Cyclistic-Bike-Casestudy

Jalen Souksamlane

9/22/2021

Introduction

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

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

Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, they believe that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, they believe there is a very good chance to convert casual riders into members. With this in mind, Cyclistic's finance analysts have a clear goal: Design marketing strategies aimed at converting casual riders into annual members

In order to do this, one question that may be asked is: **How do annual members and casual riders use Cyclistic bikes differently?**

Data

Motivate International Inc. ("Motivate") operates the City of Chicago's ("City") Divvy bicycle sharing service. Motivate and the City are committed to supporting bicycling as an alternative transportation option. As part of that commitment, the City permits Motivate to make certain Divvy system data owned by the City available to the public. Cyclistic is a fictional company, but for the purposes of this case study, the datasets from Motivate are appropriate in finding an answer to the question.

The data from this case study is provided by Motivate and will be of the last 12 months, one dataset for each month. The variables for each dataset include 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, member_casual.

```r
# reading in and renaming the data
aug2020 <- read.csv("~/Programming/Divvy Bike Project/data/202008-divvy-tripdata.csv")
sep2020 <- read.csv("~/Programming/Divvy Bike Project/data/202009-divvy-tripdata.csv")
oct2020 <- read.csv("~/Programming/Divvy Bike Project/data/202010-divvy-tripdata.csv")
nov2020 <- read.csv("~/Programming/Divvy Bike Project/data/202011-divvy-tripdata.csv")
dec2020 <- read.csv("~/Programming/Divvy Bike Project/data/202012-divvy-tripdata.csv")
jan2021 <- read.csv("~/Programming/Divvy Bike Project/data/202101-divvy-tripdata.csv")
feb2021 <- read.csv("~/Programming/Divvy Bike Project/data/202102-divvy-tripdata.csv")
mar2021 <- read.csv("~/Programming/Divvy Bike Project/data/202103-divvy-tripdata.csv")
```

```
apr2021 <- read.csv("~/Programming/Divvy Bike Project/data/202104-divvy-tripdata.csv")
may2021 <- read.csv("~/Programming/Divvy Bike Project/data/202105-divvy-tripdata.csv")
jun2021 <- read.csv("~/Programming/Divvy Bike Project/data/202106-divvy-tripdata.csv")
jul2021 <- read.csv("~/Programming/Divvy Bike Project/data/202107-divvy-tripdata.csv")
```

Firstly, creating a data frame with all the data within it would make cleaning and transforming easier in the future.

```
# adding all data sets to a list
data <- list(aug2020, sep2020, oct2020, nov2020, dec2020, jan2021,
    feb2021, mar2021, apr2021, may2021, jun2021, jul2021)

# checking the strucure of each data set before combining
# into one
for (x in data) {
    print(str(x))
}
```

It seems that from December onwards, the start_station_id and the end_station_id are factors instead of integers. In order to combine all the data into one frame, this must be consistent throughout all of the data.

```
# changing data type from factor to integer
dec2020 <- dec2020 %>%
    mutate(start_station_id = as.numeric(start_station_id), end_station_id = as.numeric(end_station_id)
jan2021 <- jan2021 %>%
    mutate(start_station_id = as.numeric(start_station_id), end_station_id = as.numeric(end_station_id)
feb2021 <- feb2021 %>%
    mutate(start_station_id = as.numeric(start_station_id), end_station_id = as.numeric(end_station_id)
mar2021 <- mar2021 %>%
    mutate(start_station_id = as.numeric(start_station_id), end_station_id = as.numeric(end_station_id)
apr2021 <- apr2021 %>%
    mutate(start_station_id = as.numeric(start_station_id), end_station_id = as.numeric(end_station_id)
may2021 <- may2021 %>%
    mutate(start_station_id = as.numeric(start_station_id), end_station_id = as.numeric(end_station_id)
jun2021 <- jun2021 %>%
    mutate(start_station_id = as.numeric(start_station_id), end_station_id = as.numeric(end_station_id)
jul2021 <- jul2021 %>%
    mutate(start_station_id = as.numeric(start_station_id), end_station_id = as.numeric(end_station_id)

# creating singular data frame
clean_data <- rbind(aug2020, sep2020, oct2020, nov2020, dec2020,
    jan2021, feb2021, mar2021, apr2021, may2021, jun2021, jul2021)
```

The removal of unnecessary data is essential. This will reduce the amount of data that needs to be prepared and cleaned in order to move on to the next step.

The variables that will be removed are start_lat, start_lng, end_lat, end_lng. These variables are being removed because they hold no significance to this analysis. Latitude and Longitude are indicators of distance traveled, but in this instance, if the user were to start and end in the same spot, the difference between the latitude and longitude would be 0. Therefore, in this dataset, latitude and longitude are not good indicators.

```
clean_data_1 <- subset(clean_data, select = (-c(start_lat, end_lat,
    start_lng, end_lng)))
```

The removal of n/a values or zeroes will also help declutter the dataset.

```r
# filling in empty factors with na
clean_data_1 <- clean_data_1 %>%
    mutate_if(is.factor, na_if, y = "")

# checking columns for n/a values
apply(clean_data_1, 2, function(x) any(is.na(x)))
```

```
##            ride_id      rideable_type         started_at           ended_at
##              FALSE              FALSE              FALSE              FALSE
## start_station_name   start_station_id   end_station_name     end_station_id
##               TRUE               TRUE               TRUE               TRUE
##      member_casual
##              FALSE
```

```r
# removing n/a values
clean_data_2 <- na.omit(clean_data_1)

# number of observations removed
print((dim(clean_data_1) - dim(clean_data_2))[1])
```

```
## [1] 563682
```

Upon further inspection of the data, it appears that there is test data in the dataset. Removal of these observations will allow us to only analyze rides with actual members.

```r
clean_data_3 <- clean_data_2[!(clean_data_2$start_station_id ==
    676), ]
```

In order to perform calculations on the ride length, we must split the date and time of each ride.

```r
# changing factors to strings
clean_data_4 <- clean_data_3 %>%
    mutate(started_at = as.character(started_at), ended_at = as.character(ended_at))

# splitting started_at into two different variables,
# start_date and start_time
start <- str_split_fixed(clean_data_4$started_at, " ", 2)
start_date <- as.Date(start[, 1], "%Y-%m-%d")
start_time <- chron(times = start[, 2])

# splitting ended_at into two different variables,
# start_date and start_time
end <- str_split_fixed(clean_data_4$ended_at, " ", 2)
end_date <- as.Date(end[, 1], "%Y-%m-%d")
end_time <- chron(times = end[, 2])

# adding newly created columns to dataframe
clean_data_4$start_date <- start_date
clean_data_4$start_time <- start_time
clean_data_4$end_date <- end_date
clean_data_4$end_time <- end_time

# removing old columns
clean_data_5 <- subset(clean_data_4, select = (-c(started_at,
    ended_at)))
```

```r
# creating a column for ride length
ride_length <- abs(clean_data_5$start_time - clean_data_5$end_time)
clean_data_5$ride_length <- ride_length

# creating a column for day of the week
clean_data_5$weekday <- weekdays(clean_data_5$start_date)

# fixing ordering of weekdays
clean_data_5$weekday <- ordered(clean_data_5$weekday, levels = c("Sunday",
    "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))

# creating a column for month
clean_data_5$month <- month(ymd(clean_data_5$start_date), label = TRUE)
```

Finally, we will remove more columns that will not be used in the analysis process. These columns are ride_id, start_station_id, and end_station_id.

```r
final_data <- subset(clean_data_5, select = -c(ride_id, start_station_id,
    end_station_id))

# final dimensions of dataset
dim(final_data)
```

```
## [1] 4159416      11
```

Analysis

Descriptive Analysis on ride length

```r
# mean, median, max, and min of ride_length
summary(final_data$ride_length)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 00:00:00 00:07:28 00:13:22 00:30:07 00:24:30 23:59:26
```

```r
# comparing the average ride_length between riders
aggregate(final_data$ride_length ~ final_data$member_casual,
    FUN = mean)
```

```
##   final_data$member_casual final_data$ride_length
## 1                   casual               00:46:23
## 2                   member               00:17:25
```

```r
# comparing the median ride_length between riders
aggregate(final_data$ride_length ~ final_data$member_casual,
    FUN = median)
```

```
##   final_data$member_casual final_data$ride_length
## 1                   casual               00:18:42
## 2                   member               00:10:33
```

```r
# comparing the max ride_length bewteen riders
aggregate(final_data$ride_length ~ final_data$member_casual,
    FUN = max)
```

```
##   final_data$member_casual final_data$ride_length
## 1                   casual               23:59:26
## 2                   member               23:59:13
```

```r
# comparing the min ride_length between riders
aggregate(final_data$ride_length ~ final_data$member_casual,
    FUN = min)
```

```
##   final_data$member_casual final_data$ride_length
## 1                   casual               00:00:00
## 2                   member               00:00:00
```

```r
# comparing the average ride_length between riders by
# weekdays
aggregate(final_data$ride_length ~ final_data$member_casual +
    final_data$weekday, FUN = mean)
```

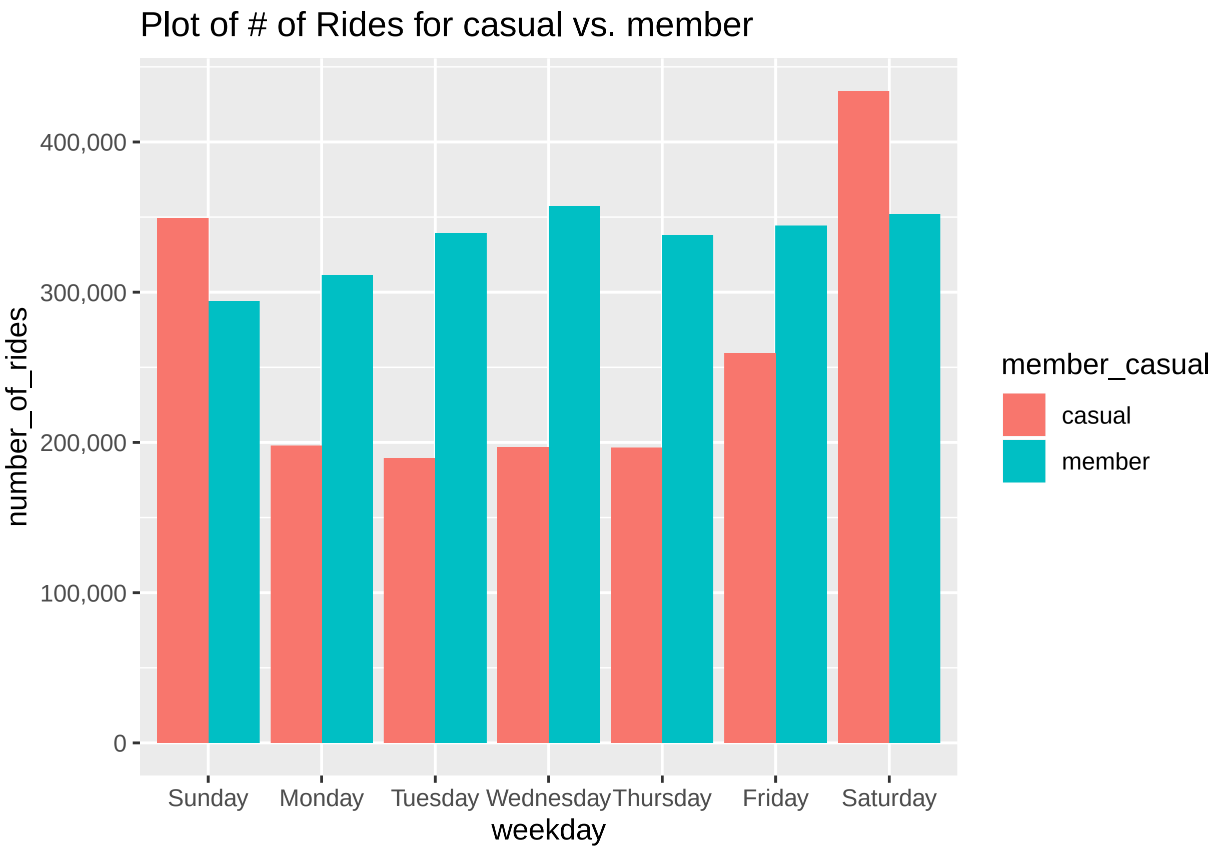
```
##    final_data$member_casual final_data$weekday final_data$ride_length
## 1                    casual             Sunday               00:45:10
## 2                    member             Sunday               00:18:02
## 3                    casual             Monday               00:42:15
## 4                    member             Monday               00:15:24
## 5                    casual            Tuesday               00:40:02
## 6                    member            Tuesday               00:15:34
## 7                    casual          Wednesday               00:39:03
## 8                    member          Wednesday               00:15:31
## 9                    casual           Thursday               00:41:07
## 10                   member           Thursday               00:16:20
## 11                   casual             Friday               00:53:48
## 12                   member             Friday               00:19:13
## 13                   casual           Saturday               00:53:17
## 14                   member           Saturday               00:21:38
```

```r
# comparing the average ride_length between riders by month
aggregate(final_data$ride_length ~ final_data$member_casual +
    final_data$month, FUN = mean)
```

```
##    final_data$member_casual final_data$month final_data$ride_length
## 1                    casual              Jan               00:27:55
## 2                    member              Jan               00:13:45
## 3                    casual              Feb               00:37:11
## 4                    member              Feb               00:17:07
## 5                    casual              Mar               00:40:38
## 6                    member              Mar               00:15:51
## 7                    casual              Apr               00:45:11
## 8                    member              Apr               00:17:29
## 9                    casual              May               00:49:08
## 10                   member              May               00:18:09
## 11                   casual              Jun               00:49:53
## 12                   member              Jun               00:18:45
## 13                   casual              Jul               00:43:56
## 14                   member              Jul               00:18:38
## 15                   casual              Aug               00:54:36
## 16                   member              Aug               00:18:43
## 17                   casual              Sep               00:47:57
## 18                   member              Sep               00:17:40
## 19                   casual              Oct               00:36:28
## 20                   member              Oct               00:15:48
## 21                        casual              Nov                 00:34:56
## 22                        member              Nov                 00:14:38
## 23                        casual              Dec                 00:28:49
## 24                        member              Dec                 00:14:32
```

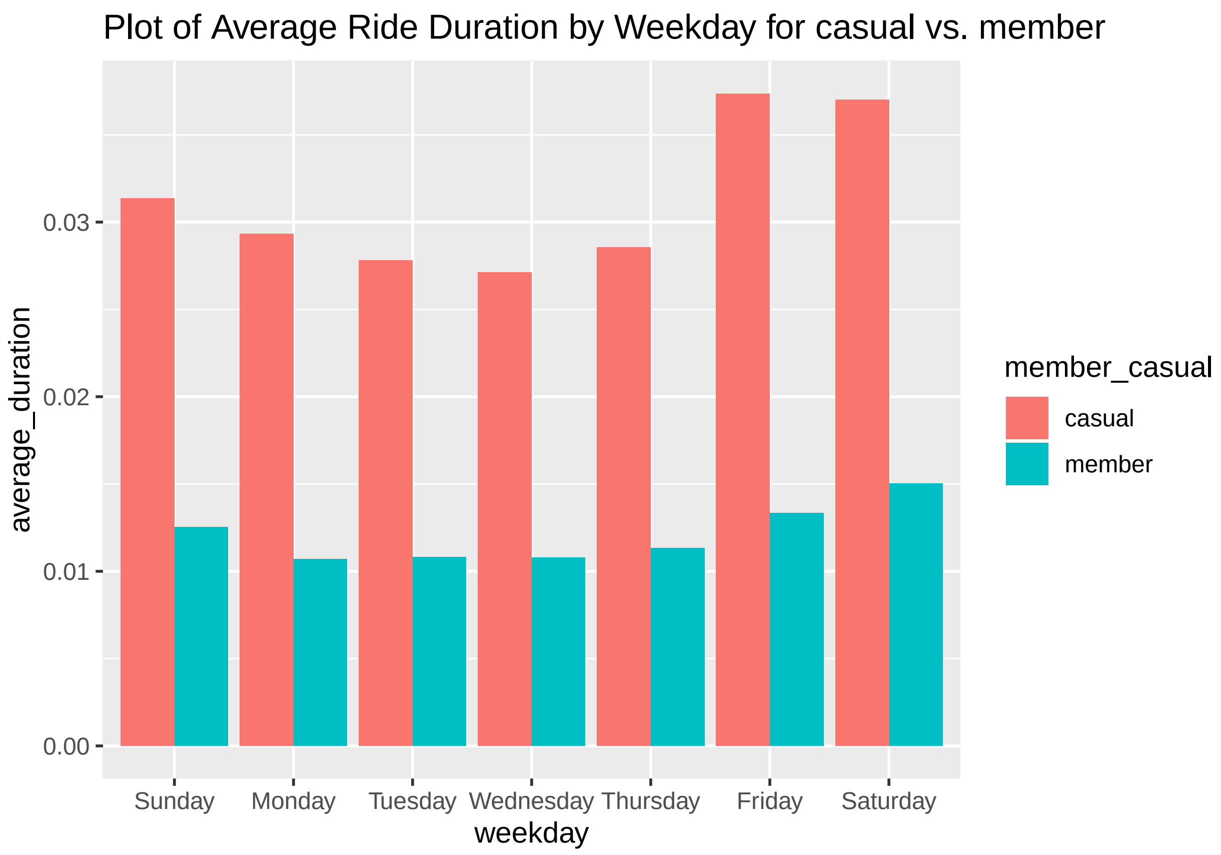
Data Visualization of ride length

```
final_data %>%
    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)) +
    geom_col(position = "dodge") + scale_y_continuous(labels = scales::comma) +
    ggtitle("Plot of # of Rides for casual vs. member")
```



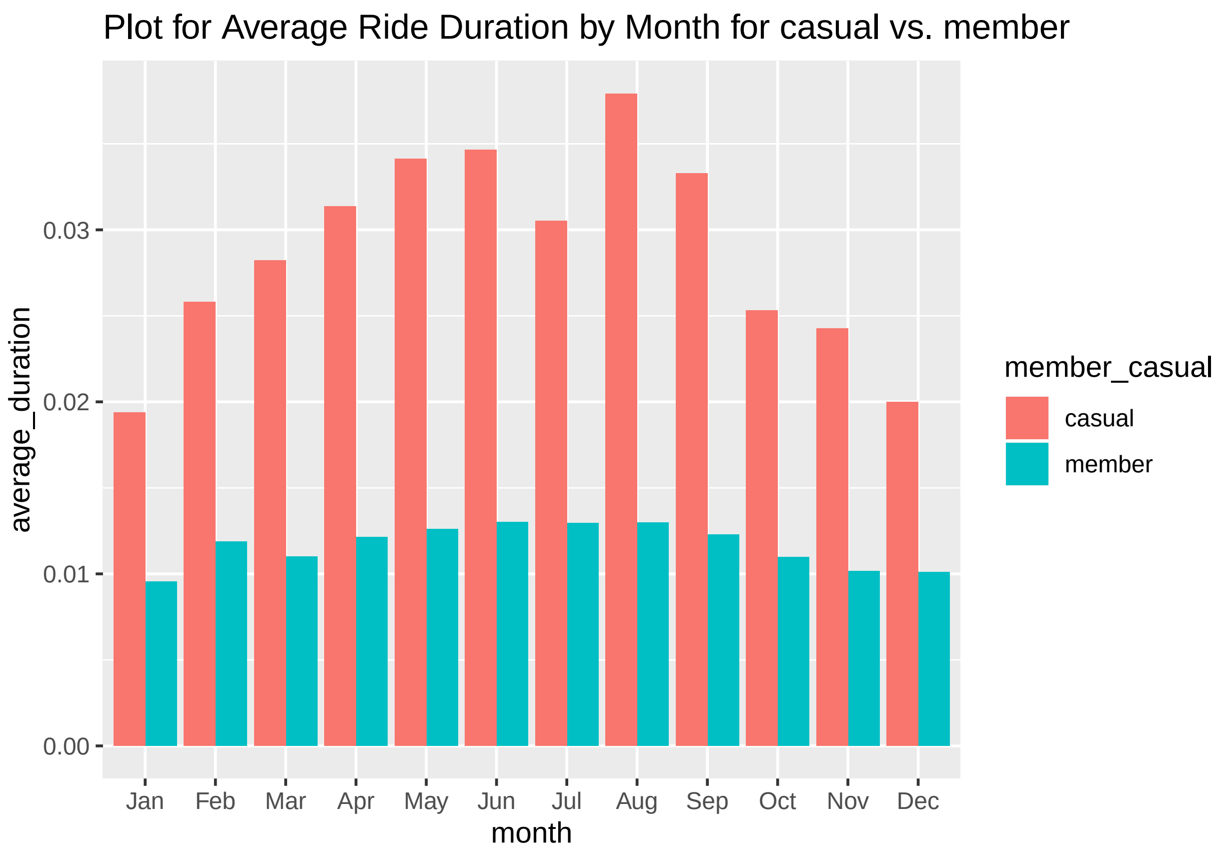
```
final_data %>%
    group_by(member_casual, weekday) %>%
    summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
    arrange(member_casual, weekday) %>%
    ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
    geom_col(position = "dodge") + ggtitle("Plot of Average Ride Duration by Weekday for casual vs. meml
```

```
## Don't know how to automatically pick scale for object of type times. Defaulting to continuous.
```



```r
final_data %>%
    group_by(member_casual, month) %>%
    summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
    arrange(member_casual, month) %>%
    ggplot(aes(x = month, y = average_duration, fill = member_casual)) +
    geom_bar(stat = "identity", position = "dodge") + ggtitle("Plot for Average Ride Duration by Month f
```

```
## Don't know how to automatically pick scale for object of type times. Defaulting to continuous.
```



Notes

- On average, casual riders ride almost twice as long as members do
- Members ride more often during the week while Casuals ride more often during the weekend
- While member ride duration is about the same throughout the year, casual ride duration is longer during spring/summer seasons, peaking in the month of August

Descriptive Analysis of rideable type

```
# total usage of bikes
summary(final_data$rideable_type)
```

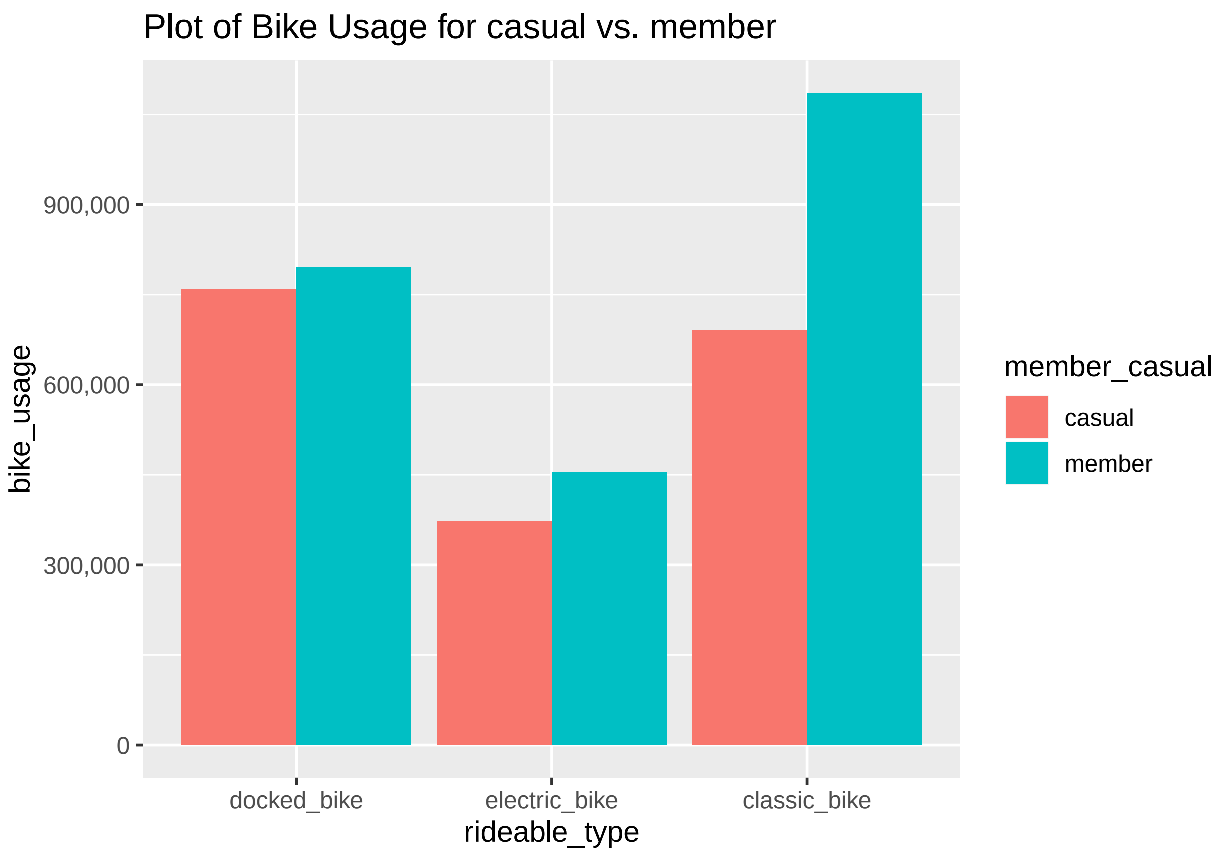
```
##   docked_bike electric_bike  classic_bike
##       1555427        827537       1776452
```

```
# comparing the usages of bikes by member type
final_data %>%
    group_by(rideable_type, member_casual) %>%
    summarize(freq = n())
```

```
## # A tibble: 6 x 3
## # Groups:   rideable_type [3]
##   rideable_type member_casual    freq
##   <fct>         <fct>           <int>
## 1 docked_bike   casual         758780
## 2 docked_bike   member         796647
## 3 electric_bike casual         373669
## 4 electric_bike member         453868
## 5 classic_bike  casual         690636
## 6 classic_bike  member        1085816
```

Data Visualization of rideable type

```
final_data %>%
    group_by(member_casual, rideable_type) %>%
    summarise(number_of_rides = n(), bike_usage = length(rideable_type)) %>%
    arrange(member_casual, rideable_type) %>%
    ggplot(aes(x = rideable_type, y = bike_usage, fill = member_casual)) +
    geom_col(position = "dodge") + scale_y_continuous(labels = scales::comma) +
    ggtitle("Plot of Bike Usage for casual vs. member")
```



Notes

- Docked bike usage is about the same between casuals and members
- Electric bike usage is slightly higher for members
- Classic bike usage is significantly higher for members

Descriptive Analysis for stations

```
final_data %>%
    group_by(start_station_name) %>%
    summarize(freq = n()) %>%
    top_n(5)
```

```
## Selecting by freq
## # A tibble: 5 x 2
##   start_station_name          freq
##   <fct>                      <int>
## 1 Clark St & Elm St          38532
## 2 Lake Shore Dr & Monroe St 41937
## 3 Michigan Ave & Oak St      37762
## 4 Streeter Dr & Grand Ave    65091
## 5 Theater on the Lake        39761
```

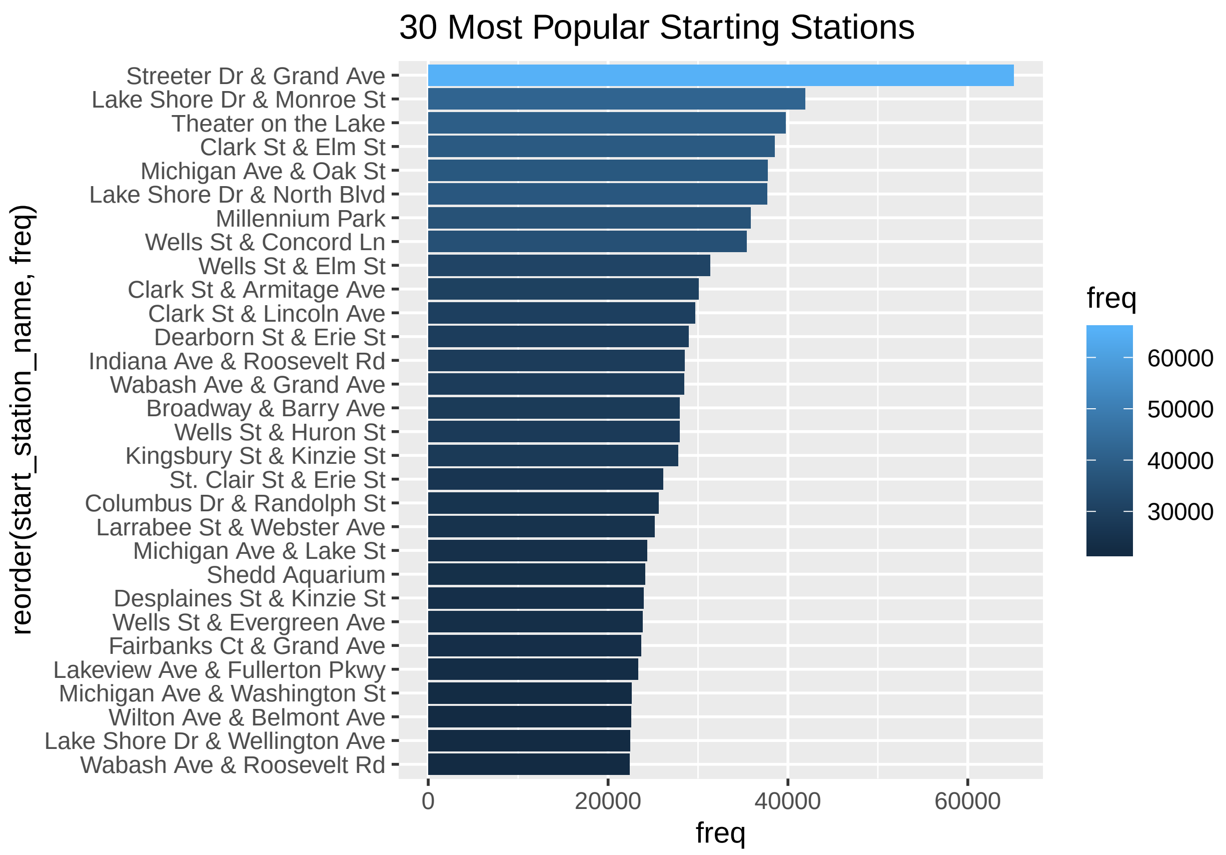
```
final_data %>%
    group_by(end_station_name) %>%
    summarize(freq = n()) %>%
    top_n(5)
```

```
## Selecting by freq
```

```
## # A tibble: 5 x 2
##   end_station_name             freq
##   <fct>                       <int>
## 1 Lake Shore Dr & Monroe St  40955
## 2 Lake Shore Dr & North Blvd 41460
## 3 Michigan Ave & Oak St      38770
## 4 Streeter Dr & Grand Ave    67781
## 5 Theater on the Lake        41178
```

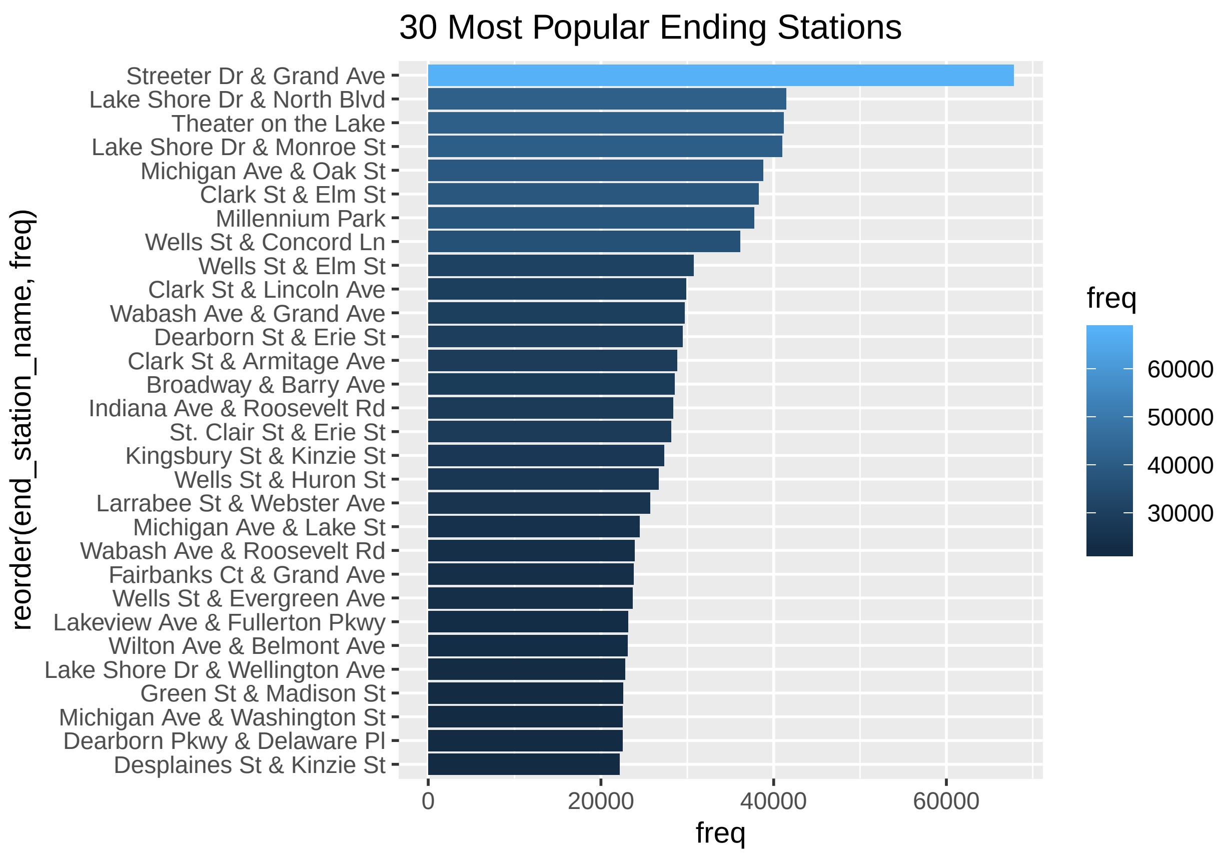
```
final_data %>%
    group_by(start_station_name) %>%
    summarize(freq = n()) %>%
    top_n(30) %>%
    ggplot(aes(reorder(start_station_name, freq), y = freq, fill = freq)) +
    geom_bar(stat = "identity", position = "dodge") + coord_flip() +
    ggtitle("30 Most Popular Starting Stations")
```

```
## Selecting by freq
```



```
final_data %>%
    group_by(end_station_name) %>%
    summarize(freq = n()) %>%
    top_n(30) %>%
    ggplot(aes(reorder(end_station_name, freq), y = freq, fill = freq)) +
    geom_bar(stat = "identity", position = "dodge") + coord_flip() +
    ggtitle("30 Most Popular Ending Stations")
```

```
## Selecting by freq
```



Notes

- Streeter Dr & Grand Ave appears to be the station most riders start and end at

Conclusions

As seen in the analysis, annual members and casual riders differ in the way they use Cyclistic bikes. For example, casual riders usually take longer rides and prefer to ride on weekends rather than weekdays. This can be due to the fact that casual riders use the bikes in their free time when they are off work. On the other hand, annual members take shorter rides and there is no significant preference for days. This can be explained by annual riders using these bikes as a way of daily transportation. Furthermore, annual members have a strong preference for classic bikes. Thus, maybe classic bikes are more suited for daily commutes.

Recommendations

- Focus advertising on the 5 most popular starting and ending stations as they will have the most traffic
- Advertise the benefits of riding a bike to work (This might help persuade casual riders to buy an annual membership)
- Focus advertising for annual membership on the weekends as more casual riders are renting bikes on those days