, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Trip Duration by User Type and Gender",
       x = "User Type",
       y = "Average Trip Duration (minutes)") +
  theme_minimal() +
  scale_fill_manual(values = c("Male" = "blue", "Female" = "pink"))
```



Interpretation:

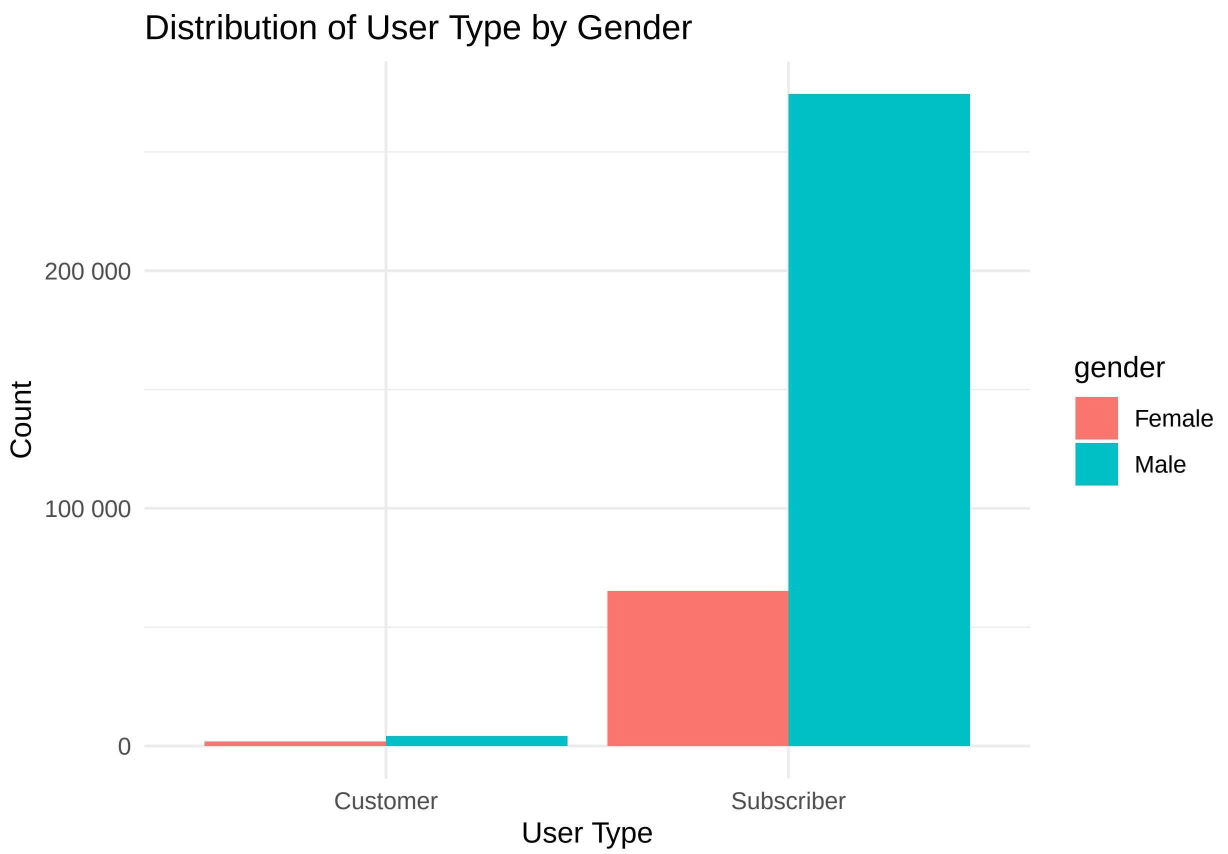
User types' trip duration is very close for males and females. In both user categories, female and male have almost identical trip duration, where females ride bikes slightly **longer** than males.

```
usertype_gender_count <- trips_2019 %>%
  count(usertype, gender)

print(usertype_gender_count)
```

2. Gender vs User Type

```
##     usertype gender      n
## 1   Customer Female   1875
## 2   Customer   Male   4060
## 3 Subscriber Female  65043
## 4 Subscriber   Male 274380
```

```
ggplot(usertype_gender_count, aes(x = usertype, y = n, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous(labels = scales::label_number()) +
  labs(title = "Distribution of User Type by Gender",
       x = "User Type",
       y = "Count") +
  theme_minimal()
```



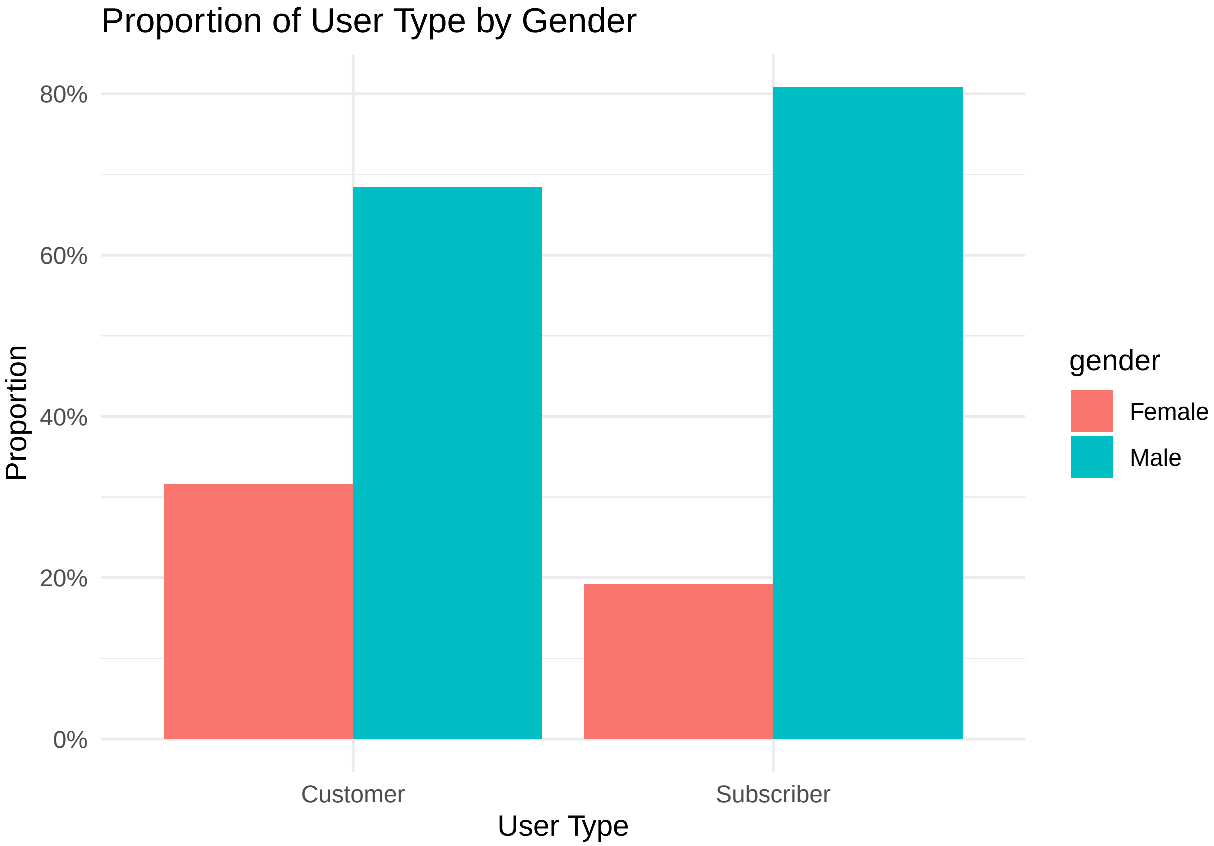
Interpretation: Frequency of bike use is higher in males than females in both user types. Due to the great difference between customers and subscribers in bike use frequency, the plot is not very intuitive in identifying gender variation in customer group. Thus, I will calculate proportion of bike use frequency of gender in each user type.

Proportion of gender in each user type

```r
usertype_gender_prop <- trips_2019 %>%
  count(usertype, gender) %>%
  group_by(usertype) %>%
  mutate(prop = n / sum(n)) %>%
  ungroup()

print(usertype_gender_prop)
```

```
## # A tibble: 4 x 4
##   usertype   gender      n  prop
##   <chr>      <chr>   <int> <dbl>
## 1 Customer   Female   1875 0.316
## 2 Customer   Male     4060 0.684
## 3 Subscriber Female  65043 0.192
## 4 Subscriber Male   274380 0.808
```

```r
ggplot(usertype_gender_prop, aes(x = usertype, y = prop, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous(labels = scales::percent) +  # Format y-axis labels as percentages
  labs(title = "Proportion of User Type by Gender",
       x = "User Type",
       y = "Proportion") +
  theme_minimal()
```



Interpretation:

Both count and proportion data show that male subscribers have considerably higher bike use **frequency** than female subscribers. Proportion graph shows that among customers, females use at around %32, while males use bikes at around %68. Among subscribers, females' frequenct of bike use is %19 while males' is %82.

Note that unlike **tripduration by gender** in **Section 1.1** where trip duration does not vary too much between males and females in user categories, the frequency of bike use by user type differ drastically between males and females. Thus, trip duration does not seem to be a key metric, unlike frequency.

3. Weekdays First, I need to find the day of bike use.

```
trips_2019$weekday <- weekdays(trips_2019$start_time)
```

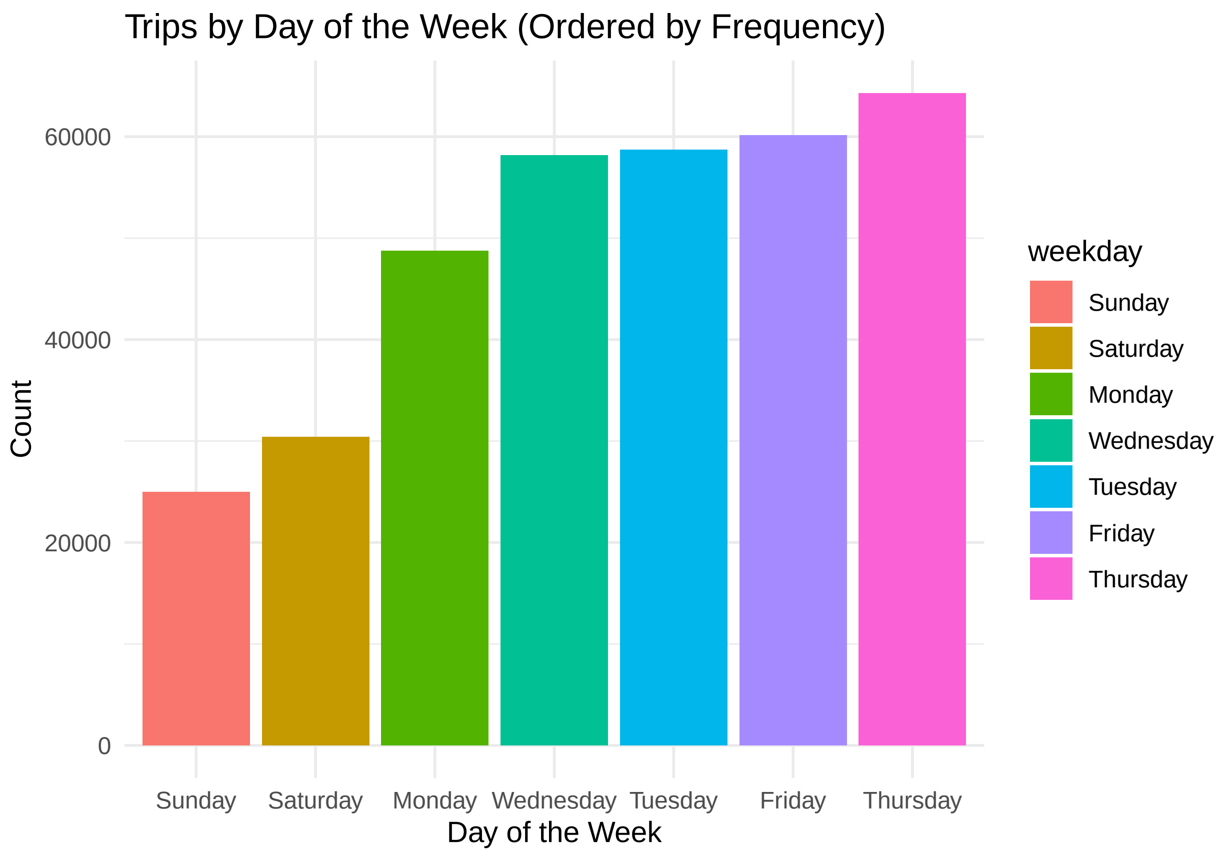
```
table(trips_2019$weekday)
```

```
##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##     60118     48729     30384     24964     64303     58711     58149
```

Calculate weekday counts and reorder weekdays by frequency

```
trips_weekday_summary <- trips_2019 %>%
  count(weekday) %>%
  arrange(n) %>%
  mutate(weekday = factor(weekday, levels = weekday))
```

```r
ggplot(data = trips_weekday_summary, aes(x = weekday, y = n, fill = weekday)) +
  geom_bar(stat = "identity") +
  labs(x = "Day of the Week", y = "Count", title = "Trips by Day of the Week (Ordered by Frequency)") +
  theme_minimal()
```



Interpretation:

Overall, weekday bike use is significantly greater than weekend use.

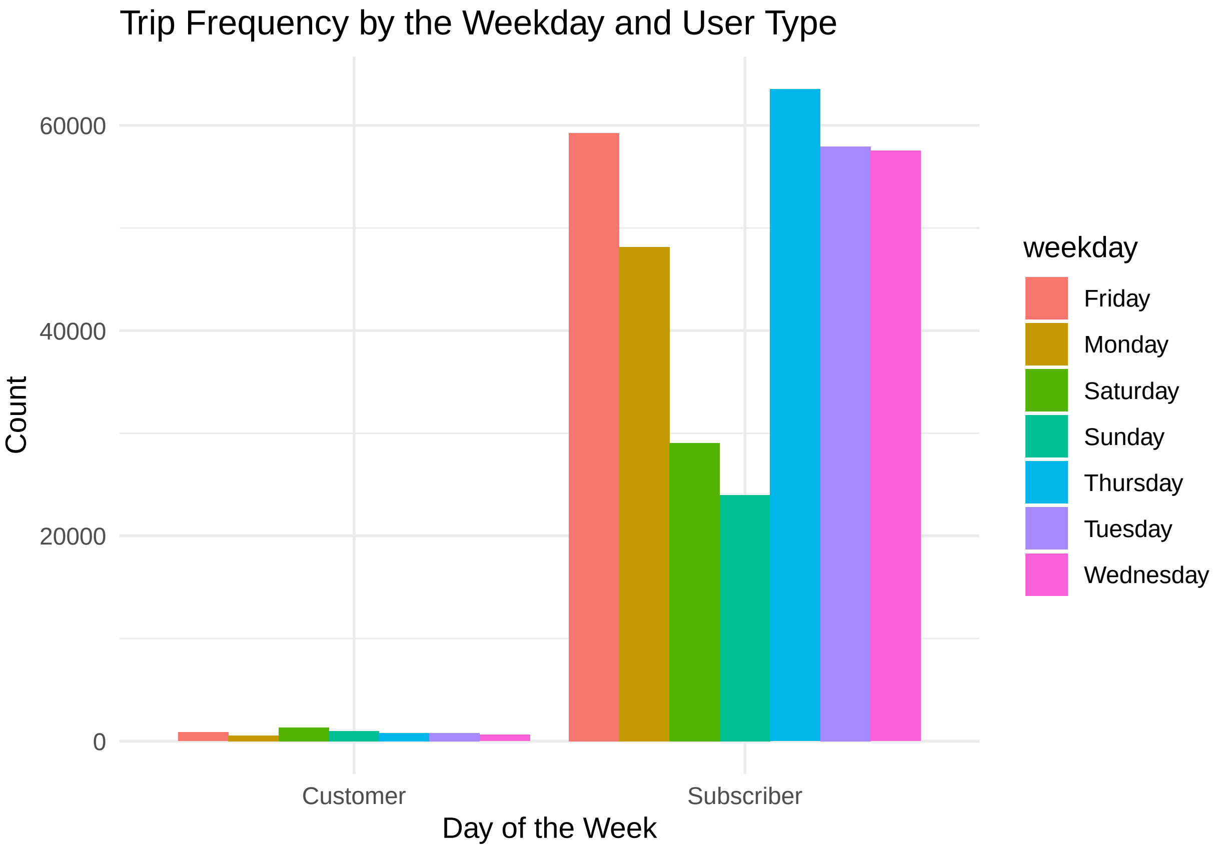
3.1 Weekday by Usertype

Frequency of weekday

```r
weekdays_by_users<-trips_2019%>%
  count(usertype, weekday)
weekdays_by_users
```

```
##      usertype   weekday     n
## 1    Customer    Friday   874
## 2    Customer    Monday   568
## 3    Customer  Saturday  1322
## 4    Customer    Sunday   976
## 5    Customer  Thursday   784
## 6    Customer   Tuesday   783
## 7    Customer Wednesday   628
## 8  Subscriber    Friday 59244
## 9  Subscriber    Monday 48161
## 10 Subscriber  Saturday 29062
## 11 Subscriber    Sunday 23988
## 12 Subscriber  Thursday 63519
## 13 Subscriber   Tuesday 57928
## 14 Subscriber Wednesday 57521
```

```
ggplot(data=weekdays_by_users)+
  geom_bar(stat="identity", position="dodge", mapping = aes(x=usertype, y=n, fill=weekday))+
   labs(x = "Day of the Week", y = "Count", title = "Trip Frequency by the Weekday and User Type") +
  theme_minimal()
```



Interpretation:

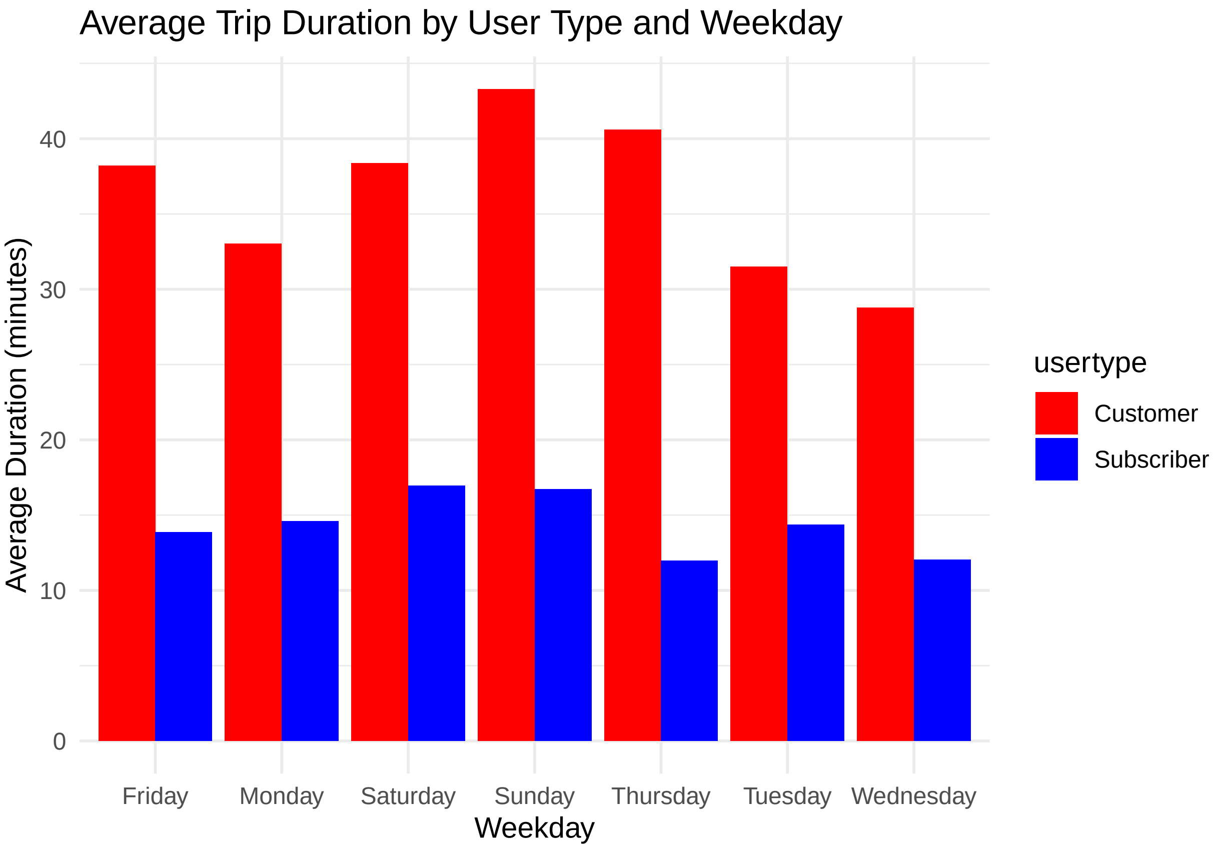
Customers and casual bikers have a unique pattern regarding the day of the week that they use bikes. While members use bikes on weekends more, casual riders prefer them on weekdays more.

```
trip_duration_by_usertype_weekday <- trips_2019 %>%
  group_by(usertype, weekday) %>%
  summarize(avg_duration = mean(tripduration, na.rm = TRUE), .groups = 'drop')
trip_duration_by_usertype_weekday
```

3.2 Trip duration by usertype and weekday

```
## # A tibble: 14 x 3
##    usertype   weekday   avg_duration
##    <chr>      <chr>            <dbl>
##  1 Customer   Friday            38.2
##  2 Customer   Monday            33.0
##  3 Customer   Saturday          38.4
##  4 Customer   Sunday            43.3
##  5 Customer   Thursday          40.6
##  6 Customer   Tuesday           31.5
##  7 Customer   Wednesday         28.8
##  8 Subscriber Friday            13.9
##  9 Subscriber Monday            14.6
## 10 Subscriber Saturday          17.0
## 11 Subscriber Sunday            16.7
## 12 Subscriber Thursday          12.0
## 13 Subscriber Tuesday           14.4
## 14 Subscriber Wednesday         12.0
```

```
ggplot(trip_duration_by_usertype_weekday, aes(x = weekday, y = avg_duration, fill = usertype)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Trip Duration by User Type and Weekday",
       x = "Weekday",
       y = "Average Duration (minutes)") +
  theme_minimal() +
  scale_fill_manual(values = c("Subscriber" = "blue", "Customer" = "red"))
```



Interpretation:

This graph shows that trip duration on weekends in each group is in peak: Both members and casual riders tend to have longer bike usage time in weekends than weekdays.

On the other hand, this conclusion contrasts with the finding in previous section **3.1 Weekday by Usertype** where casual riders, or subscribers tend not use bikes during the weekends. Their bike use frequency in weekends is less as compared to weekdays.

The contrast might be due to the fact that casual riders prefer bikes in short commutes like work, school during the week, while they use bikes for leisure time activity in weekends, hence their trip duration increases accordingly during the weekend.

```
summary(trips_2019$birthyear)
```

4. Age

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA’s 
##    1900    1975    1985    1982    1990    2003       1 
```

There are some extreme values like 1900 as birth year, so I will filter out them first. I will include birthyear equal or greater than 1940.

```
trips_2019<-trips_2019%>%
  filter(birthyear>=1940)
```

Calculate the age using birth year.

```
trips_2019<-trips_2019%>%
  mutate(age=2019-birthyear)
```

```
summary(trips_2019$age)
```

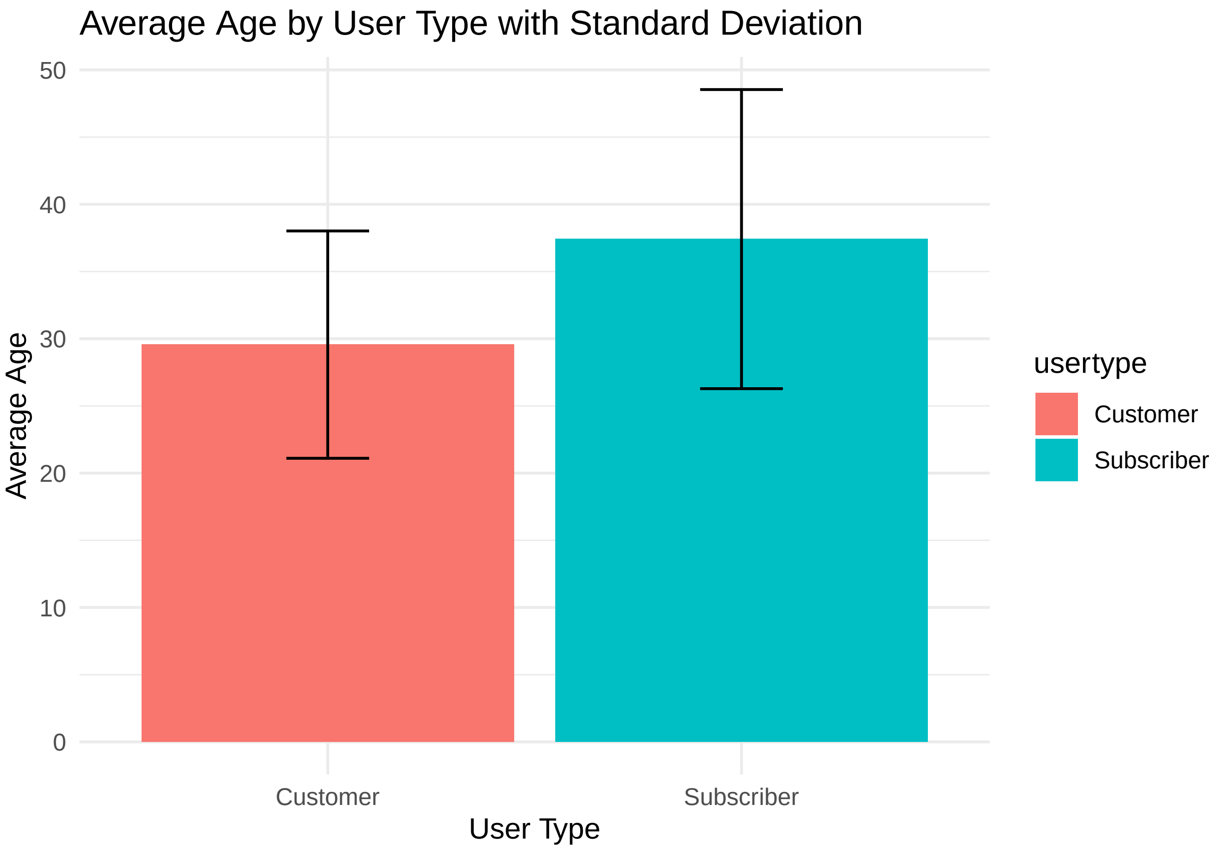
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   16.00   29.00   34.00   37.27   44.00   79.00 
```

```
summary_age_usertype <- trips_2019 %>%
  group_by(usertype) %>%
  summarize(mean_age = mean(age, na.rm = TRUE),
            sd_age = sd(age, na.rm = TRUE))

summary_age_usertype
```

4.1 User type by Age

```
## # A tibble: 2 x 3
##   usertype   mean_age sd_age
##   <chr>         <dbl>  <dbl>
## 1 Customer       29.6   8.46
## 2 Subscriber     37.4  11.1
```

```
ggplot(summary_age_usertype, aes(x = usertype, y = mean_age, fill = usertype)) +
  geom_bar(stat = "identity") +
  geom_errorbar(aes(ymin = mean_age - sd_age, ymax = mean_age + sd_age), width = 0.2) +
  labs(title = "Average Age by User Type with Standard Deviation",
       x = "User Type",
       y = "Average Age") +
  theme_minimal()
```



Interpretation:

Customers (mean age=29.55) are relatively younger than casual riders (mean age=37.40). Note that standard deviation is higher in Subscribers meaning there is wider dispersion/variation of data in this group.

4. Statistical Analysis

I want to find which predictors are significant in predicting if a user will be a **customer** or **subscriber**. For this, I need to build a **logistic regression** model because the outcome variable is a binomial type.

There are mainly two ways to build a (logistic) regression. First, we can start with an intercept model only, then incrementally add one predictor into the model, and evaluate the model performance to see each added variable have predictive value for the outcome. This process continues untill all predictors are fed into the model.

Another is the opposite, where we feed all predictors into the model initially and remove each independent variable at a time, and compare the model performance in each step. I will adopt the latter, which is the step-wise backward selection.

To be able to do this, dummy coding is required for **usertype**. I need to convert levels/factors of **usertype** into 0 and 1, because the outcome variable in a logistic regression should be a binomial data. **Customer** level in **usertype** will be the reference category with 0 because it is less frequent and less interesting. In other words, the **Subscriber** level in **usertype** will be 1, being compared to the reference category during cooefficient interpretation.

Before that, I need to convert **usertype** into a factor, which then can be converted into 1 and 0.

```
trips_2019$usertype <- as.factor(trips_2019$usertype)
glimpse(trips_2019)
```

```
## Rows: 345,160
## Columns: 14
## $ trip_id           <int> 21742443, 21742444, 21742445, 21742446, 21742447, 21~
## $ start_time        <dttm> 2019-01-01 00:04:37, 2019-01-01 00:08:13, 2019-01-0~
## $ end_time          <dttm> 2019-01-01 00:11:07, 2019-01-01 00:15:34, 2019-01-0~
## $ bikeid            <int> 2167, 4386, 1524, 252, 1170, 2437, 2708, 2796, 6205,~
## $ tripduration      <dbl> 6.500000, 7.350000, 13.816667, 29.716667, 6.066667, ~
## $ from_station_id   <int> 199, 44, 15, 123, 173, 98, 98, 211, 150, 268, 299, 2~
## $ from_station_name <chr> "Wabash Ave & Grand Ave", "State St & Randolph St", ~
## $ to_station_id     <int> 84, 624, 644, 176, 35, 49, 49, 142, 148, 141, 295, 4~
## $ to_station_name   <chr> "Milwaukee Ave & Grand Ave", "Dearborn St & Van Bure~
## $ usertype          <fct> Subscriber, Subscriber, Subscriber, Subscriber, Subs~
## $ gender            <chr> "Male", "Female", "Female", "Male", "Male", "Female"~
## $ birthyear         <int> 1989, 1990, 1994, 1993, 1994, 1983, 1984, 1990, 1995~
## $ weekday           <chr> "Tuesday", "Tuesday", "Tuesday", "Tuesday", "Tuesday~
## $ age               <dbl> 30, 29, 25, 26, 25, 36, 35, 29, 24, 23, 25, 25, 33, ~
```

```
trips_2019$usertype_new <- ifelse(trips_2019$usertype == "Subscriber", 1, 0)

table(trips_2019$usertype_new)
```

```
## 
##      0      1 
##   5934 339226
```

```
table(trips_2019$usertype) # checking if dummy coding is accurate!
```

```
## 
##   Customer Subscriber 
##       5934     339226
```

```
backward_model<-glm(usertype_new~tripduration+weekday+gender+age, data=trips_2019, family = binomial())
```

```
stepwise_backward_model<-step(backward_model,direction = "backward")
```

```
## Start:  AIC=54435.78
## usertype_new ~ tripduration + weekday + gender + age
## 
##                Df Deviance   AIC
## <none>               54416 54436
## - tripduration  1    54423 54441
## - gender        1    54706 54724
## - weekday       6    55658 55666
## - age           1    57737 57755
```

Interpreation:

"none" is the baseline model with all predictors being fed into the model. I will compare the Deviance and AIC of "none" with the removal of each predictor. **The least the Deviance and AIC is, the better the model is.** Thus, if removing any predictor leads to an increase in Deviance and AIC, the model gets worse, which indicates that it is a significant predictor on **user type**.

The model output shows that all variables are important in predicting **usertype** because removal of each yields an increase in Deviance and AIC. The **age** being the most significant predictor while **trip duration** being the least significant variable.

```
summary(stepwise_backward_model)
```

```
## 
## Call:
```

```
## glm(formula = usertype_new ~ tripduration + weekday + gender +
##     age, family = binomial(), data = trips_2019)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       5.156e-01  7.439e-02   6.931 4.19e-12 ***
## tripduration     -4.300e-05  1.253e-05  -3.433 0.000598 ***
## weekdayMonday     2.412e-01  5.450e-02   4.426 9.62e-06 ***
## weekdaySaturday  -8.927e-01  4.465e-02 -19.993  < 2e-16 ***
## weekdaySunday    -7.678e-01  4.766e-02 -16.110  < 2e-16 ***
## weekdayThursday   1.903e-01  4.976e-02   3.825 0.000131 ***
## weekdayTuesday    9.029e-02  4.980e-02   1.813 0.069823 .
## weekdayWednesday  3.063e-01  5.287e-02   5.793 6.92e-09 ***
## genderMale        5.040e-01  2.865e-02  17.590  < 2e-16 ***
## age               9.943e-02  2.106e-03  47.213  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 59989  on 345159  degrees of freedom
## Residual deviance: 54416  on 345150  degrees of freedom
## AIC: 54436
##
## Number of Fisher Scoring iterations: 8
```

Interpretation

Intercept is the odds of being a subscriber, or casual rider when all predictors are in zero value. Positive cooefficients indicate the increase in being a subscriber, while negative cooefficient signals decrease in being a subscriber.

All predictors except Tuesday is significant. Positive predictors, meaning increase in likelihood/log-odds of being a subscriber in each unit increase, are weekdayMonday, weekdayThursday, weekdayWednesday, genderMale, and age.

Negative predictors, meaning decrease in likelihood/log-odds of being a subscriber in each unit decrease, are tripduration, weekdaySaturday, and weekdaySunday.

Possible Campaigns

A. Monday, Thursday, Wednesday, male, and age are positively associated with being a "Subscriber."

  i. Subscribers use bikes on weekdays. Offeringw weekday biking perks for customers may can turn subscribers into customers.

  ii. Subscribers are mostly males. Thus, creating offers specifically for male users can be considered. Offering exclusive bike features for male customers might be plausible.

  iii. As the age increases, they tend to be subscribers. Thus, there might be a senior's discount for people who are over certain age, say 40 years old, when they become customers.

B. Tripduration, Saturday, and Sunday are negatively associated with being a "Subscriber."

  i. Subscribers have lesser trip duration. Thus, offering frequent bike benefits rather than trip duration for customers may convince subscribers to become customers. Or, encouraging longer trip duration by deals and rewards when subscribers become customers can be considered.

  ii. Subscribers prefer bikes on weekends less. Thus, one campaign might be to offer free/discounted rates on weekday use for customers if they prefer bikes on the weekends. This may help with the engagement of them all week and becoming a customer. The campaign might goes as: “Ride three weekdays, get one weekend deal, and ride five weekdays and get two weekend deal.”