Google Analytics Capstone Project

Case study: How does a bike-share navigate speedy success?

Danny Nguyen

Table of Contents

Introduction

I worked on the Google Data Analytics Professional Certificate Capstone Project, "Case study: How does a bike-share navigate speedy success?". I assume the position of a junior data analyst working on the marketing analyst team at Cyclistic.

The **overall goal** is to design marketing strategies aimed at **converting casual riders into annual members**.

To do so, I will follow the steps of the data analysis process: Ask, Prepare, Process, Analyze, Share, and Act, to make recommendations backed by compelling data insights and professional data visualizations.

Background

Cyclistic is a fictional company that offers a bike-share program that has 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.

Previously, the Cyclistic's marketing strategy was to focus on building general awareness and appeal to broad consumer segments. The approach was to offer flexible pricing plans: single-ride passes, full-day passes, and annual memberships.

Customers who purchased **single-ride or full-day passes** are referred to as **casual riders** while those who purchase annual memberships are Cyclistic **annual members**.

Ask

The director of marketing believes that maximizing the number of annual members will be key to future growth and that there is a solid opportunity to convert casual riders into members.

The concerning stakeholders are the Cyclistic executive team.

To assist with accomplishing the **business task**, I am assigned with answering the following question: **"How do annual members and casual riders use Cyclistic bikes differently?"**.

Understanding this difference will be a key factor in developing the strategy to convert casual riders into annual members.

Prepare

We use historical Cyclistic trip data to analyze and identify trends.

This is public data (made available by Motivate International Inc. under this license and all personal customer information have been removed for data-privacy issues.

We begin by downloading the past 12 months (July 2023 - June 2024) from divvy-tripdata.

The datasets are stored separately by month in CSV files.

The datasets are: reliable and original as it is collected directly from the company's customers as a primary source, comprehensive as critical information for our findings are present, current as we are using data from the most recent 12 months, and cited as seen in the license.

Therefore we can assume there are no issues with bias or credibility in this data before we begin our analysis.

```
#Load packages
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.
0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(lubridate)
library(conflicted)
  conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.

  conflict_prefer("lag", "dplyr")

## [conflicted] Will prefer dplyr::lag over any other package.

  conflict_prefer("wday", "lubridate")

## [conflicted] Will prefer lubridate::wday over any other package.

  conflict_prefer("hour", "lubridate")

## [conflicted] Will prefer lubridate::hour over any other package.

library(hms)
library(here)

## here() starts at C:/Analytics/Capstone/Completed Project

library(skimr)
library(janitor)
library(data.table)

# Set global options
knitr::opts_chunk$set(
  echo = TRUE,
  warning = FALSE,
  message = FALSE
  )

# Turn off scientific notation
options(scipen=999)

# Load .csv files, 12 months of data from July 2023 to June 2024
jul2023 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202307-divvy-tripd
ata.csv")
aug2023 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202308-divvy-tripd
ata.csv")
sep2023 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202309-divvy-tripd
ata.csv")
oct2023 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202310-divvy-tripd
ata.csv")
nov2023 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202311-divvy-tripd
ata.csv")
dec2023 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202312-divvy-tripd
ata.csv")
jan2024 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202401-divvy-tripd
ata.csv")
feb2024 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202402-divvy-tripd
ata.csv")
mar2024 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202403-divvy-tripd
ata.csv")
```

```r
apr2024 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202404-divvy-tripd
ata.csv")
may2024 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202405-divvy-tripd
ata.csv")
jun2024 <- read.csv("C:/Analytics/Capstone/Case_Study/data/202406-divvy-tripd
ata.csv")

#Merge all data frames
cyclistic_merged <- rbind(jul2023, aug2023, sep2023, oct2023, nov2023, dec202
3, jan2024, feb2024, mar2024, apr2024, may2024, jun2024)
```

Inspect data pre-clean up

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

##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"

#Preview first 6 rows of data frame
head(cyclistic_merged)

##            ride_id rideable_type          started_at            ended_at
## 1 9340B064F0AEE130 electric_bike 2023-07-23 20:06:14 2023-07-23 20:22:44
## 2 D1460EE3CE0D8AF8  classic_bike 2023-07-23 17:05:07 2023-07-23 17:18:37
## 3 DF41BE31B895A25E  classic_bike 2023-07-23 10:14:53 2023-07-23 10:24:29
## 4 9624A293749EF703 electric_bike 2023-07-21 08:27:44 2023-07-21 08:32:40
## 5 2F68A6A4CDB4C99A  classic_bike 2023-07-08 15:46:42 2023-07-08 15:58:08
## 6 9AEE973E6B941A9C  classic_bike 2023-07-10 08:44:47 2023-07-10 08:49:41
##         start_station_name start_station_id                    end_station
_name
## 1    Kedzie Ave & 110th St            20204 Public Rack - Racine Ave & 109
th Pl
## 2  Western Ave & Walton St     KA1504000103          Milwaukee Ave & Gran
d Ave
## 3  Western Ave & Walton St     KA1504000103             Damen Ave & Pierc
e Ave
## 4 Racine Ave & Randolph St            13155            Clinton St & Madis
on St
## 5    Clark St & Leland Ave     TA1309000014                    Montrose H
arbor
## 6 Racine Ave & Randolph St            13155              Sangamon St & La
ke St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1            877  41.69241 -87.70091 41.69483 -87.65304        member
## 2          13033  41.89842 -87.68660 41.89158 -87.64838        member
## 3   TA1305000041  41.89842 -87.68660 41.90940 -87.67769        member
## 4   TA1305000032  41.88411 -87.65694 41.88275 -87.64119        member
```

```
## 5   TA1308000012  41.96709 -87.66729 41.96398 -87.63818        member
## 6   TA1306000015  41.88407 -87.65685 41.88578 -87.65102        member
```

```r
#See list of columns and data types
str(cyclistic_merged)
```

```
## 'data.frame':    5734381 obs. of  13 variables:
##  $ ride_id           : chr  "9340B064F0AEE130" "D1460EE3CE0D8AF8" "DF41BE3
1B895A25E" "9624A293749EF703" ...
##  $ rideable_type     : chr  "electric_bike" "classic_bike" "classic_bike"
"electric_bike" ...
##  $ started_at        : chr  "2023-07-23 20:06:14" "2023-07-23 17:05:07" "2
023-07-23 10:14:53" "2023-07-21 08:27:44" ...
##  $ ended_at          : chr  "2023-07-23 20:22:44" "2023-07-23 17:18:37" "2
023-07-23 10:24:29" "2023-07-21 08:32:40" ...
##  $ start_station_name: chr  "Kedzie Ave & 110th St" "Western Ave & Walton
St" "Western Ave & Walton St" "Racine Ave & Randolph St" ...
##  $ start_station_id  : chr  "20204" "KA1504000103" "KA1504000103" "13155"
...
##  $ end_station_name  : chr  "Public Rack - Racine Ave & 109th Pl" "Milwauk
ee Ave & Grand Ave" "Damen Ave & Pierce Ave" "Clinton St & Madison St" ...
##  $ end_station_id    : chr  "877" "13033" "TA1305000041" "TA1305000032" ..
.
##  $ start_lat         : num  41.7 41.9 41.9 41.9 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.7 41.9 41.9 41.9 42 ...
##  $ end_lng           : num  -87.7 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```r
#Statistical summary of data
summary(cyclistic_merged)
```

```
##    ride_id          rideable_type       started_at          ended_at        
##  Length:5734381     Length:5734381     Length:5734381     Length:5734381    
##  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:5734381     Length:5734381     Length:5734381     Length:5734381    
##  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.   :41.63   Min.   :-87.94   Min.   : 0.00   Min.   :-88.12  
##  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.   :-87.46   Max.   :42.19   Max.   :  0.00  
##                                   NA's   :7919    NA's   :7919    
##  member_casual     
##  Length:5734381    
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
##                    
```

```
#Quick check to ensure member_casual only has two distinct values: member or casual.
n_distinct(cyclistic_merged$member_casual)
```

```
## [1] 2
```

Process

Documenting the manipulation and cleaning of data.

```
#Create a new data frame to contain changes
cyclistic_data <- cyclistic_merged

#Calculating "ride_length" by subtracting "start_at" time from "ended_at" time in minutes.
cyclistic_data$ride_length <- difftime(cyclistic_merged$ended_at, cyclistic_merged$started_at, units = "mins")

#Check to see if there are any values of "ride_length" that are 0 or negative. We will remove these in the next steps.
nrow(subset(cyclistic_data, ride_length <= 0))
```

```
## [1] 1733
```

```
#Creating new columns that list the date, month, day, and year of each ride for further insight into data.
cyclistic_data$date <- as.Date(cyclistic_data$started_at) #default format is yyyy-mm-dd, use start date
cyclistic_data$month <- format(as.Date(cyclistic_data$date), "%m") #create column for month
cyclistic_data$day <- format(as.Date(cyclistic_data$date), "%d") #create column for day
cyclistic_data$year <- format(as.Date(cyclistic_data$date), "%Y") #create column for year
cyclistic_data$day_of_week <- wday(cyclistic_data$started_at) #calculate the day of the week
```

```
cyclistic_data$day_of_week <- format(as.Date(cyclistic_data$date), "%A") #cre
ate column for day of week
cyclistic_hour <- cyclistic_merged %>%
  separate(started_at, into = c("Date", "Time"), sep = " ") #created a new df
to separate time from "started_at" in order to source the column for hour
cyclistic_data$time <- format(as.Date(cyclistic_data$date), "%H:%M:%S") #form
at time as HH:MM:SS
cyclistic_data$time <- as_hms((cyclistic_hour$Time)) #create column for time
cyclistic_data$hour <- hour(cyclistic_data$time) #create new column for hour

#Order days of the week
cyclistic_data$day_of_week <- ordered(cyclistic_data$day_of_week, levels = c(
"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
)

#Clean the data
cyclistic_data <- distinct(cyclistic_data) #remove duplicate rows
cyclistic_data <- na.omit(cyclistic_data) #remove rows with NA values
cyclistic_data <- cyclistic_data[!(cyclistic_data$ride_length <=0),] #remove
rows where "ride_length" is 0 or negative.
cyclistic_data <- cyclistic_data %>%
  select(-c(start_station_id, end_station_id, start_lat, start_lng, end_lat,
end_lng)) #remove unneeded columns: "ride_id", "start_station_id", "end_stati
on_id", "start_lat", "start_long", "end_lat", "end_lng"

#View the data we will use
View(cyclistic_data)
```

Analyze

Aggregating, organizing, formatting, and visualizing the data in order to perform calculations and to identify trends and relationships.

```
summary(cyclistic_data)

##    ride_id           rideable_type       started_at          ended_at
##  Length:5724729     Length:5724729     Length:5724729     Length:5724729
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  start_station_name end_station_name   member_casual      ride_length
##  Length:5724729     Length:5724729     Length:5724729     Length:5724729
##  Class :character   Class :character   Class :character   Class :difftime
##  Mode  :character   Mode  :character   Mode  :character   Mode  :numeric
##
##
```

```
##
##
##       date                 month               day                 year
##  Min.   :2023-07-01   Length:5724729     Length:5724729     Length:5724729
##  1st Qu.:2023-08-27   Class :character   Class :character   Class :charact
er
##  Median :2023-11-09   Mode  :character   Mode  :character   Mode  :charact
er
##  Mean   :2023-12-16
##  3rd Qu.:2024-04-23
##  Max.   :2024-06-30
##
##     day_of_week        time                 hour
##  Sunday   :780801   Length:5724729    Min.   : 0.00
##  Monday   :747670   Class1:hms        1st Qu.:11.00
##  Tuesday  :810400   Class2:difftime   Median :15.00
##  Wednesday:838636   Mode  :numeric    Mean   :14.08
##  Thursday :836503                     3rd Qu.:18.00
##  Friday   :813351                     Max.   :23.00
##  Saturday :897368
```

```
#Total number of rides
nrow(cyclistic_data)
```

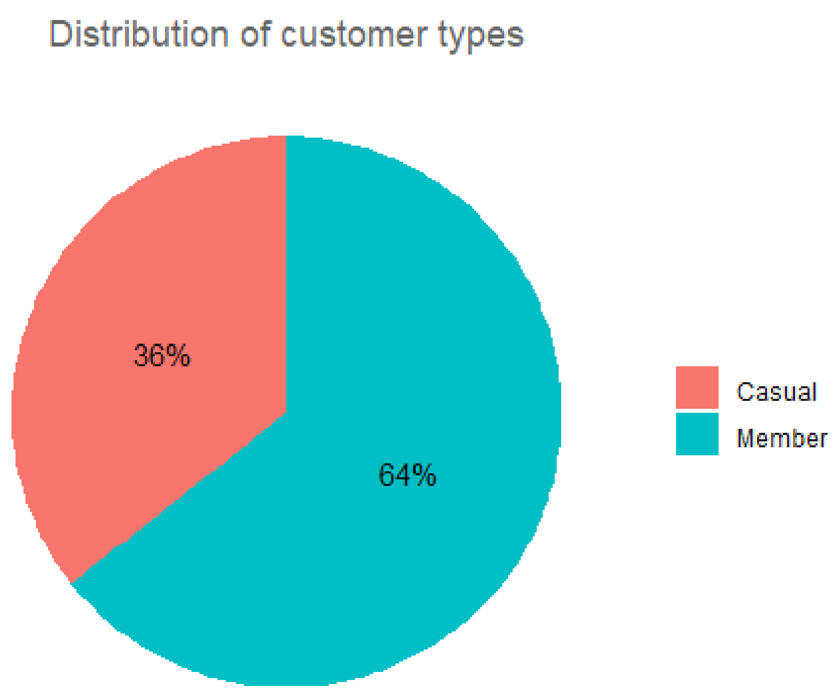
```
## [1] 5724729
```

```
#Total number of rides for each customer type
cyclistic_data %>%
  group_by(member_casual) %>%
  summarise(count = length(ride_id),
            '%' = (length(ride_id) / nrow(cyclistic_data)) *100)
```

```
## # A tibble: 2 × 3
##   member_casual   count   `%`
##   <chr>           <int> <dbl>
## 1 casual        2042350  35.7
## 2 member        3682379  64.3
```

```
#Creating a data frame for a pie chart
pie.df = data.frame("type" = c("Casual", "Member"),
                    "count" = c(.3567592, .6432408))

pie = ggplot(pie.df, aes(x = "", y = count, fill = type)) +
  geom_bar(stat = "identity", width = 1)

#Convert to pie
pie = pie + coord_polar("y", start = 0) +
  geom_text(aes(label = paste0(round(count * 100), "%")),
            position = position_stack(vjust = 0.5))

pie = pie + labs(x = NULL, y = NULL, fill = NULL, title = "Distribution of cu
```

```
stomer types")

pie = pie + theme_classic() + theme(axis.line = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          plot.title = element_text(hjust = 0.5, color = "#666666"))
pie
```



- From the table and graph, we see that casual customers make up about **36%** of the customer base where as members make up about **64%** of the customer base.

Ride Length

```
#Average ride length
cyclistic_data %>%
  summarise(mean = mean(ride_length))

##             mean
## 1 15.57333 mins

#Summary statistics of ride length by customer type
cyclistic_data %>%
  group_by(member_casual) %>%
  summarise(mean = mean(ride_length),
            'median' = median(ride_length),
```

```
            'min' = min(ride_length),
            'max' = max(ride_length))
```

```
## # A tibble: 2 × 5
##   member_casual mean          median         min               max
##   <chr>         <drtn>        <drtn>         <drtn>            <drtn>
## 1 casual        21.36991 mins 12.133333 mins 0.0017333349 mins 6891.217 mi
ns
## 2 member        12.35839 mins  8.716667 mins 0.0006499966 mins 1499.933 mi
ns
```

- Notice that the max ride length times for each customer type (6891.2 mins and 1499.9 mins) are significantly greater than their average ride length times (21.4 mins and 12.4 mins).
- Our min ride length times are also significantly smaller (0.0017 and 0.0007 mins) as well.
- This may be an issue if we try to plot or analyze as it may skew our data.
- Let us note that the highest ride time of 6891.2 minutes is almost 115 hours and lowest ride time of 0.0007 minutes is only 0.04 seconds. These times do not seem plausible and may be the result of a bike not being returned/docked on the higher side, or a technical issue with a ride being instantly started and ended on the lower side.
- We will look to exclude these values from our analysis to prevent any skewness as they do not accurately represent our target customer base.

```
#Gathering percentiles
ventiles = quantile(cyclistic_data$ride_length, seq(0, 1, by = 0.05))
format(x = ventiles, scientific = FALSE)
```

```
##  [1] "   0.0006499966 mins" "   2.2333333333 mins" "   3.3000000000 mins"
##  [4] "   4.1000000000 mins" "   4.8333333333 mins" "   5.5500000000 mins"
##  [7] "   6.2833333333 mins" "   7.0666666667 mins" "   7.8833333333 mins"
## [10] "   8.7666666667 mins" "   9.7333333333 mins" "  10.8000000000 mins"
## [13] "  12.0166666667 mins" "  13.4333333333 mins" "  15.1333333333 mins"
## [16] "  17.2333333333 mins" "  19.9666666667 mins" "  23.7500000000 mins"
## [19] "  29.7333333333 mins" "  42.0333333333 mins" "6891.2166666667 mins"
```

```
ventiles
```

```
## Time differences in mins
##              0%              5%             10%             15%
20%
##    0.0006499966    2.2333333333    3.3000000000    4.1000000000    4.83333
33333
##             25%             30%             35%             40%
45%
##    5.5500000000    6.2833333333    7.0666666667    7.8833333333    8.76666
```

```
66667
##             50%             55%             60%             65%             
70%
##    9.7333333333   10.8000000000   12.0166666667   13.4333333333   15.13333
33333
##             75%             80%             85%             90%             
95%
##   17.2333333333   19.9666666667   23.7500000000   29.7333333333   42.03333
33333
##            100% 
## 6891.2166666667
```

- We see that the difference between the 0th and 100th percentile is about 6,891.2 minutes whereas the difference between the 5th and 95th percentile is only about 39.8 minutes.
- We will treat the 0-5th percentile and 95-100th percentiles as outliers and exclude them in our analysis of the "ride_length" variable.

```
#Removing ride length outliers
cyclistic_data_no_outliers <- cyclistic_data %>%
  filter(ride_length > as.numeric(ventiles['5%'])) %>%
  filter(ride_length < as.numeric(ventiles['95%']))

print(paste("Removed", nrow(cyclistic_data) - nrow(cyclistic_data_no_outliers
), "rows as outliers"))
```

```
## [1] "Removed 574234 rows as outliers"
```

```
#Average ride length without outliers
cyclistic_data_no_outliers %>%
  summarise(mean = mean(ride_length))
```

```
##            mean
## 1 12.23896 mins
```

```
#Summary statistics of ride length without outliers
cyclistic_data_no_outliers %>%
  group_by(member_casual) %>%
  summarise(mean = mean(ride_length),
            'median' = median(ride_length),
            'min' = min(ride_length),
            'max' = max(ride_length))
```

```
## # A tibble: 2 × 5
##   member_casual mean          median        min           max          
##   <chr>         <drtn>        <drtn>        <drtn>        <drtn>       
## 1 casual        13.94048 mins 11.38333 mins 2.233567 mins 42.03228 mins
## 2 member        11.37071 mins  9.00000 mins 2.233367 mins 42.03183 mins
```

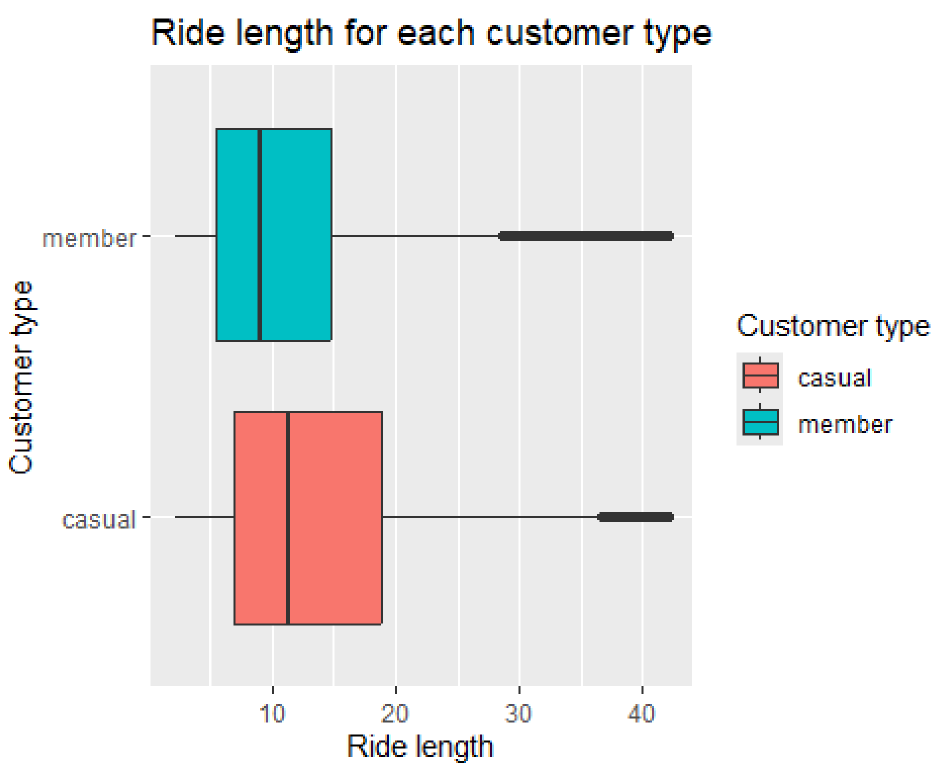
- Without outliers, we see interesting changes to our data.

- The mean time for casual customers drops by about 7 minutes. Whereas the mean time for members only drops by about 1 minute. This is expected as we saw the max time for casual customers was substantially higher than for members prior to excluding the outliers.

- Median times were stable before and after the change which makes sense since the median should be resistant to outliers as a measure.

- And more interestingly, the min and max times for casuals and members are now almost identical.

```
#Visualizing distribution of ride length for each customer type
ggplot(cyclistic_data_no_outliers, aes(x = member_casual, y = ride_length, fi
ll = member_casual)) +
  labs(x = "Customer type", y = "Ride length", title = "Ride length for each
customer type", fill = "Customer type") +
  geom_boxplot() +
  coord_flip()
```



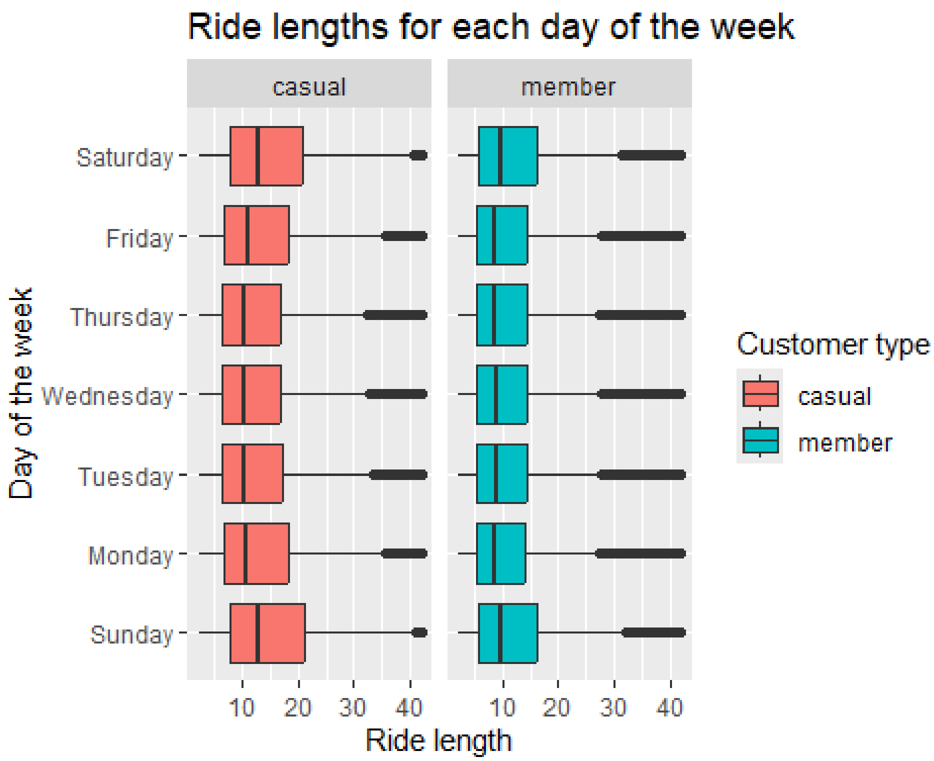
- From the box plot, we see that casual customers have more riding time than members but also have a larger interquartile range, telling us that there is more spread/variability in casual customers' riding times.

- We will dive further by plotting by day of the week next.

```r
#Average ride length by each day of the week for each customer type
aggregate(cyclistic_data_no_outliers$ride_length ~ cyclistic_data_no_outliers
$member_casual + cyclistic_data_no_outliers$day_of_week, FUN = mean)
```

```
##    cyclistic_data_no_outliers$member_casual
## 1                                    casual
## 2                                    member
## 3                                    casual
## 4                                    member
## 5                                    casual
## 6                                    member
## 7                                    casual
## 8                                    member
## 9                                    casual
## 10                                   member
## 11                                   casual
## 12                                   member
## 13                                   casual
## 14                                   member
##    cyclistic_data_no_outliers$day_of_week
## 1                                  Sunday
## 2                                  Sunday
## 3                                  Monday
## 4                                  Monday
## 5                                 Tuesday
## 6                                 Tuesday
## 7                               Wednesday
## 8                               Wednesday
## 9                                Thursday
## 10                               Thursday
## 11                                 Friday
## 12                                 Friday
## 13                               Saturday
## 14                               Saturday
##    cyclistic_data_no_outliers$ride_length
## 1                           15.32333 mins
## 2                           12.24953 mins
## 3                           13.50030 mins
## 4                           10.99174 mins
## 5                           12.99102 mins
## 6                           11.14718 mins
## 7                           12.83649 mins
## 8                           11.13308 mins
## 9                           12.76811 mins
## 10                          11.04282 mins
## 11                          13.65410 mins
## 12                          11.15515 mins
## 13                          15.27409 mins
## 14                          12.21037 mins
```

```
#Visualizing average ride length by day of the week for each customer type
ggplot(cyclistic_data_no_outliers, aes(x = day_of_week, y = ride_length, fill
= member_casual)) +
  geom_boxplot() +
  labs(x = "Day of the week", y = "Ride length", title = "Ride lengths for ea
ch day of the week", fill = "Customer type") +
  facet_wrap(~member_casual) +
  coord_flip()
```



- We see that casual customers' riding times follow a curved distribution, peaking towards the weekend, primarily on Saturday and Sunday, and falling towards the middle of week on Wednesdays on average.
- Members' riding times remain seemingly constant throughout the weekday and increases during the weekend on average. The consistency in members' ride times may be due to members riding to and from the locations each weekday.

Day of the Week

```
#Total rides for each day of the week by customer type
cyclistic_data %>%
  group_by(member_casual) %>%
  count(day_of_week)
```

```
## # A tibble: 14 × 3
## # Groups:   member_casual [2]
##    member_casual day_of_week      n
##    <chr>         <ord>        <int>
##  1 casual        Sunday      356854
##  2 casual        Monday      236380
##  3 casual        Tuesday     238406
##  4 casual        Wednesday   247179
##  5 casual        Thursday    252787
##  6 casual        Friday      294314
##  7 casual        Saturday    416430
##  8 member        Sunday      423947
##  9 member        Monday      511290
## 10 member        Tuesday     571994
## 11 member        Wednesday   591457
## 12 member        Thursday    583716
## 13 member        Friday      519037
## 14 member        Saturday    480938
```

```
#Percentages for total rides for each day of the week by customer type
cyclistic_data %>%
  group_by(day_of_week) %>%
  summarise(count = length(ride_id),
            '%' = (length(ride_id) / nrow(cyclistic_data)) * 100,
            'members_%' = (sum(member_casual == "member") / length(ride_id))
*100,
            'casual_%' = (sum(member_casual == "casual") / length(ride_id)) *
100)
```

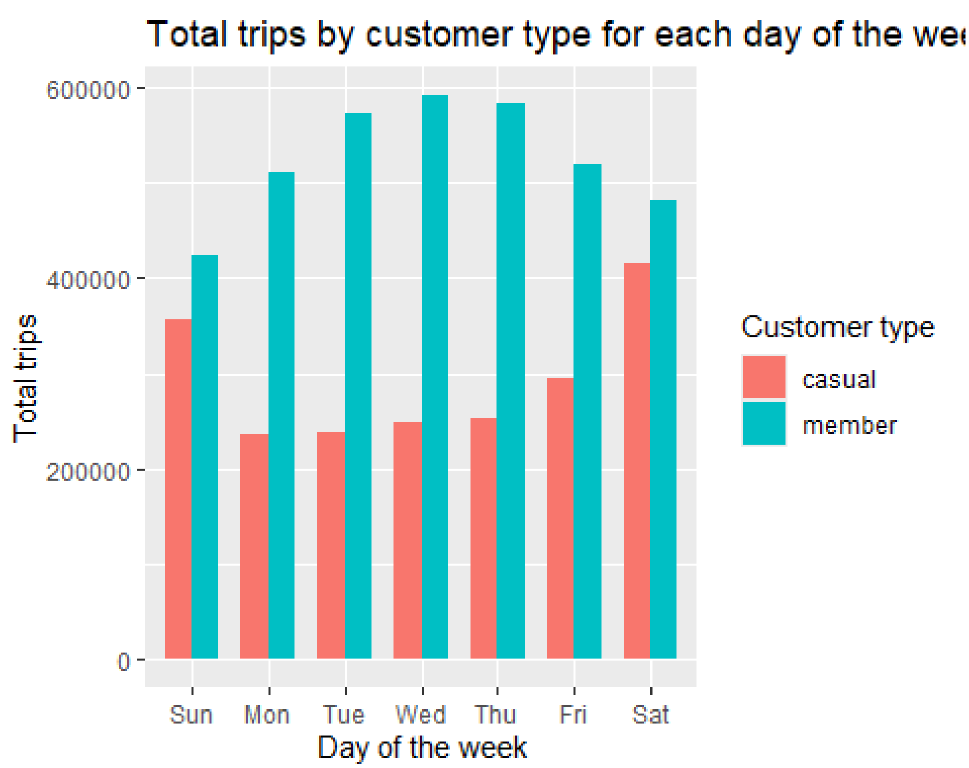
```
## # A tibble: 7 × 5
##   day_of_week  count   `%` `members_%` `casual_%`
##   <ord>        <int> <dbl>       <dbl>      <dbl>
## 1 Sunday      780801  13.6        54.3       45.7
## 2 Monday      747670  13.1        68.4       31.6
## 3 Tuesday     810400  14.2        70.6       29.4
## 4 Wednesday   838636  14.6        70.5       29.5
## 5 Thursday    836503  14.6        69.8       30.2
## 6 Friday      813351  14.2        63.8       36.2
## 7 Saturday    897368  15.7        53.6       46.4
```

```
#Analyze ridership data by customer type and weekday
cyclistic_data %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),
        average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)
```

```
## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
```

```
##    <chr>         <ord>          <int> <drtn>
##  1 casual        Sun           356854 24.91630 mins
##  2 casual        Mon           236380 21.00678 mins
##  3 casual        Tue           238406 18.99438 mins
##  4 casual        Wed           247179 18.60025 mins
##  5 casual        Thu           252787 18.26033 mins
##  6 casual        Fri           294314 20.51807 mins
##  7 casual        Sat           416430 24.03063 mins
##  8 member        Sun           423947 13.83024 mins
##  9 member        Mon           511290 11.84054 mins
## 10 member        Tue           571994 11.91936 mins
## 11 member        Wed           591457 11.94968 mins
## 12 member        Thu           583716 11.75783 mins
## 13 member        Fri           519037 12.11747 mins
## 14 member        Sat           480938 13.62521 mins
```

```
#Trips for each day of the week by customer type
cyclistic_data %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),
        average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday) %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  labs(title = "Total trips by customer type for each day of the week") +
  labs(x = "Day of the week", y = "Total trips", fill = "Customer type") +
  geom_col(width = 0.7, position = position_dodge(width = 0.7))
```



- This follows the distribution we gathered in our earlier plot, “Ride length by customer type for each day of the week”.
- We can gather that casual customers are primarily using the bikeshare on the weekends, primarily on Sunday and Saturday. Whereas members are riding increasingly more throughout the week, peaking on Wednesdays and decreasing until the week ends.

Hour

```
#Total number of rides per hour of the day by customer type
cyclistic_data %>%
  group_by(member_casual) %>%
  count(hour)
```

```
## # A tibble: 48 × 3
## # Groups:   member_casual [2]
##    member_casual  hour     n
##    <chr>         <int> <int>
##  1 casual            0 35346
##  2 casual            1 23349
##  3 casual            2 14357
##  4 casual            3  8021
##  5 casual            4  5920
##  6 casual            5 11189
##  7 casual            6 27770
##  8 casual            7 50533
##  9 casual            8 69569
## 10 casual            9 70463
## # ℹ 38 more rows
```

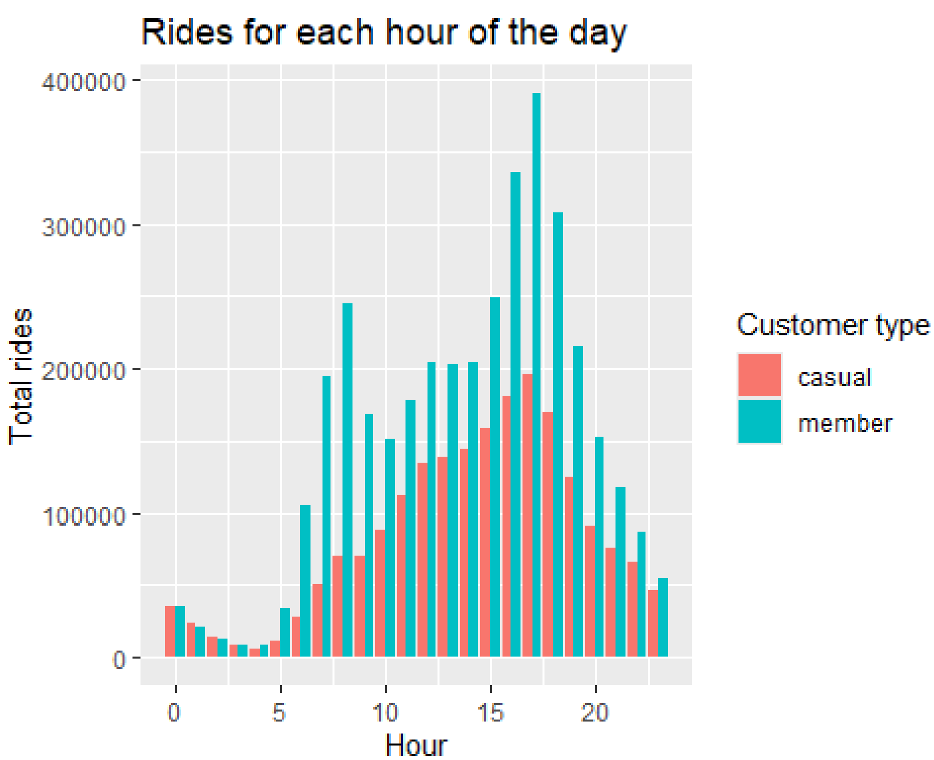
```
#Percentages for total rides per month by customer type
cyclistic_data %>%
  group_by(hour) %>%
  summarise(count = length(ride_id),
            '%' = (length(ride_id) / nrow(cyclistic_data)) * 100,
            'members_%' = (sum(member_casual == "member") / length(ride_id))
* 100,
            'casual_%' = (sum(member_casual == "casual") / length(ride_id)) *
100)
```

```
## # A tibble: 24 × 5
##     hour  count   `%` `members_%` `casual_%`
##    <int>  <int> <dbl>       <dbl>      <dbl>
##  1     0  70101 1.22         49.6       50.4
##  2     1  44047 0.769        47.0       53.0
##  3     2  26329 0.460        45.5       54.5
##  4     3  16033 0.280        50.0       50.0
##  5     4  14855 0.259        60.1       39.9
```

```
## 6       5  45420 0.793         75.4        24.6
## 7       6 132790 2.32          79.1        20.9
## 8       7 245762 4.29          79.4        20.6
## 9       8 314794 5.50          77.9        22.1
## 10      9 238191 4.16          70.4        29.6
## # ℹ 14 more rows
```

- From the tibble, we see that from hour 5 to hour 6, ridership almost triples in count, and then almost doubles from hour 6 to hour 7.
- We also see a big percentage difference from the percentage of member and casual riders at these hours. This gap begins to decrease as the day continues but is still maintained throughout the night.
- Let's visualize this gap.

```
#Visualizing rides per hour of the day by customer type
cyclistic_data %>%
  ggplot(aes(hour, fill = member_casual)) +
  labs(x = "Hour", y = "Total rides",, title = "Rides for each hour of the da
y", fill = "Customer type") +
  geom_bar(position = 'dodge')
```



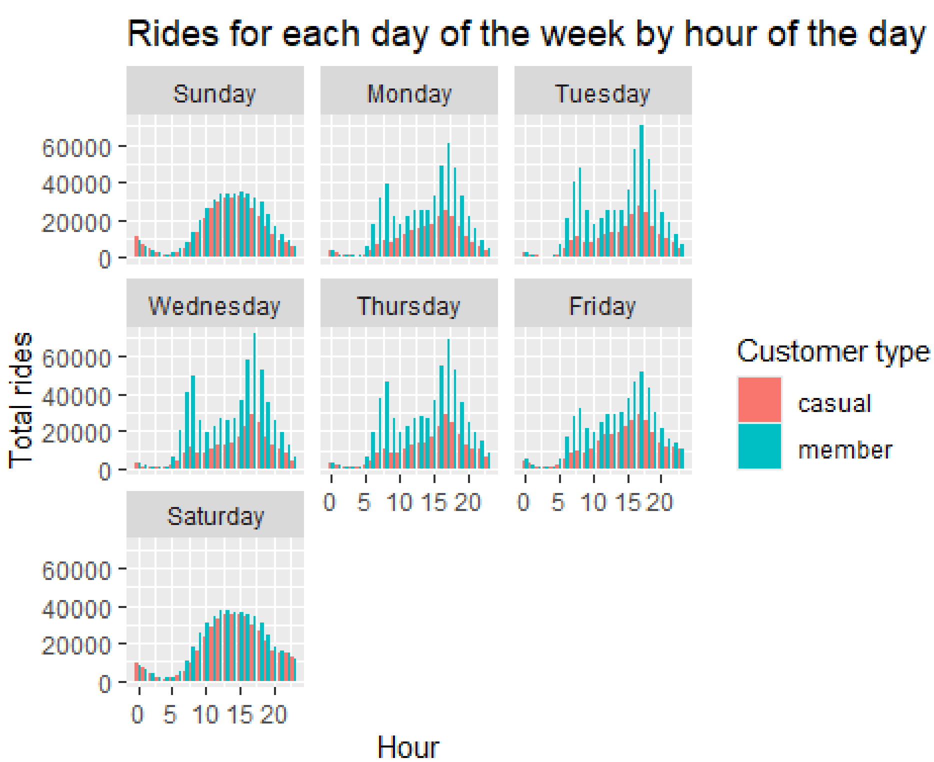
- We see from the chart that ridership peaks from the 16-18 hours (4pm-6pm), afternoon time frame for both customer types.

- There is a spike that begins at the 5 hour mark (5am) and greatly increases by each hour until the 8 hour mark (8am) for members. There is also a spike 16-18 hour marks (4pm-6pm) for members while the distribution of casual customers, for the most part, remains smooth.

- The percentage gap between casual riders and members remains close throughout the morning but increases into the afternoon and decreases towards the night as we see in the percentage make up of each hour.

- We will split the analysis by day of the week next for further analysis.

```
#Visualizing rides for each day of the week per hour of the day by customer t
ype
cyclistic_data %>%
  ggplot(aes(hour, fill = member_casual)) +
  geom_bar(position = 'dodge') +
  labs(x = "Hour", y = "Total rides", title = "Rides for each day of the week
by hour of the day ", fill = "Customer type") +
  facet_wrap(~day_of_week)
```

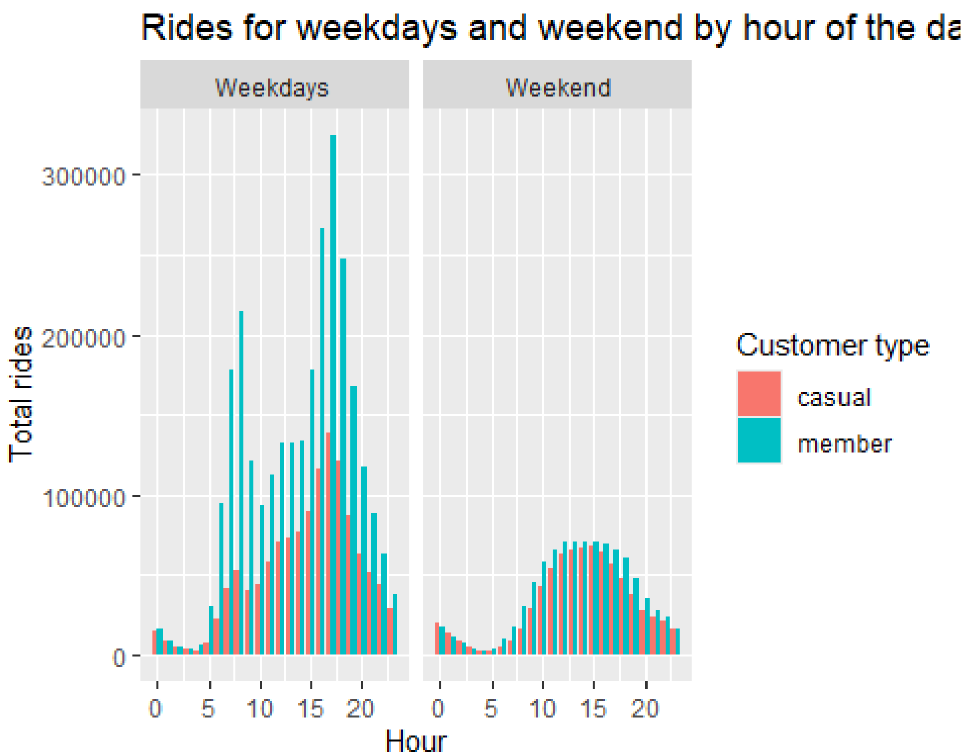


- We can see that the weekdays, Monday-Friday, all follow a similar distribution and that weekend, Sunday and Saturday, also share a similar distribution.

- Let's separate the weekdays and weekend to better understand this difference.

```
#Visualizing the weekday vs weekend difference
cyclistic_data %>%
```

```
  mutate(day_of_week = ifelse(day_of_week == 'Saturday' | day_of_week == 'Sun
day',
                              'Weekend',
                              'Weekdays')) %>%
  ggplot(aes(hour, fill = member_casual)) +
  labs(x = "Hour", y = "Total rides", title = "Rides for weekdays and weekend
by hour of the day", fill = "Customer type") +
  geom_bar(position = 'dodge') +
  facet_wrap(~day_of_week)
```



- Although the overall distributions of the weekdays and weekend plots are similar between customer types respectively, both low in the mornings, peaking in the afternoon, and dropping towards the night, we still see visible differences between the two. The weekdays distribution is much more jagged and steep whereas the weekend has a somewhat *smoother* distribution.

- The biggest difference we see when **separating the weekdays from the weekends** is that the 6am-8am and 4pm-6pm **spike is now apparent for casuals**, albeit they are not as accentuated. It is important to note that the **spike occurs at a much greater magnitude for members**. It is important to ascertain the reasoning behind these spikes.

- One assumption we can make is that these are times riders are likely to be commuting to and from work, school, or other daily routine activities. Therefore we can infer that a large number of riders opt in to membership for the sake of commuting during the workweek.

- This assumption may be further supported by the noticeable gap of total rides between members during the weekdays and members during the weekend. From the side by side comparison, we see that the distributions for casual riders are similar if you disregard the 6-8am and 4-6pm spikes during the weekdays. But the total number of rides for members are drastically higher during the weekdays than the weekends.

We another layer to the analysis by filtering out the weekend from the summary.

```
#Creating new data frame without weekends
cyclistic_no_weekend <- cyclistic_data %>%
  filter(day_of_week != "Saturday" & day_of_week != "Sunday")

#Total number of rides per hour of the day by customer type without weekends
cyclistic_no_weekend %>%
  group_by(member_casual) %>%
  count(hour)

## # A tibble: 48 × 3
## # Groups:   member_casual [2]
##    member_casual  hour     n
##    <chr>         <int> <int>
##  1 casual            0 15614
##  2 casual            1  9358
##  3 casual            2  5499
##  4 casual            3  3367
##  5 casual            4  3127
##  6 casual            5  8303
##  7 casual            6 22846
##  8 casual            7 41700
##  9 casual            8 52729
## 10 casual            9 40995
## # ℹ 38 more rows

#Percentages for total rides per month by customer type without weekends
cyclistic_no_weekend %>%
  group_by(hour) %>%
  summarise(count = length(ride_id),
            '%' = (length(ride_id) / nrow(cyclistic_no_weekend)) * 100,
            'members_%' = (sum(member_casual == "member") / length(ride_id))
* 100,
            'casual_%' = (sum(member_casual == "casual") / length(ride_id)) *
100)

## # A tibble: 24 × 5
##     hour  count   `%` `members_%` `casual_%`
##    <int>  <int> <dbl>       <dbl>      <dbl>
##  1     0  32595 0.805        52.1       47.9
##  2     1  18012 0.445        48.0       52.0
##  3     2  10255 0.253        46.4       53.6
##  4     3   7228 0.179        53.4       46.6
```

```
## 5     4   9287 0.230        66.3       33.7
## 6     5  38462 0.950        78.4       21.6
## 7     6 118082 2.92         80.7       19.3
## 8     7 219365 5.42         81.0       19.0
## 9     8 267276 6.61         80.3       19.7
## 10    9 162940 4.03         74.8       25.2
## # ℹ 14 more rows
```

- From 6am to 8am, members' have 315.6% more rides than casuals.
- From 4pm to 6pm, members' have 123.11% more rides than casuals.

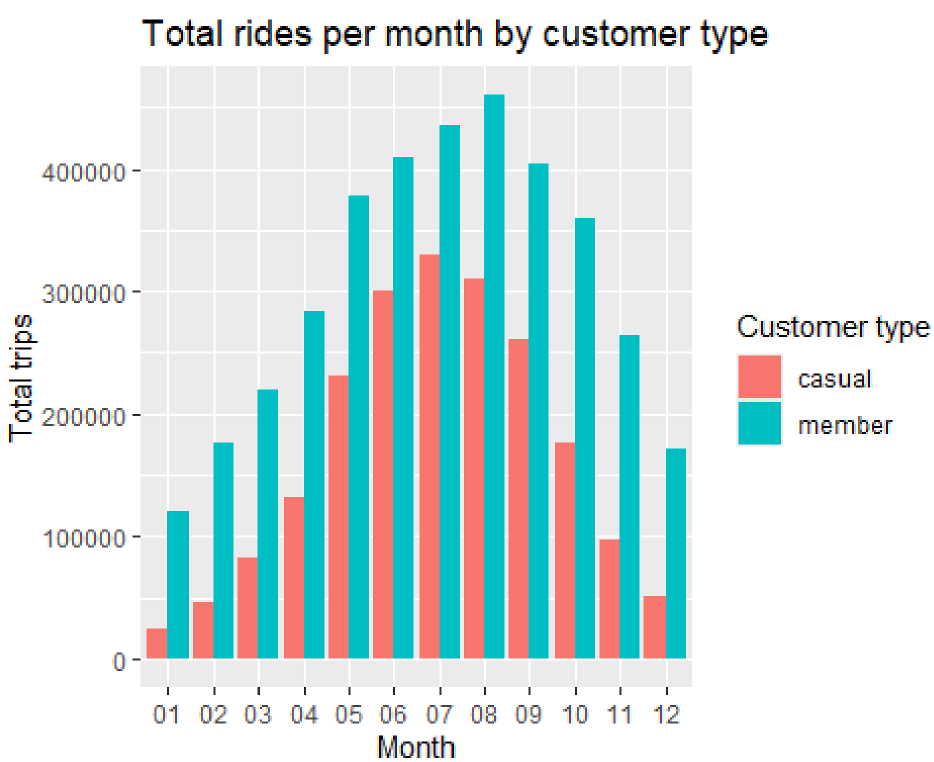
Month

```
#Total rides per month by customer type
cyclistic_data %>%
  group_by(member_casual) %>%
  count(month)

## # A tibble: 24 × 3
## # Groups:   member_casual [2]
##    member_casual month      n
##    <chr>         <chr>  <int>
##  1 casual        01     24339
##  2 casual        02     46957
##  3 casual        03     82218
##  4 casual        04    131366
##  5 casual        05    230363
##  6 casual        06    300071
##  7 casual        07    330142
##  8 casual        08    309931
##  9 casual        09    260836
## 10 casual        10    176553
## # ℹ 14 more rows

#Percentages for total rides per month by customer type
cyclistic_data %>%
  group_by(month) %>%
  summarise(count = length(ride_id),
            '%' = (length(ride_id) / nrow(cyclistic_data)) * 100,
            'members_%' = (sum(member_casual == "member") / length(ride_id))
*100,
            'casual_%' = (sum(member_casual == "casual") / length(ride_id)) *
100)

## # A tibble: 12 × 5
##    month  count   `%` `members_%` `casual_%`
##    <chr>  <int> <dbl>       <dbl>      <dbl>
##  1 01    144488  2.52        83.2       16.8
```

```
##   2 02     222818   3.89         78.9        21.1
##   3 03     301185   5.26         72.7        27.3
##   4 04     414358   7.24         68.3        31.7
##   5 05     608619 10.6           62.1        37.9
##   6 06     709426 12.4           57.7        42.3
##   7 07     766183 13.4           56.9        43.1
##   8 08     770179 13.5           59.8        40.2
##   9 09     665313 11.6           60.8        39.2
## 10 10     536362   9.37         67.1        32.9
## 11 11     362019   6.32         72.9        27.1
## 12 12     223779   3.91         77.0        23.0
```

- From the tibble, we can infer the distribution will take a bell shaped curve. Our counts are lowest towards the winter months and highest during the summer months.
- We see a larger percentage makeup of members during fall and winter. And the gap gradually decreases the closer we are to spring and summer months.

```
#Visualizing rides per month by customer type
cyclistic_data %>%
  ggplot(aes(month, fill = member_casual)) +
    geom_bar(position = 'dodge') +
    labs(x = "Month", y = "Total trips", title = "Total rides per month by cu
stomer type", fill = "Customer type")
```



- It would be possible to make the assumption that the large between members and casual riders is created out of the necessity for the bikes. We may infer that casual riders are using the bikes more so for leisure, so it would be natural for them to not want to ride during the cold winter months.
- Although the winter months have the lowest percentage of bike rides of the year (months 11 to 02, totaling 16.65% of rides), members make up ~75% of ridership between those months. We may further support the inference that a portion of members rely on the bikeshare for their daily commute.
- Given the previous information and now the plot, we can infer that rides follow a seasonal pattern, with more people opting to ride bikes during the warmer months of the year.

```
#Export data to local drive for Tableau visualization
fwrite(cyclistic_data, "C:\\Analytics\\Capstone\\Case_Study\\output\\cyclisti
c_data.csv")
fwrite(cyclistic_data_no_outliers, "C:\\Analytics\\Capstone\\Case_Study\\outp
ut\\cyclistic_data_no_outliers.csv")
```

Please click here to view my Tableau dashboard for this project.

Share

Summarize important findings.

What we gathered from the data:

- 5,724,729 total rides consisting of 64.3% from members and 35.7% from casual riders.
- Average ride length was 12.23896 mins after removing outliers.
- Ridership peaks in the afternoon (4pm-6pm).
- Highest percentage of rides in the afternoon (4pm-6pm).
- Ridership spikes during the weekdays, in the morning from 6am to 8am, and in the afternoon from 4pm to 6pm.
- Ridership follows seasonal patterns, with the highest volume of rides during the Summer months (6-9) and lowest volume of rides during the Winter months (11-2).

Main differences between members and casuals:

- Casual rides averaged about 2.6 minutes longer than members.
- More variability in length of casual customer riding times.

- Length of member riding times are more constant throughout the workweek.
- The weekday ridership spikes occur at much greater magnitudes for members. **Member's have 315.6% more rides than casuals from 6am to 8am** and **123.11% more rides from 4pm to 6pm**.
- Members ride more than casuals each month.
- Member's have 231.5% more rides than casuals during the winter months (Nov, Dec, Jan, Feb), but only 42.39% higher than casuals during the Summer months (Jun, Jul, Aug, Sept).
- Members ride the most during the weekdays while casuals ride the most during the weekend.

We revisit the question, "How do annual members and casual riders use Cyclistic bikes differently?".

- We presume that annual members use bikes for commuting to their daily commitments/activities such as school or work.
- This presumption is supported by the data showing the ridership spikes for members at typical times for the start and end of a workday or school-day. Both the consistency of the the spikes and also the consistency of ride lengths throughout the workweek are also supporting factors.
- We then presume that casual riders use the bikes primarily for recreational usage.
- This is inferred from the data showing the high volume of rides on the weekends, the more variable ride lengths, and the assumption of less dependency for the bikes as a means of transportation during the winter months.

Act

Finally, we make recommendations to develop the strategy to convert casual riders into annual members for the marketing team.

1. Prioritize ad slots and allocate budget towards the Summer months and perhaps even late Spring and early Fall if the weather is warm enough. Any marketing strategies should be implemented during these times as these are the peak ridership months. The Summer months would be the most effective time of the year to push for casual rider to convert to annual members.
2. Develop a campaign to advertise Cyclistic bikeshare as a reliable, cheap, and convenient way to commute to work during the week. An example of an ad that can showcase these qualities is an ad where an actor gets ready for work and is clearly not in a rush, strolls to a Cyclistic bike station, easily rents a bike using their pass,

and rides past people stuck in traffic in their cars who are clearly stressed about the time, and ends with them happily going into office with time to spare.

3. Introduce a ‘Weekender’ pass as a lower tier subscription. The pass can give unlimited access to the bikes during the weekend with a set amount of “free” weekday passes as an incentive to subscribe. At a competitive price point to purchasing two full day passes, the ‘Weekender’ pass can become the first step in familiarizing casual riders to Cyclistic subscriptions and eventually converting them to full annual members.

4. The main difference between casual and members is the fact casuals do not ride as much during the weekdays. To bridge this gap, promotions can be offered to casual riders during the weekdays as an incentive. These promotions can range from discounted pricing, free rides, or any other type of bonus with the end goal of getting casual riders to ride during the weekday more frequently. Creating a habit for casual riders to use the bikes during the weekdays may eventually convert them into annual members.

Conclusion

The certificate has taught me a lot and I thoroughly enjoyed putting what I learned about R and data analysis into practice. It was a refreshing challenge, and I found myself enjoying trying to find different ways and angles I can manipulate and dive into the data more.

Thank you very much for reading!