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

Philip Johnson

2023-07-30

Table of content

Introduction

This is my report for the capstone project after the completion of my 5 months Google Data Analytics Professional Certificate on Coursera.

I work with a fictional bike-share company called Cyclistic. The objective is to design marketing strategies aimed at converting casual riders into annual members. I was assigned to spot differences in how annual members and casual riders use Cyclistic bikes differently.

I used Cyclistic historical trip data to determine trends or relationships, analyse data, aggregate data and format data correctly. After then, I develop an action plan based on those findings.

Cyclistic is a bike-share company with two categories of customers; casual riders (customers who purchase single-ride or full-day passes) and annual members (Customers who purchase annual memberships). 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.

Characters and teams:

1. **Cyclistic:** A bike-share program that features more than 5,800 bicycles and 600 docking stations.
2. **Lily Moreno:** The director of marketing and my manager.
3. **Cyclistic marketing analytics team:** A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy.
4. **Cyclistic executive team:** The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

ASK

Problem Explanation Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. There are 3 questions that 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?

For this project, the director of marketing assigned me to answer the first question (How do annual members and casual riders use Cyclistic bikes differently?).

Business Task: Analyse fictional company (Cyclistic) trip data to spot differences in how annual members and casual riders use Cyclistic bikes differently.

Prepare

Data Sources Description: I will be using Cyclistic historical trip data link. It is a public data made available by Motivate International Inc. under this license. It includes Cyclistic historical trip data of customers for each month. The data is organized in folders containing CSV files of the data classified by month and year. Each record represents a trip and each trip is anonymised.

I will be using data from July 2020 to June 2021. There are 12 files with naming convention of YYYYMM-divvy-tripdata. Each CSV files have 13 columns which names are ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id, end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng and member_casual.

Data-privacy issues prohibit me from using riders' personally identifiable information. This means that I won't be able to connect pass purchases to credit card numbers to determine if casual riders live in the Cyclistic service area or if they have purchased multiple single passes.

In terms of bias and credibility, Cyclistic is a fictional company and the data is a public data, the data is reliable, original, current because it's updated, comprehensive because not missing essential information required for the analysis and was collected ethically. Employed both manual and automated processes to verify data integrity.

Process

Tools used: Excel and R I downloaded the previous 12 months of Cyclistic trip data from July 2020 to June 2021. Unzip the files, created a folder on my desktop and housed the files. Created subfolders for the .CSV file and the .XLS file so I can have a copy of the original data. I launched Excel, opened each file, and chose to Save As an Excel Workbook file. For each .XLS file, I applied the following process:

1. Changed some column names; **rideable_type** to **ride_type**, **started_at** to **start_datetime**, **ended_at** to **end_datetime**, and **membership_casual** to **customer_type**.
2. Checked for duplicate values using **(Data < remove duplicates < for all columns)**, but no duplicate values.
3. Checked for misspelled values under **bike_type**, **end_station_name**, **start_station_name** and **customer_type** using **(Review < spelling)**, no misspelled values found.

After making these updates for all the 12 files, I saved each .XLS file as a new .CSV file.

Since the datasets are large, I decided to continue my cleaning or manipulation process with **R** programming.

Load the packages needed

```r
library(plyr)
library(tidyverse)
```

```
## -- 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::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::desc()      masks plyr::desc()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(dplyr)
library(lubridate)
library(readr)
library(skimr)
library(ggplot2)
library(RColorBrewer)
library(patchwork)
```

Load data for 6 months due to **R** RAM space (from January 2021 to June 2021).

```
biketrips_01_2021 <- read_csv("C:/Users/HP/Documents/cyclistic_project/csv_files/01_2021_cyclistic_bike
```

```
## Rows: 96834 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): ride_id, bike_type, start_datetime, end_datetime, start_station_nam...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
biketrips_02_2021 <- read_csv("C:/Users/HP/Documents/cyclistic_project/csv_files/02_2021_cyclistic_bike
```

```
## Rows: 49622 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): ride_id, bike_type, start_datetime, end_datetime, start_station_nam...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```r
biketrips_03_2021 <- read_csv("C:/Users/HP/Documents/cyclistic_project/csv_files/03_2021_cyclistic_bike
```

```
## Rows: 228496 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): ride_id, bike_type, start_datetime, end_datetime, start_station_nam...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```r
biketrips_04_2021 <- read_csv("C:/Users/HP/Documents/cyclistic_project/csv_files/04_2021_cyclistic_bike
```

```
## Rows: 337230 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): ride_id, bike_type, start_datetime, end_datetime, start_station_nam...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```r
biketrips_05_2021 <- read_csv("C:/Users/HP/Documents/cyclistic_project/csv_files/05_2021_cyclistic_bike
```

```
## Rows: 531633 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): ride_id, bike_type, start_datetime, end_datetime, start_station_nam...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```r
biketrips_06_2021 <- read_csv("C:/Users/HP/Documents/cyclistic_project/csv_files/06_2021_cyclistic_bike
```

```
## Rows: 729595 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): ride_id, bike_type, start_datetime, end_datetime, start_station_nam...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

Combine all the data

```r
merged_biketrips <- bind_rows(biketrips_01_2021, biketrips_02_2021, biketrips_03_2021, biketrips_04_202
```

Check the summary and structure of the data

```r
str(merged_biketrips)
```

```
## spc_tbl_ [1,973,410 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1973410] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA4
##  $ bike_type         : chr [1:1973410] "electric_bike" "electric_bike" "electric_bike" "electric_bike
##  $ start_datetime    : chr [1:1973410] "1/23/2021 16:14" "1/27/2021 18:43" "1/21/2021 22:35" "1/7/202
##  $ end_datetime      : chr [1:1973410] "1/23/2021 16:24" "1/27/2021 18:47" "1/21/2021 22:37" "1/7/202
##  $ start_station_name: chr [1:1973410] "California Ave & Cortez St" "California Ave & Cortez St" "Cal
##  $ start_station_id  : chr [1:1973410] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:1973410] NA NA NA NA ...
##  $ end_station_id    : chr [1:1973410] NA NA NA NA ...
##  $ start_lat         : num [1:1973410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:1973410] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:1973410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:1973410] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ customer_type     : chr [1:1973410] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   bike_type = col_character(),
##   ..   start_datetime = col_character(),
##   ..   end_datetime = col_character(),
##   ..   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(),
##   ..   customer_type = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
summary(merged_biketrips)
```

```
##    ride_id           bike_type         start_datetime     end_datetime      
##  Length:1973410     Length:1973410     Length:1973410     Length:1973410    
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##  start_station_name start_station_id   end_station_name   end_station_id    
##  Length:1973410     Length:1973410     Length:1973410     Length:1973410    
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##
```

```
##    start_lat       start_lng         end_lat         end_lng      
##  Min.   :41.64   Min.   :-87.78   Min.   :41.51   Min.   :-88.07  
##  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   :-87.64   Mean   :41.90   Mean   :-87.64  
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63  
##  Max.   :42.07   Max.   :-87.52   Max.   :42.15   Max.   :-87.49  
##                                   NA's   :1920    NA's   :1920    
##  customer_type     
##  Length:1973410    
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
## 
```

Reveal the column names

```r
colnames(merged_biketrips)
```

```
##  [1] "ride_id"            "bike_type"          "start_datetime"    
##  [4] "end_datetime"       "start_station_name" "start_station_id"  
##  [7] "end_station_name"   "end_station_id"     "start_lat"         
## [10] "start_lng"          "end_lat"            "end_lng"           
## [13] "customer_type"
```

Continue my data cleaning and manipulation in R Change the format of start_datetime and end_datetime columns to date and time format

```r
merged_biketrips$start_datetime <- as.POSIXct(merged_biketrips$start_datetime, format = "%m/%d/%Y %H:%M
merged_biketrips$end_datetime <- as.POSIXct(merged_biketrips$end_datetime, format = "%m/%d/%Y %H:%M")
```

Check if there are any misspelled value in column customer_type

```r
unique(merged_biketrips$customer_type)
```

```
## [1] "member" "casual"
```

Create new columns; ride_length, day_of_week, hour and month

```r
cleaned_biketrips = merged_biketrips %>%
  mutate(
    ride_length = difftime(end_datetime, start_datetime, units = "mins"),
    day_of_week = wday(start_datetime, label = T, abbr = F),
    hour_start = hour(start_datetime),
    month = month(start_datetime, label = T, abbr = F),
      )
```

Change ride_length to numeric format

```
cleaned_biketrips$ride_length <-
  as.numeric(as.character(cleaned_biketrips$ride_length))
is.numeric(cleaned_biketrips$ride_length) #returns TRUE if ride_length is numeric already
```

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

Remove bad ride_length data Ride length must be at least 1 min but not more than 24 hours(1440 minutes)

```
cleaned_biketrips = cleaned_biketrips %>%
  filter(between(ride_length, 1, 1440))
bad_data <- nrow(merged_biketrips) - nrow(cleaned_biketrips)
paste0("There are a total of ", bad_data , " bad data removed ")
```

```
## [1] "There are a total of 69062 bad data removed "
```

Check the cleaned data

```
glimpse(cleaned_biketrips)
```

```
## Rows: 1,904,348
## Columns: 17
## $ ride_id            <chr> "E19E6F1B8D4C42ED", "DC88F20C2C55F27F", "EC45C94683~
## $ bike_type          <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ start_datetime     <dttm> 2021-01-23 16:14:00, 2021-01-27 18:43:00, 2021-01-~
## $ end_datetime       <dttm> 2021-01-23 16:24:00, 2021-01-27 18:47:00, 2021-01-~
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor~
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660~
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, "Wood St & Augusta ~
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, "657", "13258", "65~
## $ start_lat          <dbl> 41.90034, 41.90033, 41.90031, 41.90040, 41.90041, 4~
## $ start_lng          <dbl> -87.69674, -87.69671, -87.69664, -87.69666, -87.696~
## $ end_lat            <dbl> 41.89000, 41.90000, 41.90000, 41.92000, 41.94000, 4~
## $ end_lng            <dbl> -87.72000, -87.69000, -87.70000, -87.69000, -87.710~
## $ customer_type      <chr> "member", "member", "member", "member", "casual", "~
## $ ride_length        <dbl> 10, 4, 2, 11, 53, 5, 6, 3, 7, 5, 9, 9, 11, 21, 6, 4~
## $ day_of_week        <ord> Saturday, Wednesday, Thursday, Thursday, Saturday, ~
## $ hour_start         <int> 16, 18, 22, 13, 14, 5, 15, 9, 19, 12, 15, 15, 15, 1~
## $ month              <ord> January, January, January, January, January, Januar~
```

Continue my Data Cleaning and Manipulation Documentation

4. Changed the format of **start_datetime** and **end_datetime** columns to date and time format using **as.POSIXct() function**.
5. Created new data frame called **cleaned_biketrips** to house the new columns created which are **ride_length**, **day_of_week**, **months** and **hour**.
6. Created the 4 columns using the **mutate()** to house all; **difftime()** function to create the **ride_length** column subtracting the **end_datetime** columns from **start_datetime** column, **wday()** function to calculate the day of the week that each ride started called **day_of_week** column, **month()** and **hour()** functions to calculate the month and hour that each ride started called *month* and **start_hour** respectively.
7. Changed the **ride_length** column to numeric format.
8. Removed the bad data in column **ride_length** which are less than 1min or more than 24 hours (1440 minutes).

Analyse

Performed data aggregation and calculation, identified trends & relationships, analysed data and formatted data correctly using **R** programming.

Check the max value in ride_length

```
max(cleaned_biketrips$ride_length)
```

```
## [1] 1439
```

Check the min value in ride_length

```
min(cleaned_biketrips$ride_length)
```

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

Find the overall mean of ride_length among annual members and casual riders.

```
mean_ridelength_member <-
  mean(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "member"], na.rm = TRUE)
mean_ridelength_casual <-
  mean(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "casual"], na.rm = TRUE)

paste0("Members mean ride length is ", mean_ridelength_member)
```

```
## [1] "Members mean ride length is 14.3854362141648"
```

```
paste0("Casual mean ride length is ", mean_ridelength_casual)
```

```
## [1] "Casual mean ride length is 30.5959822885869"
```

Find the overall median of ride_length of annual members and casual riders.

```
median_ride_length_member <-
  median(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "member"], na.rm = TRUE)
median_ride_length_casual <-
  median(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "casual"], na.rm = TRUE)

paste0("Members median ride length is ", median_ride_length_member)
```

```
## [1] "Members median ride length is 10"
```

```
paste0("Casual median ride length is ", median_ride_length_casual)
```

```
## [1] "Casual median ride length is 18"
```

Find the minimum and maximum ride length for both members and casual riders.

```r
min(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "member"])
```

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

```r
min(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "casual"])
```

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

```r
max(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "member"])
```

```
## [1] 1434
```

```r
max(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "casual"])
```

```
## [1] 1439
```

Find the total number of rides among members and casual riders.

```r
total_rides_customer <- cleaned_biketrips %>%
  group_by(customer_type) %>%
  dplyr::summarise(rides_number = n())
total_rides_customer
```

```
## # A tibble: 2 x 2
##   customer_type rides_number
##   <chr>                <int>
## 1 casual              857752
## 2 member             1046596
```

Find the total ride_length between members and casual riders

```r
total_ride_length_member <-
  sum(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "member"],
      na.rm = TRUE)
total_ride_length_casual <-
  sum(cleaned_biketrips$ride_length[cleaned_biketrips$customer_type == "casual"],
      na.rm = TRUE)

paste0("The combined distance traveled by annual members is ",
       total_ride_length_member)
```

```
## [1] "The combined distance traveled by annual members is 15055740"
```

```r
paste0("The combined distance traveled by casual riders is ",
       total_ride_length_casual)
```

```
## [1] "The combined distance traveled by casual riders is 26243765"
```

Find the most popular day of the week between annual members and casual riders.

```
favored_day <- cleaned_biketrips %>%
  group_by(customer_type, day_of_week) %>%
  dplyr::summarise(rides_number = n(), .groups = "drop")
favored_day
```

```
## # A tibble: 14 x 3
##    customer_type day_of_week rides_number
##    <chr>         <ord>              <int>
##  1 casual        Sunday            174736
##  2 casual        Monday             98628
##  3 casual        Tuesday            98422
##  4 casual        Wednesday          93825
##  5 casual        Thursday           83598
##  6 casual        Friday            115581
##  7 casual        Saturday          192962
##  8 member        Sunday            137648
##  9 member        Monday            145668
## 10 member        Tuesday           159441
## 11 member        Wednesday         162044
## 12 member        Thursday          139728
## 13 member        Friday            148489
## 14 member        Saturday          153578
```

```
favored_day_ridelength <- cleaned_biketrips %>%
  group_by(customer_type, day_of_week) %>%
  dplyr::summarise(mean_ridelength = mean(ride_length), .groups = "drop")
favored_day_ridelength
```

```
## # A tibble: 14 x 3
##    customer_type day_of_week mean_ridelength
##    <chr>         <ord>                 <dbl>
##  1 casual        Sunday                 35.1
##  2 casual        Monday                 30.7
##  3 casual        Tuesday                28.9
##  4 casual        Wednesday              26.7
##  5 casual        Thursday               26.3
##  6 casual        Friday                 28.0
##  7 casual        Saturday               32.6
##  8 member        Sunday                 16.4
##  9 member        Monday                 14.0
## 10 member        Tuesday                13.7
## 11 member        Wednesday              13.6
## 12 member        Thursday               13.3
## 13 member        Friday                 13.9
## 14 member        Saturday               15.9
```

Find the most popular starting hour between annual members and casual riders.

```
favored_starthour <- cleaned_biketrips %>%
  group_by(customer_type, hour_start) %>%
  dplyr::summarise(rides_number = n(), .groups = "drop")
favored_starthour
```

```
## # A tibble: 48 x 3
##    customer_type hour_start rides_number
##    <chr>              <int>        <int>
##  1 casual                 0        18392
##  2 casual                 1        12990
##  3 casual                 2         7858
##  4 casual                 3         4086
##  5 casual                 4         2931
##  6 casual                 5         3382
##  7 casual                 6         6876
##  8 casual                 7        12163
##  9 casual                 8        17911
## 10 casual                 9        22944
## # i 38 more rows
```

Find the most popular month between annual members and casual riders.

```
favored_month <- cleaned_biketrips %>%
  group_by(customer_type, month) %>%
  dplyr::summarise(rides_length = n(), .groups = "drop")
favored_month
```

```
## # A tibble: 10 x 3
##    customer_type month   rides_length
##    <chr>         <ord>          <int>
##  1 casual        January        17926
##  2 casual        March          83357
##  3 casual        April         135412
##  4 casual        May           254378
##  5 casual        June          366679
##  6 member        January        78035
##  7 member        March         143195
##  8 member        April         198582
##  9 member        May           271620
## 10 member        June          355164
```

```
favored_month_ridelength <-cleaned_biketrips %>%
  group_by(customer_type, month) %>%
  dplyr::summarise(mean_ride_length = mean(ride_length), .groups = "drop")
favored_month_ridelength
```

```
## # A tibble: 10 x 3
##    customer_type month   mean_ride_length
##    <chr>         <ord>              <dbl>
##  1 casual        January             21.6
##  2 casual        March               31.6
##  3 casual        April               31.3
##  4 casual        May                 31.9
##  5 casual        June                29.6
##  6 member        January             12.8
##  7 member        March               14.0
##  8 member        April               14.7
##  9 member        May                 14.6
## 10 member        June                14.6
```

Fine the top 15 starting stations per number of rides for both annual members and casual riders

```
# Calculate the daily average rides for each stations first
options(dplyr.summarise.inform = FALSE)
avg_rides_start_station <- cleaned_biketrips %>%
  filter(start_station_name != " ") %>%
  group_by(start_station_name, customer_type) %>%
  dplyr::summarise(rides_number = n())
avg_rides_start_station <- avg_rides_start_station[!avg_rides_start_station$start_station_name == "",]

#Then find the top 15 for both members and casual riders
top_15_stations <-
  rbind(
    avg_rides_start_station %>% filter(customer_type == "member") %>% arrange(desc(rides_number)) %>% he
    avg_rides_start_station %>% filter(customer_type == "casual") %>% arrange(desc(rides_number)) %>% he
  )

top_15_stations
```

```
## # A tibble: 30 x 3
## # Groups:   start_station_name [24]
##    start_station_name         customer_type rides_number
##    <chr>                      <chr>                <int>
##  1 Clark St & Elm St          member                9377
##  2 Wells St & Concord Ln      member                8424
##  3 Kingsbury St & Kinzie St   member                8221
##  4 Wells St & Elm St          member                7635
##  5 Dearborn St & Erie St      member                7417
##  6 Wells St & Huron St        member                7064
##  7 St. Clair St & Erie St     member                6720
##  8 Lake Shore Dr & North Blvd member                6649
##  9 Broadway & Barry Ave       member                6401
## 10 Desplaines St & Kinzie St  member                6183
## # i 20 more rows
```

Summary of Analysis

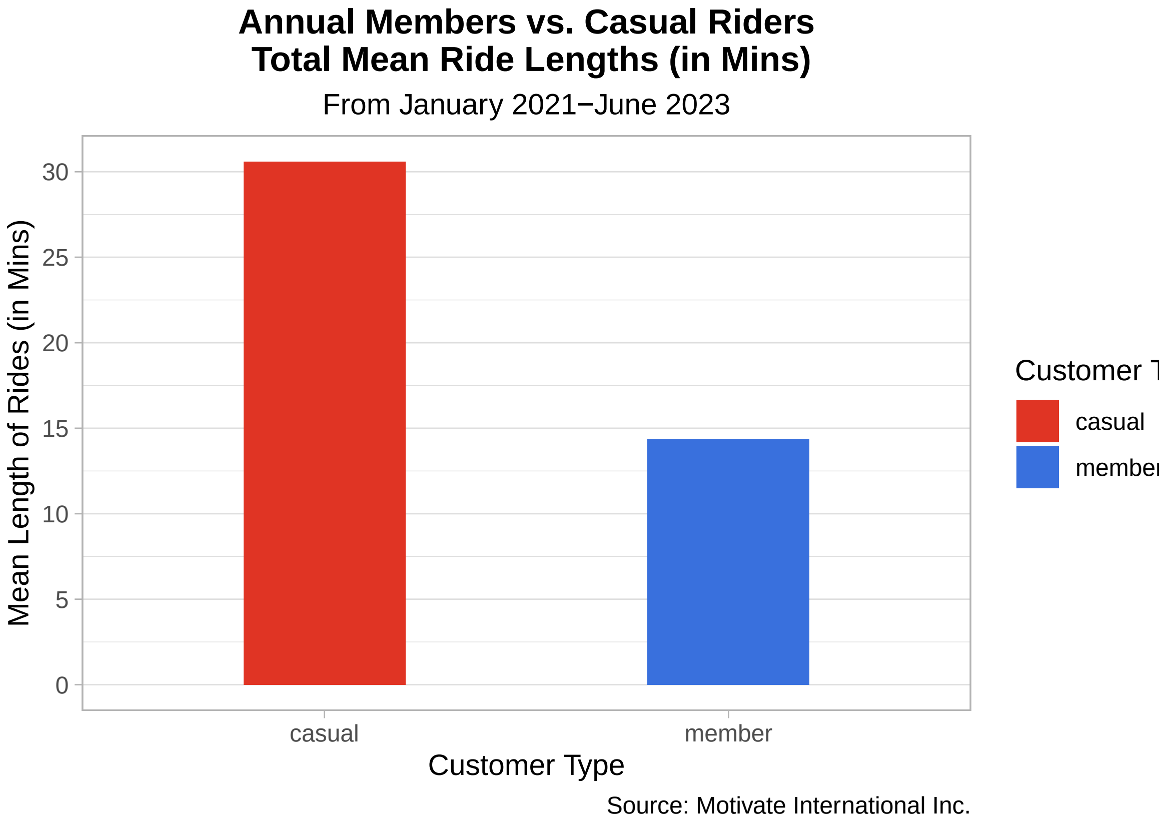
1. Casual riders have greater mean and median ride lengths than annual members.
2. Casual riders and annual member have the same minimum ride lengths, but casual riders have more maximum ride lengths than members.
3. From January 2021 to June 2021, annual members have higher rides than casual riders, but casual riders have higher total ride lengths than annual members.
4. Annual members have the most number of rides during Wednesdays and Thursdays, while casual riders have less and mostly prefer to ride bikes on Fridays, Saturdays and Sundays. Casual riders have significantly longer rides than annual members in all days of the week, with Sunday being the longest of the week.
5. Most annual members and casual riders prefer to begin their rides between 4PM and 6PM.
6. The month of June has the highest number of rides for both casual and annual members. The month of May has the longest rides for casual riders while May and June both have the longest rides for annual members with the same records.
7. The top start station for annual members are Clark St & Elm St., while the top start station for casual riders are Streeter Dr & Grand Ave.

Share

I visualised my findings using **R** programming

```
data_vis_1 <-
  cleaned_biketrips %>%
  group_by(customer_type) %>%
  dplyr::summarise(mean_ridelength = mean(ride_length),.groups = "drop") %>%
  ggplot(aes(x = customer_type,
             y = mean_ridelength,
             fill = customer_type)) +
  geom_bar(width = 0.4, position = position_dodge(width = 0.6), stat = "identity") +
  #geom_text(aes(label = x_ridelength), position = position_dodge (width=1),vjust = -0.5, size =3.5, co
  scale_fill_manual(values = c("#e03424", "#3970dd")) +
  scale_y_continuous(n.breaks = 8) +
  labs(
    title = paste(
      "Annual Members vs. Casual Riders\n Total Mean Ride Lengths (in Mins)"
    ),
    captions =
      "Source: Motivate International Inc.\n Lyft Bikes and Scooters, LLC ("Bikeshare")",
    subtitle = "From January 2021-June 2023",
    x = "Customer Type",
    y = "Mean Length of Rides (in Mins)"
  ) + labs(fill = 'Customer Type')+
  theme_light() + theme(
    plot.title = element_text(
      color = "black",
      size = 13,
      face = "bold",
      hjust = 0.5
    ),
    plot.subtitle = element_text(hjust = 0.5)
  ) +  theme(panel.grid.major.x = element_blank(),
             panel.grid.minor.x = element_blank())

data_vis_1
```

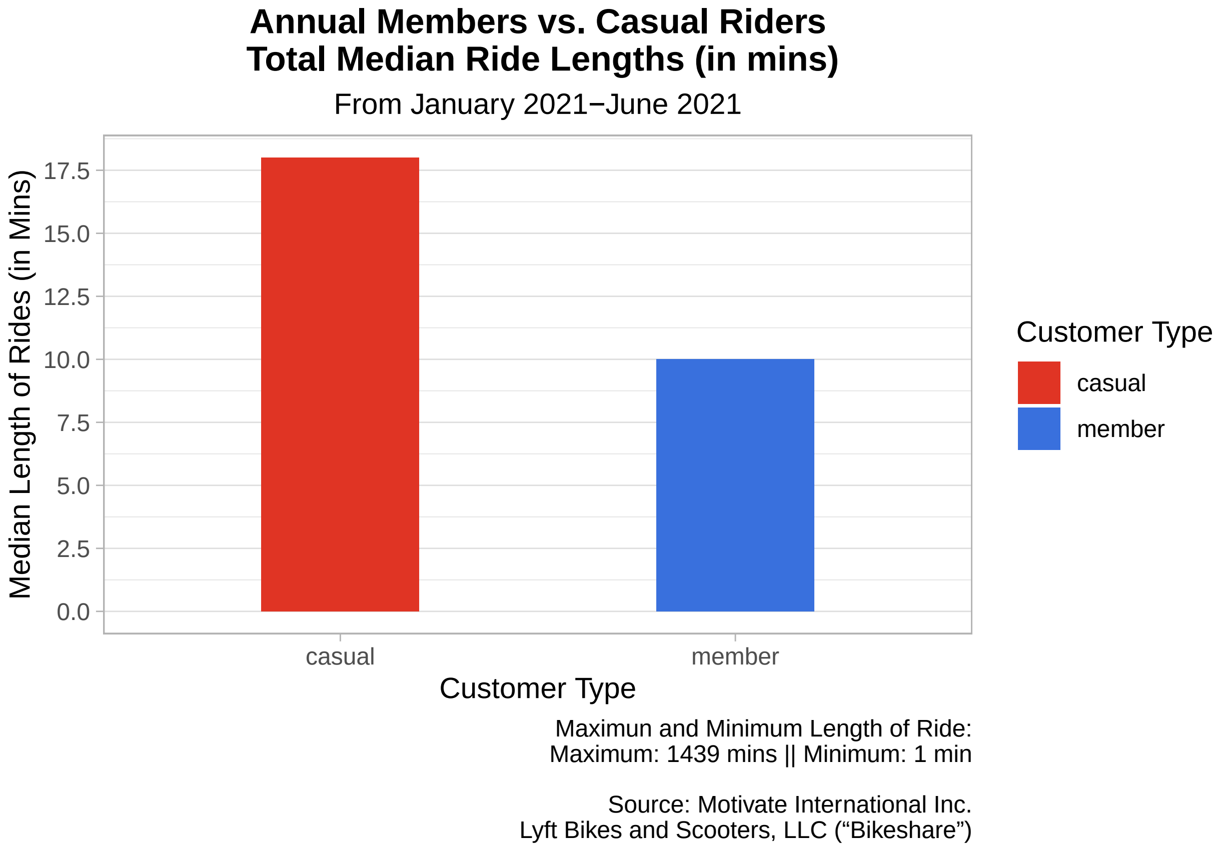


Source: Motivate International Inc.
Lyft Bikes and Scooters, LLC (“Bikeshare”)

Data Vasualisation

```r
data_vis_2 <-
  cleaned_biketrips %>%
  group_by(customer_type) %>%
  dplyr::summarise(median_ride_length = median(ride_length),.groups = "drop") %>%
  ggplot(aes(x = customer_type,
             y = median_ride_length,
             fill = customer_type)) +
  geom_bar(width = 0.4, position = position_dodge(width = 0.6), stat = "identity") +
  #geom_text(aes(label = x_ridelength), position = position_dodge (width=1),vjust = -0.5, size =3.5, co
  scale_fill_manual(values = c("#e03424", "#3970dd")) +
  scale_y_continuous(n.breaks = 8) +
  labs(
    title = paste(
      "Annual Members vs. Casual Riders\n Total Median Ride Lengths (in mins)"
    ),
    captions =
      "Maximun and Minimum Length of Ride:\nMaximum: 1439 mins || Minimum: 1 min\n
       Source: Motivate International Inc.\n Lyft Bikes and Scooters, LLC ("Bikeshare")",
    subtitle = "From January 2021-June 2021",
    x = "Customer Type",
    y = "Median Length of Rides (in Mins)"
  ) + labs(fill = 'Customer Type')+
  theme_light() + theme(
    plot.title = element_text(
      color = "black",
      size = 13,
      face = "bold",
      hjust = 0.5
    ),
    plot.subtitle = element_text(hjust = 0.5)
  ) +  theme(panel.grid.major.x = element_blank(),
             panel.grid.minor.x = element_blank())

data_vis_2
```



Annual Members vs. Casual Riders
Total Median Ride Lengths (in mins)

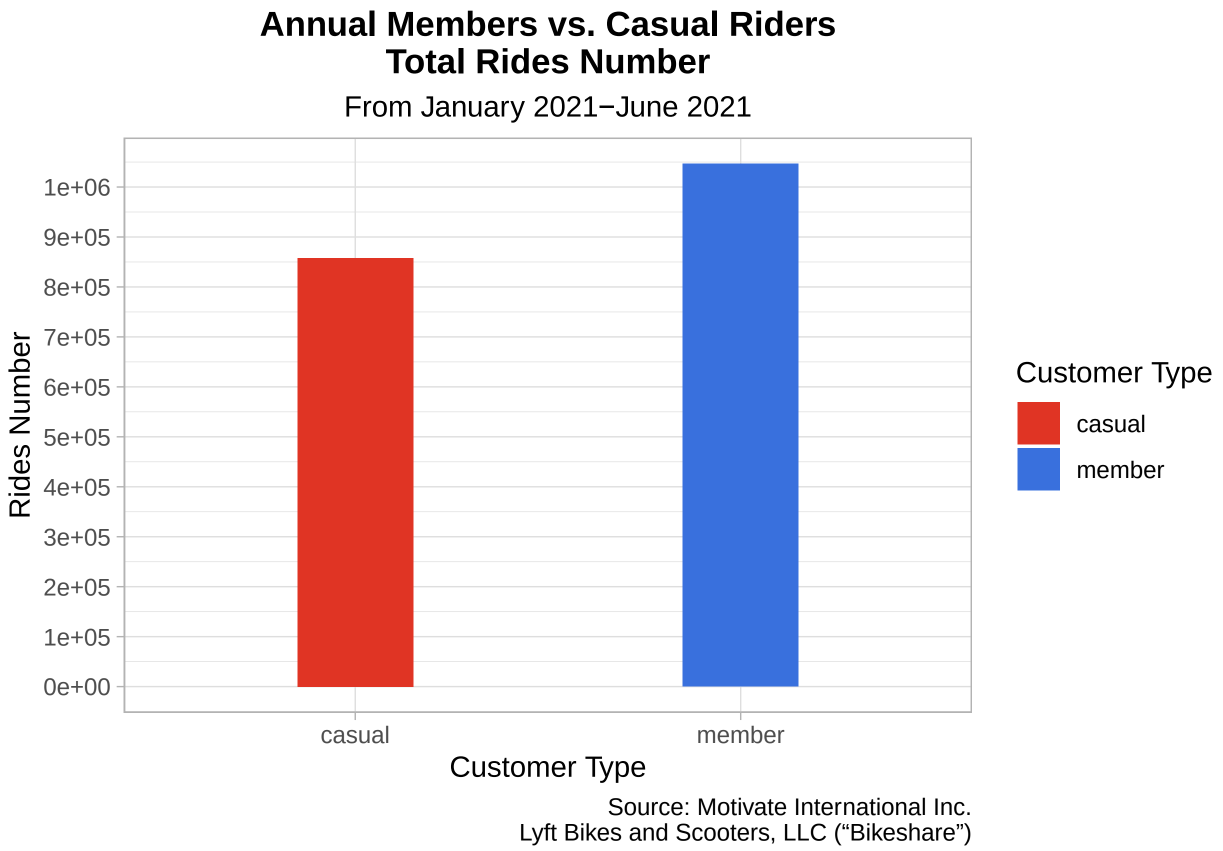
From January 2021−June 2021

Maximun and Minimum Length of Ride:
Maximum: 1439 mins || Minimum: 1 min

Source: Motivate International Inc.
Lyft Bikes and Scooters, LLC (“Bikeshare”)

```
data_vis_3 <- cleaned_biketrips %>%
  group_by(customer_type) %>%
  dplyr::summarise(rides_number = n(), .groups = "drop") %>%
  ggplot(aes(x = customer_type, y = rides_number, fill = customer_type)) +
  geom_col(width = 0.3, position = position_dodge(width = 0.6)) +
  scale_fill_manual(values = c("#e03424", "#3970dd")) +
  scale_y_continuous(n.breaks = 15) +
  labs(
    title = paste(
      "Annual Members vs. Casual Riders",
      sep = "\n",
      "Total Rides Number"
    ),
    caption = paste(
      "Source: Motivate International Inc.",
      sep = "\n",
      "Lyft Bikes and Scooters, LLC ("Bikeshare")"
    ),
    subtitle = "From January 2021-June 2021",
    x = "Customer Type",
    y = "Rides Number"
  ) + labs(fill = 'Customer Type')+
  theme_light() + theme(
    plot.title = element_text(
      color = "black",
      size = 13,
      face = "bold",
      hjust = 0.5
    ),
    plot.subtitle = element_text(hjust = 0.5)
  )

data_vis_3
```



Annual Members vs. Casual Riders
Total Rides Number

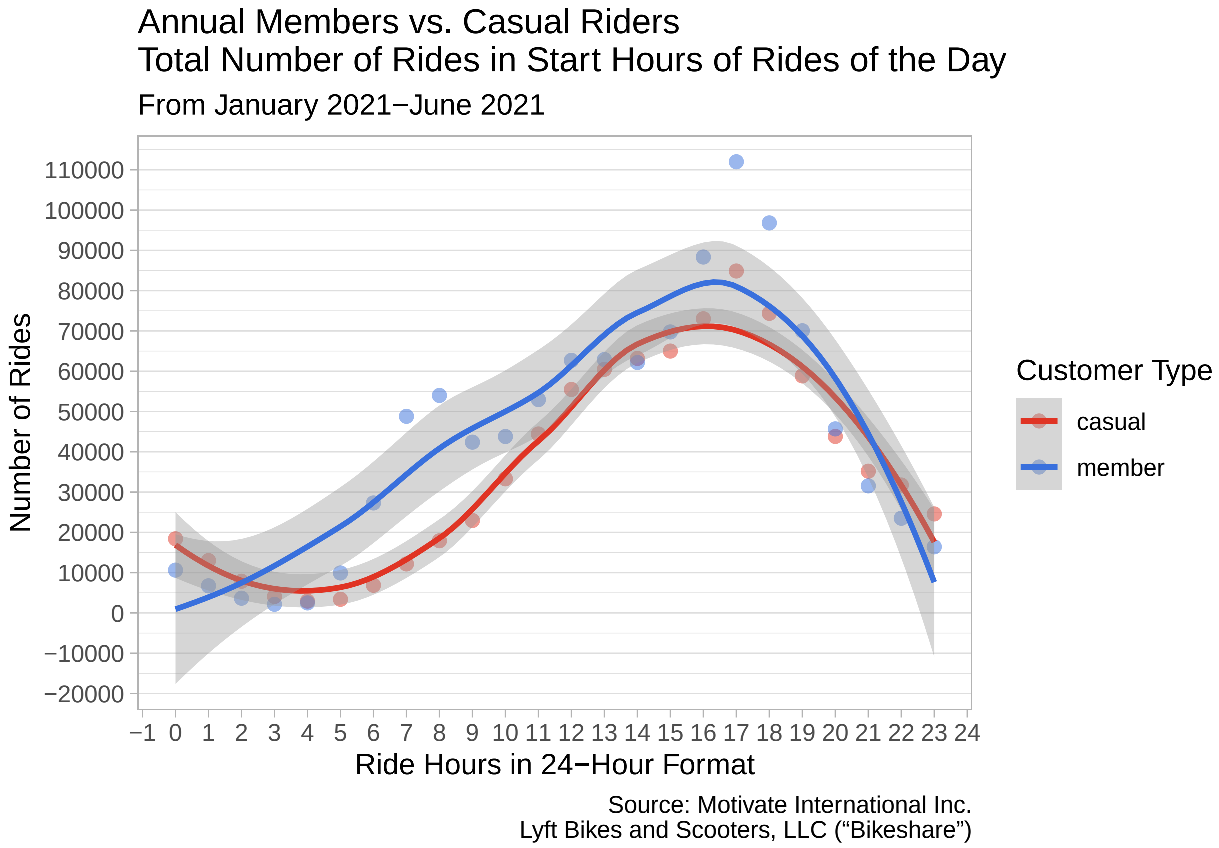
From January 2021−June 2021

Source: Motivate International Inc.
Lyft Bikes and Scooters, LLC ("Bikeshare")

```
data_vis_4 <- cleaned_biketrips %>%
  group_by(customer_type, hour_start) %>%
  dplyr::summarise(rides_number = n(), .groups = "drop") %>%
  ggplot(aes(x = hour_start, y = rides_number, col = customer_type)) + geom_point (alpha = 0.5, size =
  scale_colour_manual(name = "Customer Type",
                      values = c("casual" = "#e03424", "member" = "#3970dd")) +
  scale_y_continuous(n.breaks = 12) +
  scale_x_continuous(n.breaks = 24) +
  labs(
    title = paste(
      "Annual Members vs. Casual Riders",
      sep = "\n",
      "Total Number of Rides in Start Hours of Rides of the Day"
    ),
    caption = paste(
      "Source: Motivate International Inc.",
      sep = "\n",
      "Lyft Bikes and Scooters, LLC ("Bikeshare")"
    ),
    subtitle = "From January 2021-June 2021",
    x = "Ride Hours in 24-Hour Format",
    y = "Number of Rides"
  ) +
  geom_smooth() +
  theme_light() + theme(panel.grid.major.x = element_blank(),
                        panel.grid.minor.x = element_blank())

data_vis_4
```

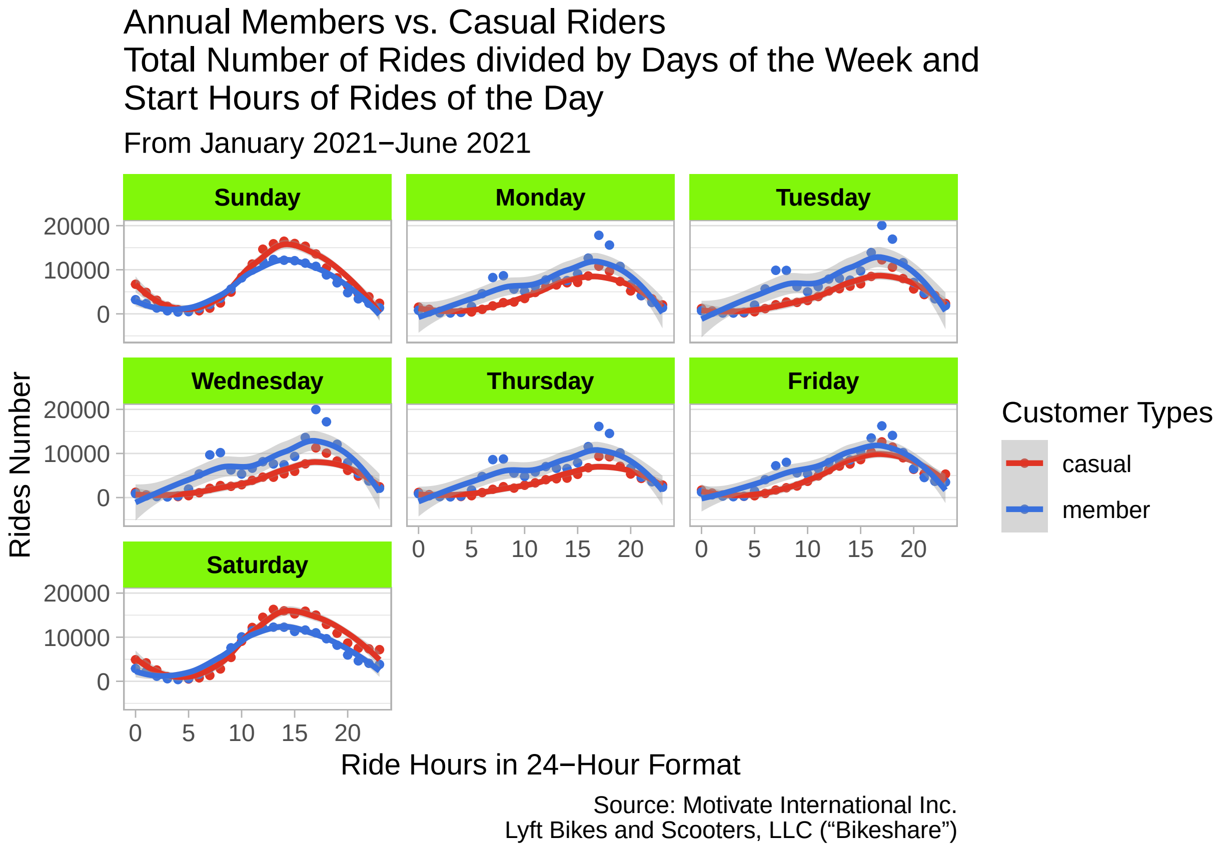
```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```



```r
data_vis_5 <- cleaned_biketrips %>%
  group_by(customer_type, hour_start, day_of_week) %>%
  dplyr::summarise(rides_number = n(), .groups = "drop") %>%
  ggplot(aes(x = hour_start, y = rides_number, col = customer_type)) + geom_point (size = 1) +
  scale_colour_manual(name = "Customer Types",
                      values = c("casual" = "#e03424", "member" = "#3970dd")) +
  scale_y_continuous(n.breaks = 4) +
  scale_x_continuous(n.breaks = 7) +
  labs(
    title = paste(
      "Annual Members vs. Casual Riders",
      sep = "\n",
      "Total Number of Rides divided by Days of the Week and\nStart Hours of Rides of the Day"
    ),
    caption = paste(
      "Source: Motivate International Inc.",
      sep = "\n",
      "Lyft Bikes and Scooters, LLC ("Bikeshare")"
    ),
    subtitle = "From January 2021-June 2021",
    x = "Ride Hours in 24-Hour Format",
    y = "Rides Number"
  ) + facet_wrap(.~day_of_week) +
  geom_smooth() + theme_light()+ theme(panel.grid.major.x = element_blank(),
                                       panel.grid.minor.x = element_blank()) +
  theme(strip.background = element_rect(fill = c("#81f70a"))) +
  theme(strip.text = element_text(colour = 'black', face = "bold"))

data_vis_5
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula = ’y ~ x’
```


Annual Members vs. Casual Riders
Total Number of Rides divided by Days of the Week and
Start Hours of Rides of the Day

From January 2021−June 2021

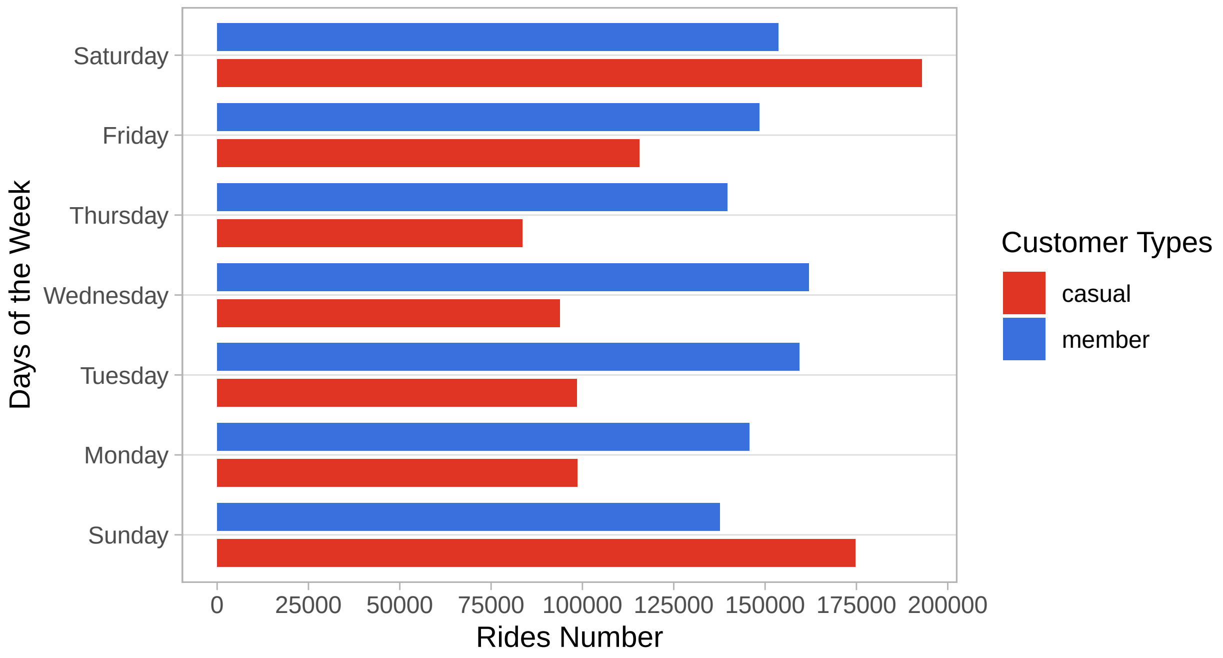
Source: Motivate International Inc.
Lyft Bikes and Scooters, LLC (“Bikeshare”)

```
data_vis_6 <- cleaned_biketrips %>%
  group_by(customer_type, day_of_week) %>%
  dplyr::summarise(rides_number = n(), .groups = "drop") %>%
  ggplot(aes(x = day_of_week,
             y = rides_number,
             fill = customer_type)) +
  geom_bar(width = 0.7, position = position_dodge(width = 0.9), stat = "identity") + coord_flip() +
  scale_fill_manual(values = c("#e03424", "#3970dd")) +
  scale_y_continuous(n.breaks = 10) +
  labs(
    title = paste(
      "Annual Members vs. Casual Riders\n Total Number of Rides per Days of the Week"
    ),
    captions =
      "Source: Motivate International Inc.\nLyft Bikes and Scooters, LLC ("Bikeshare")",
    subtitle = "From January 2021-June 2021",
    x = "Days of the Week",
    y = "Rides Number"
  ) +  labs(fill = 'Customer Types')+
  theme_light() + theme(
    plot.title = element_text(
      color = "black",
      size = 13,
      face = "bold",
      hjust = 0.5
    ),
    plot.subtitle = element_text(hjust = 0.5)
  ) +  theme(panel.grid.major.x = element_blank(),
             panel.grid.minor.x = element_blank()) +
  theme(strip.background = element_rect(fill = c("#81f70a"))) +
  theme(strip.text = element_text(colour = 'black', face = "bold"))

data_vis_6
```

Annual Members vs. Casual Riders
Total Number of Rides per Days of the Week

From January 2021−June 2021



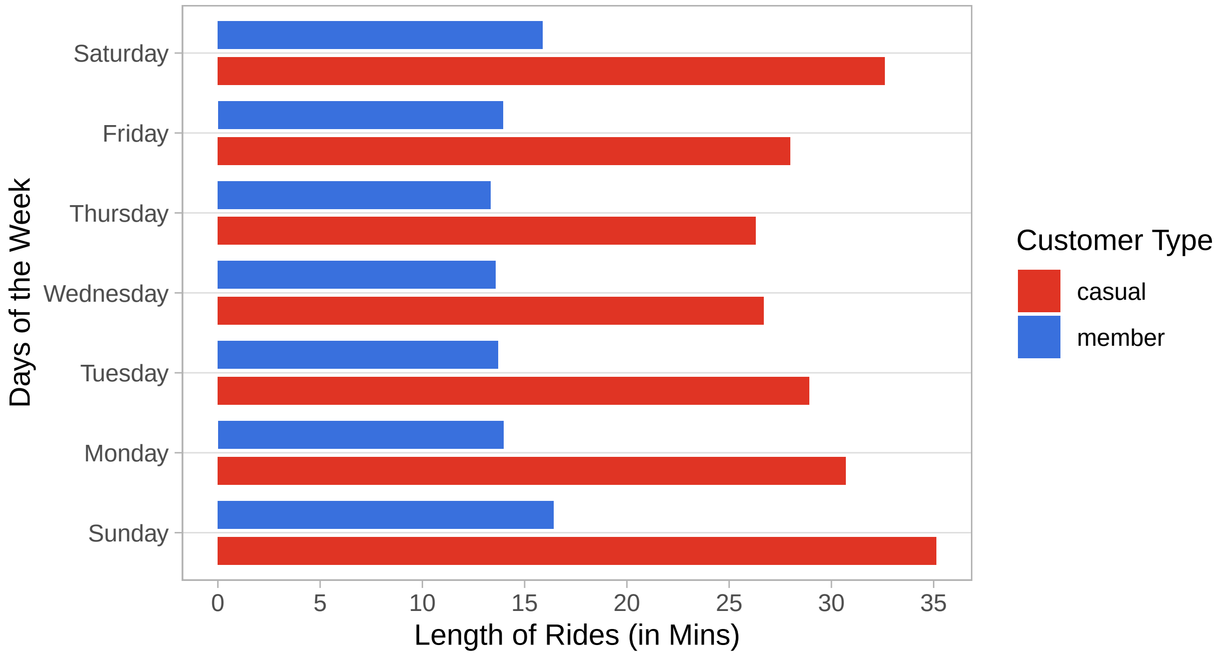
Source: Motivate International Inc.
Lyft Bikes and Scooters, LLC (“Bikeshare”)

```
data_vis_7 <- cleaned_biketrips %>%
  group_by(customer_type, day_of_week) %>%
  dplyr::summarise(mean_ride_length = mean(ride_length), .groups = "drop") %>%
  ggplot(aes(x = day_of_week,
             y = mean_ride_length,
             fill = customer_type)) +
  geom_bar(width = 0.7, position = position_dodge(width = 0.9), stat = "identity") + coord_flip() +
  scale_fill_manual(values = c("#e03424", "#3970dd")) +
  scale_y_continuous(n.breaks = 11) +
  labs(
    title = paste(
      "Annual Members vs. Casual Riders\n Average Length of Rides (in Mins) per Days of the Week"
    ),
    captions =
      "Source: Motivate International Inc.\nLyft Bikes and Scooters, LLC ("Bikeshare")",
    subtitle = "From January 2021-June 2021",
    x = "Days of the Week",
    y = "Length of Rides (in Mins)"
  ) +  labs(fill = 'Customer Type')+
  theme_light() + theme(
    plot.title = element_text(
      color = "black",
      size = 13,
      face = "bold",
      hjust = 0.5
    ),
    plot.subtitle = element_text(hjust = 0.5)
  ) +  theme(panel.grid.major.x = element_blank(),
             panel.grid.minor.x = element_blank()) +
  theme(strip.background = element_rect(fill = c("#81f70a"))) +
  theme(strip.text = element_text(colour = 'black', face = "bold"))

data_vis_7
```

Annual Members vs. Casual Riders
Average Length of Rides (in Mins) per Days of the Week

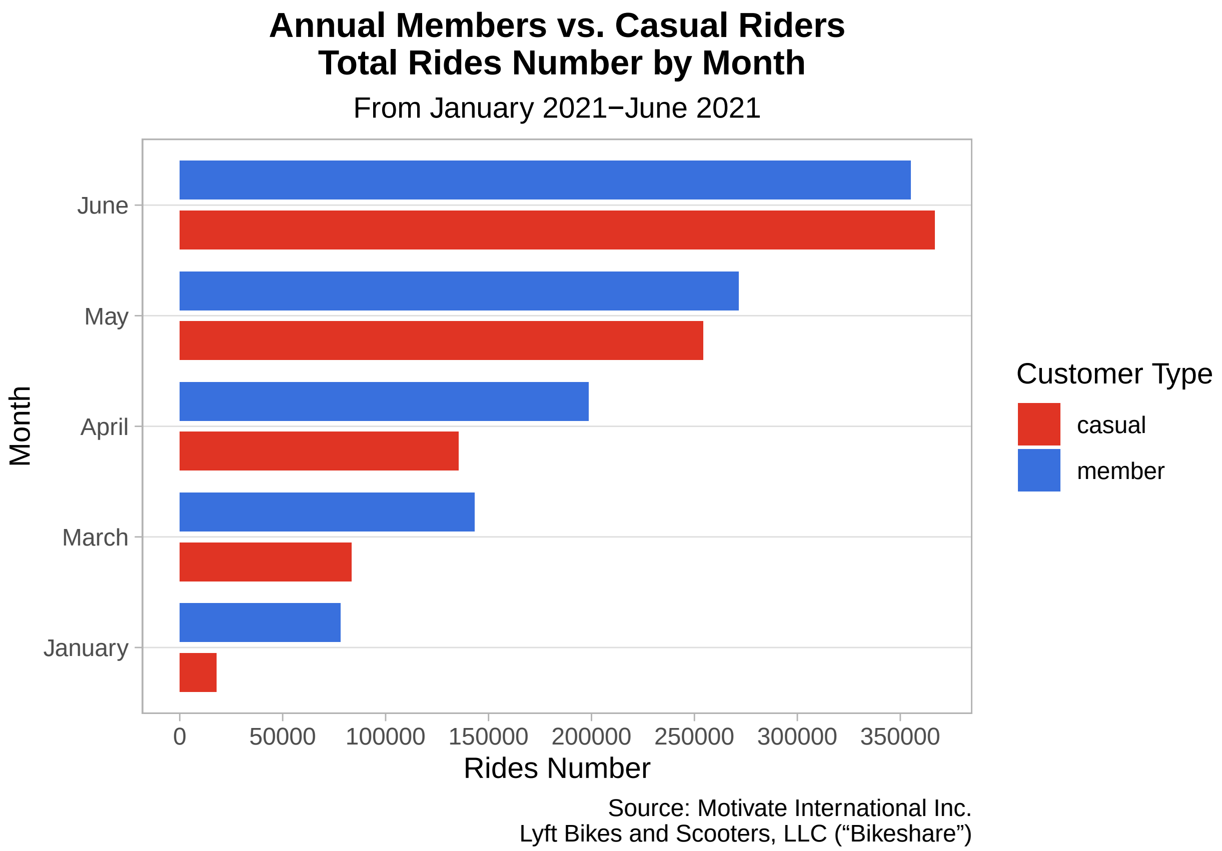
From January 2021−June 2021



Source: Motivate International Inc.
Lyft Bikes and Scooters, LLC (“Bikeshare”)

```
data_vis_8 <- cleaned_biketrips %>%
  group_by(customer_type, month) %>%
  dplyr::summarise(rides_number = n(), .groups = "drop") %>%
  ggplot(aes(x = month,
             y = rides_number,
             fill = customer_type)) +
  geom_bar(width = 0.7, position = position_dodge(width = 0.9), stat = "identity") + coord_flip() +
  scale_fill_manual(values = c("#e03424", "#3970dd")) +
  scale_y_continuous(n.breaks = 10) +
  labs(
    title = paste(
      "Annual Members vs. Casual Riders\n Total Rides Number by Month"
    ),
    captions =
      "Source: Motivate International Inc.\nLyft Bikes and Scooters, LLC (\"Bikeshare\")",
    subtitle = "From January 2021-June 2021",
    x = "Month",
    y = "Rides Number"
  ) +  labs(fill = 'Customer Type')+
  theme_light() + theme(
    plot.title = element_text(
      color = "black",
      size = 13,
      face = "bold",
      hjust = 0.5
    ),
    plot.subtitle = element_text(hjust = 0.5)
  ) +  theme(panel.grid.major.x = element_blank(),
             panel.grid.minor.x = element_blank()) +
  theme(strip.background = element_rect(fill = c("#81f70a"))) +
  theme(strip.text = element_text(colour = 'black', face = "bold"))

data_vis_8
```

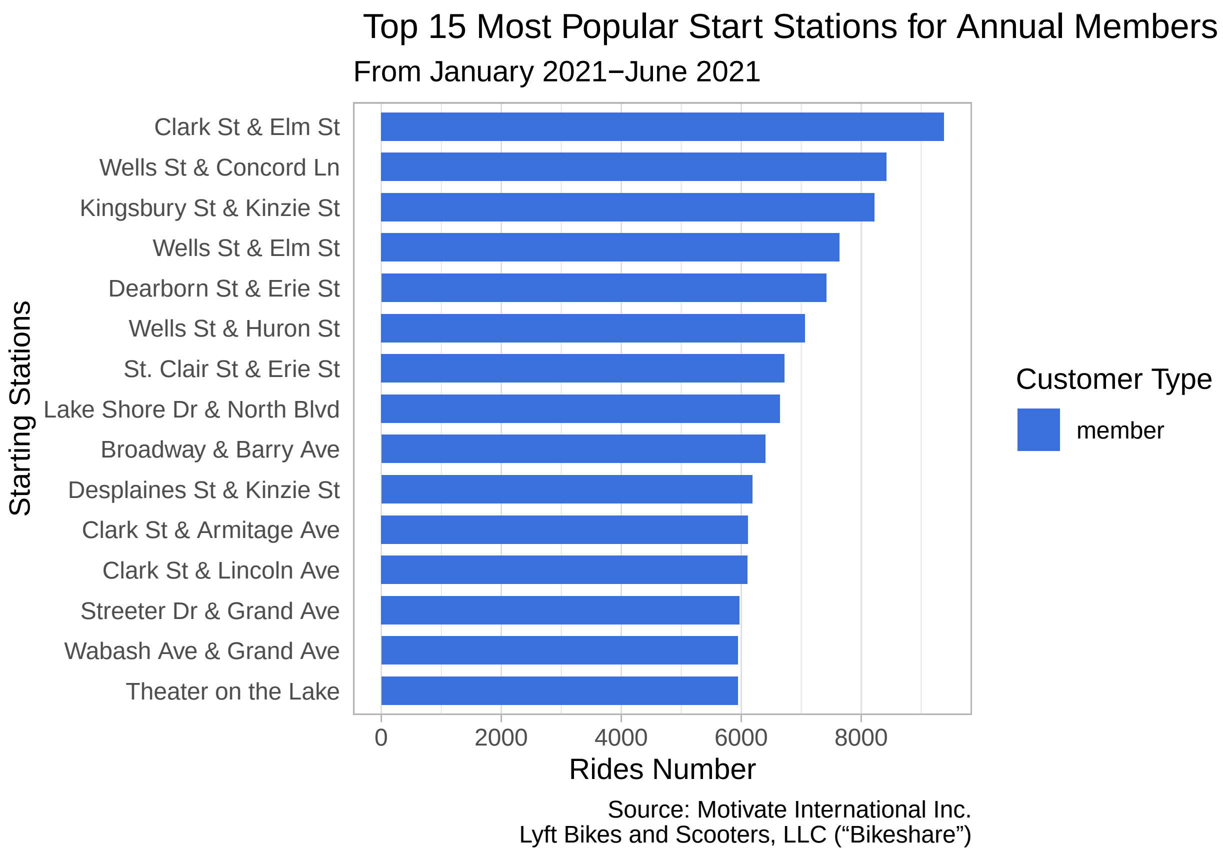


Annual Members vs. Casual Riders
Total Rides Number by Month

From January 2021−June 2021

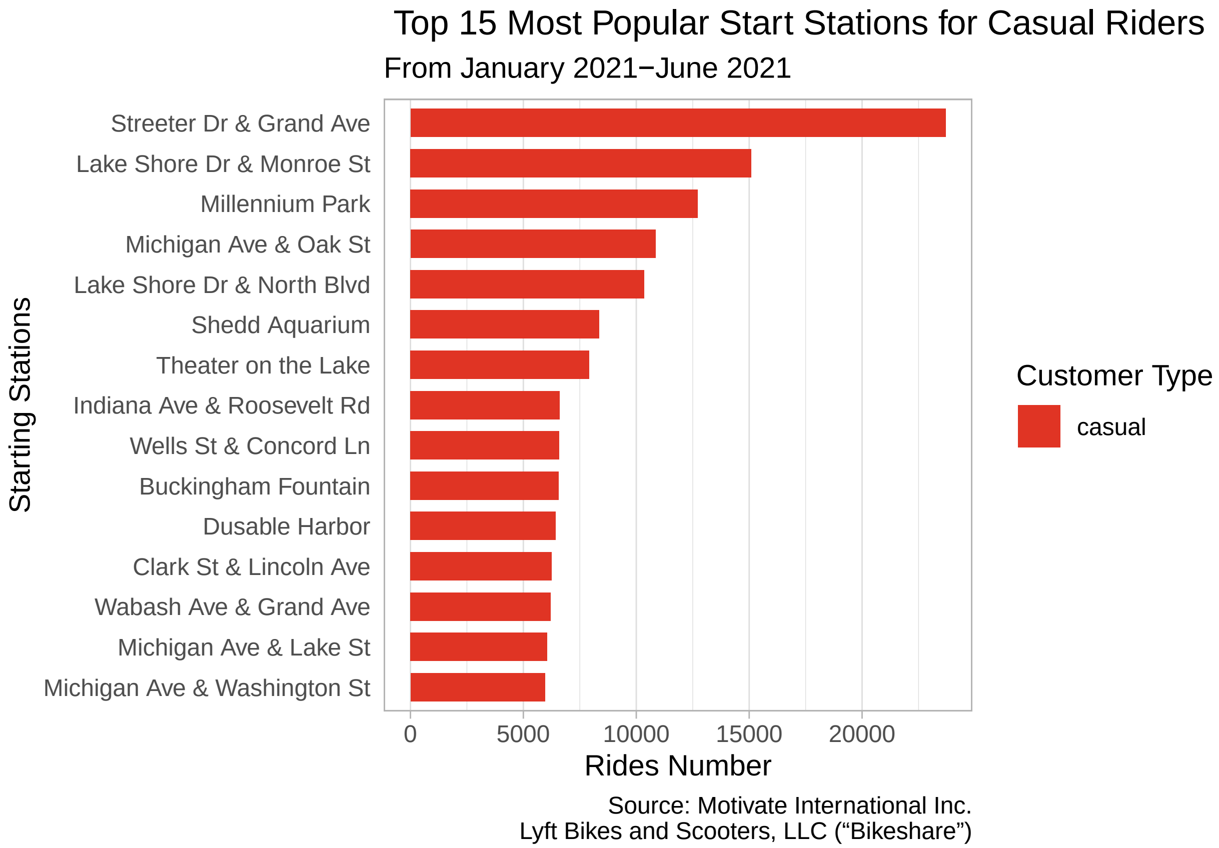
Source: Motivate International Inc.
Lyft Bikes and Scooters, LLC (“Bikeshare”)

```
data_vis_9 <- top_15_stations %>%
  filter(customer_type == "member") %>%
  ggplot() +
  geom_bar(aes(x = reorder(start_station_name, rides_number), y = rides_number,
               fill = customer_type), stat = "identity", width = 0.7) + coord_flip() +
  scale_y_continuous(n.breaks = 7) +
  scale_fill_manual(values = "#3970dd") +
  labs(
    title = paste(" Top 15 Most Popular Start Stations for Annual Members"),
    captions =
      "Source: Motivate International Inc.\nLyft Bikes and Scooters, LLC ("Bikeshare")",
    subtitle = "From January 2021-June 2021",
    x = "Starting Stations",
    y = "Rides Number"
  ) +  labs(fill = 'Customer Type') +
  theme_light() + theme(
    plot.title = element_text(
      color = "black",
      size = 13,
      face = "bold",
      hjust = 0.5
    ),
    plot.subtitle = element_text(hjust = 0.5)
  ) + theme_light() +
  theme(
    panel.grid.major.y = element_blank(),
    axis.ticks.y = element_blank()
  )
data_vis_9
```



```
data_vis_10 <- top_15_stations %>%
  filter(customer_type == "casual") %>%
  ggplot() +
  geom_bar(aes(x = reorder(start_station_name, rides_number), y = rides_number,
               fill = customer_type), stat = "identity", width = 0.7) + coord_flip() +
  scale_y_continuous(n.breaks = 7) +
  scale_fill_manual(values = "#e03424") +
  labs(
    title = paste(" Top 15 Most Popular Start Stations for Casual Riders"),
    captions =
      "Source: Motivate International Inc.\nLyft Bikes and Scooters, LLC ("Bikeshare")",
    subtitle = "From January 2021-June 2021",
    x = "Starting Stations",
    y = "Rides Number"
  ) +  labs(fill = 'Customer Type') +
  theme_light() + theme(
    plot.title = element_text(
      color = "black",
      size = 13,
      face = "bold",
      hjust = 0.5
    ),
    plot.subtitle = element_text(hjust = 0.5)
  ) + theme_light() +
  theme(
    panel.grid.major.y = element_blank(),
    axis.ticks.y = element_blank()
  )

data_vis_10
```



Top 15 Most Popular Start Stations for Casual Riders

From January 2021−June 2021

Source: Motivate International Inc.
Lyft Bikes and Scooters, LLC (“Bikeshare”)

Act

After spotting the differences between casual and member riders, marketing strategies to target casual riders can be developed to persuade them to become members.

Here are my top 3 recommendations based on my findings:

1. Introduce discounts to weekend rides, since casual riders prefer to ride bikes on Friday, Saturday and Sunday. Also offer discounts for peak time 4PM and 6PM.
2. Streeter Dr & Grand Ave stations are the most popular stations for casual riders, therefore there should be more focus on these stations in the aspect of special discount for long ride, coupons, complimentary trips, and marketing campaigns.
3. Start a fun bike competition with prizes in the months of May and June for riders, since casual riders have longest rides and highest number of rides in both months.