# Divvy Trips 2019 User Type Analysis

Spencer Miceli

6/7/2021, updated 10/30/21

## Case Study Scenario

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

Moreno has assigned you the question to answer: How do annual members and casual riders use Cyclistic bikes differently?

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

### 1. Statement of the business task

The financial analysts of the company has determined that subscription members are more profitable than customer members. Analysis on how subscription members and casual riders use Cyclistic bikes differently could lead to new insights on marketing campaigns to target casual riders and convert them to subscriptions. Lily Moreno, the marketing analytics team, and the executive team could benefit from this analysis in approving and creating new marketing strategies for Cyclistic.

### 2. Description of data sources used

The data is open public data for this case study, at link. Using this data will aid in answering how Customers and Subscribers use Cyclistic bikes differently since many data points can be classified by user type.

In this analysis, data from all four quarters of 2019 will be used. The data is organized in CSV zip files per quarter of the year. The dataset likely uses a logging device for documenting when docking and unlocking at stations. Hence, the data should not be biased. Opening and filtering each dataset in Excel, we see that key metrics are not missing values either. Therefore, we know that the data is complete and accurate, and thus reliable.

The data is original since it comes from the company's data logging system. While the data set has missing entries for birth year and gender, information such as trip_id, start_time, end_time, from_station, and to_station have no missing values.The most current Divvy_Trips data set is from 2020 Q1. Analyzing the full year of 2019 is the most current data available on the public data set. The data is used by the City of Chicago to find new ways to encourage cycling as an alternative mode of transportation to automotive vehicles.

This is open data made available by Motivate International Inc. The data allows for analysis of how customers use bikes but does not allow for using the personally identifiable information of riders. I used sorting and filtering within Microsoft Excel to find missing values. The data helps answer my business task by seeing how subscribers and riders use Cyclistic bikes. While there are not many issues with the data, the open nature of the dataset limits the depth of analysis. For example, I cannot determine whether subscribers are more likely to live near a Cyclistic service area or if casual riders purchase multiple single passes.

While I initially checked for missing values in Excel, I am using R to clean my data because I have over 3.8 million observations over four CSV files. Since Excel can only handle a million observations per spreadsheet, I chose R to clean my data instead since it can handle the additional entries. Using R ensures accuracy and completeness, since Excel would cut off hundreds of thousands of observations.

Additionally, weather data from the National Climatic Data Center will be used to supplement the Cyclistic data. Specifically, weather data from the O'Hare Airport in Chicago will be paired with the Cyclistic data to see how weather impacts the usage of Cyclistic bikes by Customers and Subscribers. This data can be found at https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/readme.txt. The original data was pulled uploaded to a local Microsoft SQL Server database and cleaned for use with the Cyclistic data.

The SQL query used for the weather data can be found inside the repo for this project.

## 3. Documentation of any cleaning or manipulation of data

Beginning the process of cleaning the data, load each CSV file into R:

```
#Install needed packages for analysis
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
#Set Working Directory to Folder with the Divvy Trips CSV Data
setwd("C:\\Users\\Spencer\\Desktop\\data\\Divvy_Trips 2019 Data")

#Load Q1-Q4 Data
Q1_df <- read_csv("Divvy_Trips_2019_Q1.csv", col_types = c("nTTnnncncccn"))
Q2_df <- read_csv("Divvy_Trips_2019_Q2.csv", col_types = c("nTTnnncncccn"))
Q3_df <- read_csv("Divvy_Trips_2019_Q3.csv", col_types = c("nTTnnncncccn"))
Q4_df <- read_csv("Divvy_Trips_2019_Q4.csv", col_types = c("nTTnnncncccn"))
```

By reading the files into R, whitespace is trimmed off automatically. I specify the column types using compact string representation so that read_csv() will give start_time and end_time the datetime data type and ensure the remaining colmuns are appropriately typed. Then, I aggregate all four quarters into one full data set for 2019. Afterwards, I remove the four quarter datasets, since we will be working with the full dataset only.

```
#Bind Q1-Q4 data into a full 2019 dataset
DT_2019_df <- as_tibble(bind_rows(Q1_df, Q2_df,Q3_df, Q4_df))

#Remove extra datasets
rm("Q1_df","Q2_df","Q3_df","Q4_df")
```

I determined that for the business task at hand, Gender and Birth Year fields did not need to be analyzed. This decision was made because there are many missing or incorrect inputs from customers and these fields do not directly contribute to answering how Customers and Subscribers differ. Thus, they were removed from the dataset. Additionally, since tripduration is given in the dataset already, end_time is unnecessary and will not be of much use for analysis. Thus, the end_time will be dropped.

```
DT_2019_df %<>% select(-c(gender, birthyear, end_time))
```

Lastly, I add new fields for date, month, and day of the week for further analysis. Should the analysis include more years than 2019, a column for year would be added as well.

```
#Adding start_date, start_time, month, and day of the week columns
DT_2019_df %<>% mutate(start_date = as_date(format(start_time, "%y-%m-%d"))) %>%
                mutate(month = as.numeric(format(start_time, "%m"))) %>%
                mutate(day_of_week = format(start_time, "%A")) %>%
                mutate(start_time = format(start_time, "%H:%M:%S"))

str(DT_2019_df)
```

```
## tibble [3,818,004 x 12] (S3: tbl_df/tbl/data.frame)
##  $ trip_id          : num [1:3818004] 21742443 21742444 21742445 21742446 21742447 ...
##  $ start_time       : chr [1:3818004] "00:04:37" "00:08:13" "00:13:23" "00:13:45" ...
##  $ bikeid           : num [1:3818004] 2167 4386 1524 252 1170 ...
```

```
##  $ tripduration     : num [1:3818004] 390 441 829 1783 364 ...
##  $ from_station_id  : num [1:3818004] 199 44 15 123 173 98 98 211 150 268 ...
##  $ from_station_name: chr [1:3818004] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ to_station_id    : num [1:3818004] 84 624 644 176 35 49 49 142 148 141 ...
##  $ to_station_name  : chr [1:3818004] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ usertype         : chr [1:3818004] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ start_date       : Date[1:3818004], format: "2019-01-01" "2019-01-01" ...
##  $ month            : num [1:3818004] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day_of_week      : chr [1:3818004] "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
```

I can verify that my data is clean and ready to analyze since all of my data contains their appropriate types. Running a quick summary on the data shows that my observations are within their expected values, except for tripduration. The irregularities in tripduration will be explored further in the analysis section.

## 4. In-depth Analysis and Supporting Visualizations

First, we see that of all rides in 2019, 2.93 million (77%) of rides were completed by subscribers and 880,000 (23%) were completed by customers.

```
table((DT_2019_df$usertype))
```

```
##
##   Customer Subscriber
##     880637    2937367
```

```
prop.table(table((DT_2019_df$usertype)))
```

```
##
##   Customer Subscriber
##  0.2306538  0.7693462
```

Checking summary statistics of Subscribers and Customers, we see that while the maximum and minimum trip duration for both users are similar, their mean and median trip durations differ.

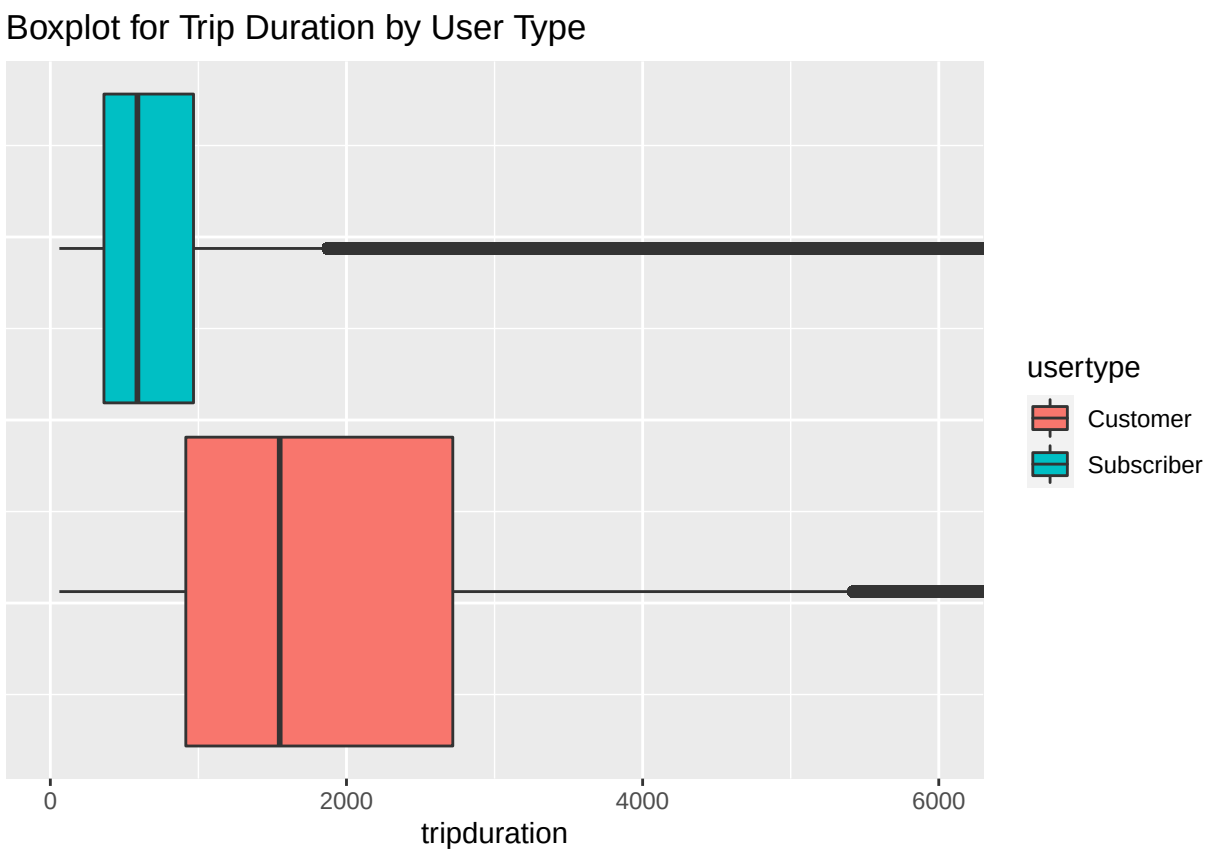
```
#Compare trip duration summary statistics between customers and subscribers
DT_2019_df %>% select(tripduration, usertype) %>% group_by(usertype) %>%
    summarize(min=min(tripduration),
              Q1 = quantile(tripduration, 1/4),
              mean=mean(tripduration),
              median = median(tripduration),
              Q3 = quantile(tripduration, 3/4),
              max=max(tripduration),
              IQR=IQR(tripduration),
              predicted_maximum = quantile(tripduration, 3/4) + 1.5*IQR,
              num_of_outliers = sum(tripduration > predicted_maximum),
              std_dev = sd(tripduration),
    )
```

```
## # A tibble: 2 x 11
##   usertype     min    Q1  mean median    Q3      max   IQR predicted_maximum
##   <chr>      <dbl> <dbl> <dbl>  <dbl> <dbl>    <dbl> <dbl>             <dbl>
```

```
## 1 Customer      61   915 3421.   1549  2718 10628400  1803             5422.
## 2 Subscriber    61   362  859.    588   967  9056633   605             1874.
## # ... with 2 more variables: num_of_outliers <int>, std_dev <dbl>
```

The accompanying boxplot for the above summary statistics:

```
DT_2019_df %>% group_by(usertype) %>% ggplot() +
  geom_boxplot(mapping = aes(x=tripduration, group = usertype, fill = usertype)) +
  coord_cartesian(xlim = c(0,6000)) +
  theme(axis.ticks.y = element_blank(),
        axis.text.y = element_blank()
  ) +
  labs(title = "Boxplot for Trip Duration by User Type")
```




The mean trip duration is much larger than the median trip duration for both Customers and Subscribers. This is due to the large number of outliers for both user types that skew the means. Thus, we will look mostly at the median trip duration for this analysis.

Next, we will look at the median trip duration for Subscribers and Customers by day of the week.

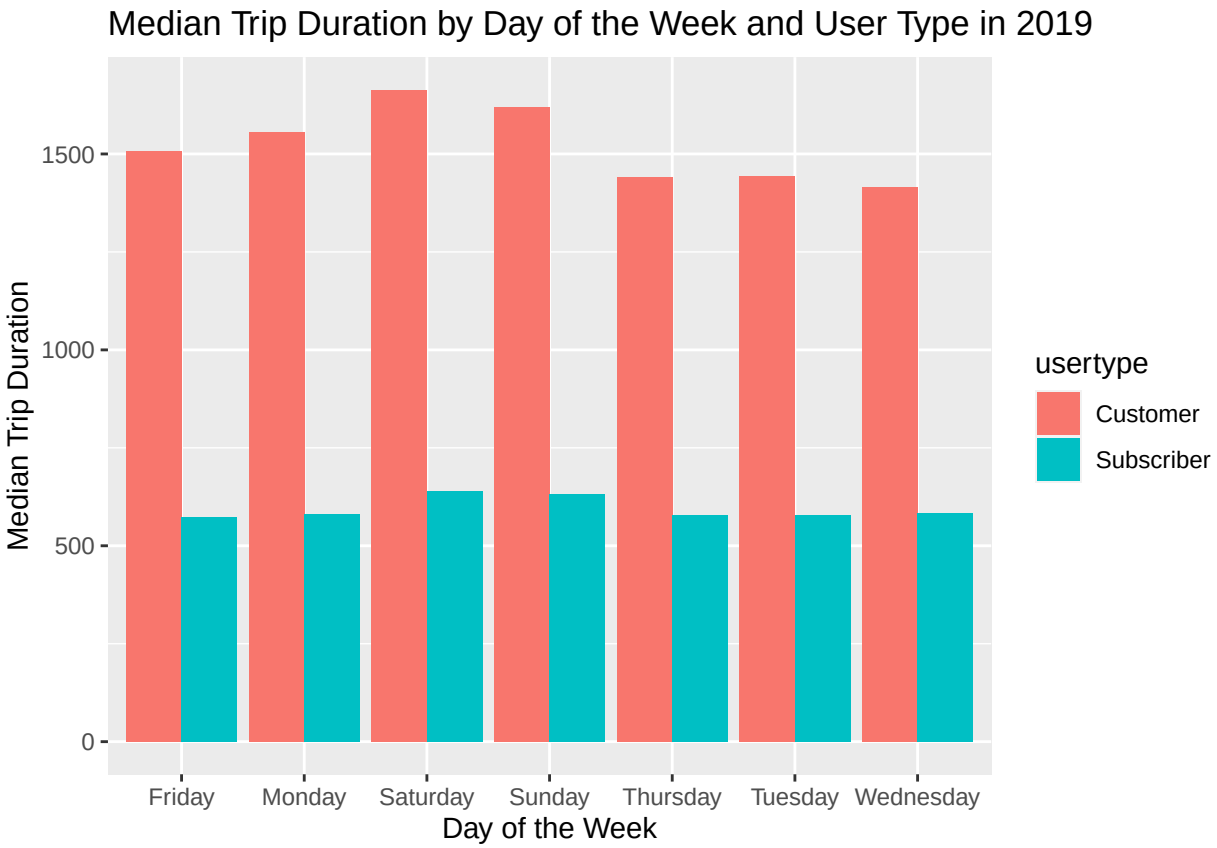
```
#Correctly order the days of the week, then check the mean trip duration by day and user type
DT_2019_df %>% group_by(usertype, day_of_week) %>%  summarize(mean=mean(tripduration), median = median(
                                                    max=max(tripduration), min=min(tripduration), IQR=IQR(
                                                    std_dev = sd(tripduration) )
```

```
## `summarise()` has grouped output by 'usertype'. You can override using the `.groups` argument.
```

```
## # A tibble: 14 x 8
## # Groups:   usertype [2]
##    usertype   day_of_week  mean median      max   min   IQR std_dev
##    <chr>      <chr>       <dbl>  <dbl>    <dbl> <dbl> <dbl>   <dbl>
##  1 Customer   Friday      3610.   1508  7939448    61 1751   64006.
##  2 Customer   Monday      3270.   1556  7247750    61 1866   50127.
##  3 Customer   Saturday    3244.   1664  8120385    61 1915   40624.
##  4 Customer   Sunday      3371.   1619  8585902    61 1896   50929.
##  5 Customer   Thursday    3597.   1440 10628400    61 1644   67909.
##  6 Customer   Tuesday     3445.   1443  7522062    61 1668.  57636.
##  7 Customer   Wednesday   3620.   1416  7606871    61 1606   68952.
##  8 Subscriber Friday       834.    572  4840300    61  579   13362.
##  9 Subscriber Monday       855.    580  8203637    61  585   18576.
## 10 Subscriber Saturday     978.    640  4809091    61  709   13766.
## 11 Subscriber Sunday       924.    632  2910775    61  708    9643.
## 12 Subscriber Thursday     827.    579  6028601    61  583   12955.
## 13 Subscriber Tuesday      849.    579  9056633    61  582   19362.
## 14 Subscriber Wednesday    828.    582  5628778    61  586   13304.
```

To supplement the data, we'll construct two visualizations:

```
## `summarise()` has grouped output by 'usertype'. You can override using the `.groups` argument.
```




We see two general trends emerge from these visualizations. First, Customers take longer trips than subscribers. Secondly, both Subscribers and Customers ride for slightly longer durations on the weekends.

Next, I analyze which stations our customers starting their trips from. To do this, first I create a new table that shows corresponds the station ID with the number of rides started at that station. Then, I run some summary statistics on these tables.

```
#Creating tables for number of Trips based on Subscribers and Customers
Sub_DT_df <- DT_2019_df %>% filter(usertype == "Subscriber") %>% select(from_station_id) %>%
  group_by(from_station_id)%>% summarize(count = n())

Cus_DT_df <- DT_2019_df %>% filter(usertype == "Customer") %>% select(from_station_id) %>%
  group_by(from_station_id)%>% summarize(count = n())

#Summary Statistics
(Sub_DT_ss <- Sub_DT_df %>% summarize(
                min=min(count),
                Q1 = quantile(count, 1/4),
                mean=mean(count),
                median = median(count),
                Q3 = quantile(count, 3/4),
                max=max(count),
                IQR=IQR(count),
                predicted_maximum = quantile(count, 3/4) + 1.5*IQR,
                num_of_outliers = sum(count > predicted_maximum),
                std_dev = sd(count)
              )
)
```

```
## # A tibble: 1 x 10
##     min    Q1  mean median    Q3   max   IQR predicted_maximum num_of_outliers
##   <int> <dbl> <dbl>  <dbl> <dbl> <int> <dbl>             <dbl>           <int>
## 1     2  511. 4784.  2630. 7144. 50575 6632.            17092.              25
## # ... with 1 more variable: std_dev <dbl>
```

```
(Cus_DT_ss <- Cus_DT_df %>% summarize(
                min=min(count),
                Q1 = quantile(count, 1/4),
                mean=mean(count),
                median = median(count),
                Q3 = quantile(count, 3/4),
                max=max(count),
                IQR=IQR(count),
                predicted_maximum = quantile(count, 3/4) + 1.5*IQR,
                num_of_outliers = sum(count > predicted_maximum),
                std_dev = sd(count)
              )
)
```

```
## # A tibble: 1 x 10
##     min    Q1  mean median    Q3   max   IQR predicted_maximum num_of_outliers
##   <int> <dbl> <dbl>  <dbl> <dbl> <int> <dbl>             <dbl>           <int>
## 1     1  150. 1434.    527  1546 53104 1396.             3640.              49
## # ... with 1 more variable: std_dev <dbl>
```

We see that our Subscribers do not seem to start from a group of stations more than others. However, for

customers, the mean is close to the 3rd. Quartile, indicating that there are stations that are receiving a lot of traffic for customers.

For this report, a "high volume station" will be any station that has an annual number of start rides that is greater than the predicted maximum, which is calculated using the formula:

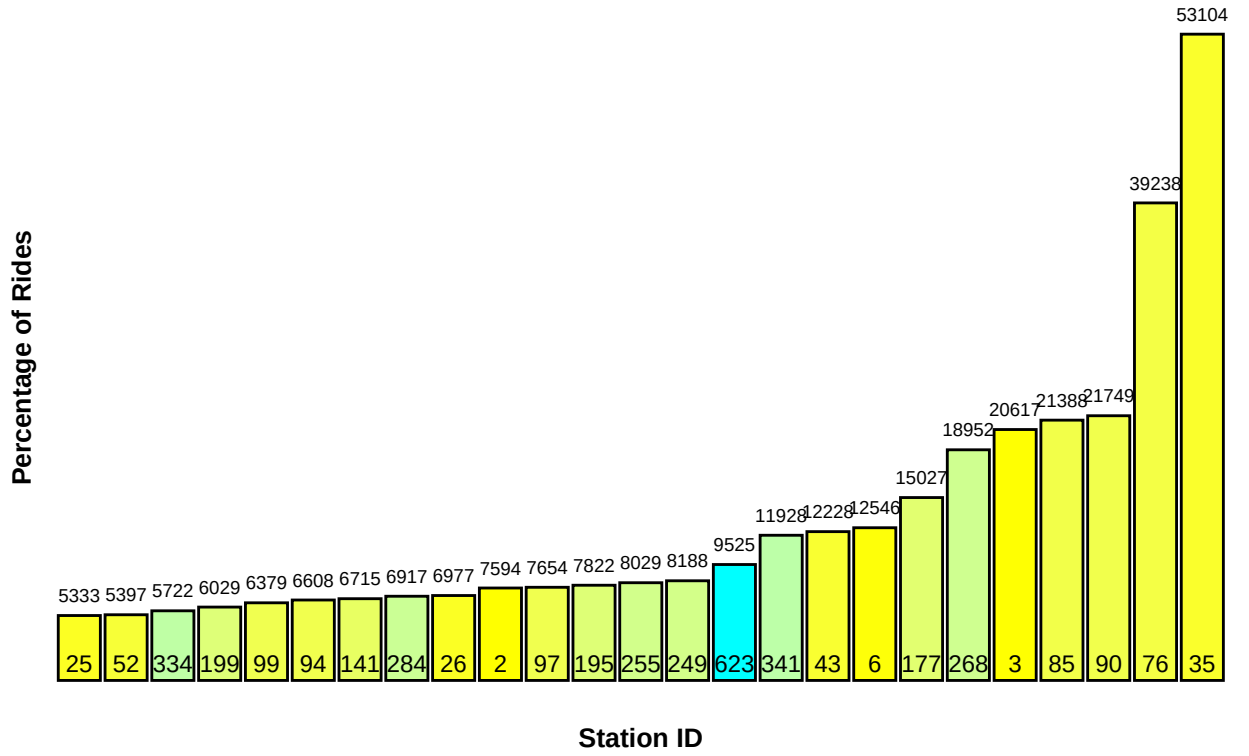
Predicted maximum = Q3 + 1.5*(Q3-Q1).

Using the quartile statistics given, and the predicted maximum formula, we can find if the percentage of high volume stations that subscribers and customers are starting at much more frequently than other stations.

```
tibble(
  Subscriber = paste(round(as.numeric(Sub_DT_df %>%
                                        filter(count > Sub_DT_ss$predicted_maximum) %>%
                                        summarize(total = sum(count))/as.numeric(Sub_DT_df %>%
                                                    summarize(total= sum(count)))
                                      )*100, 2), "%")
,


  Customer = paste(round(as.numeric(Cus_DT_df %>%
                                        filter(count > Cus_DT_ss$predicted_maximum) %>%
                                        summarize(total = sum(count))/as.numeric(Cus_DT_df %>%
                                                    summarize(total= sum(count)))
                                      )*100, 2), "%")
)
```

```
## # A tibble: 1 x 2
##   Subscriber Customer
##   <chr>      <chr>
## 1 21.47 %    49.08 %
```

We see that 49% all Customers start their rides from one of 49 high volume station, whereas only 21.5% of Subscribers start from one of 25 high volume station. Let's create an accompanying visual to illustrate.




```
tibble(
  Percentage_of_rides = paste(round(as.numeric(
                              Cus_DT_df %>% arrange(desc(count)) %>% slice(1:25) %>%
                              summarize(total_rides = sum(count)) / as.numeric(Cus_DT_df %>%
                                                    summarize(total= sum(count)))
                              )*100, 2), "%")
  ,

  Percentage_of_stations = paste(round(as.numeric(
                              Cus_DT_ss$num_of_outliers /
                              Cus_DT_df %>% select(from_station_id) %>% summarize(n = n())
                              )*100, 2), "%")
)
```

```
## # A tibble: 1 x 2
##   Percentage_of_rides Percentage_of_stations
##   <chr>               <chr>
## 1 37.66 %             7.98 %
```

```
  #percentage of customer rides started at one of the top 25 highest volume stations
```

Since displaying all 49 stations in the bar chart would be quite cluttered, the Top 25 stations for Customers are shown.

We see from this visual the station ID of the top 25 high volume stations for customer rides for 2019. Recall that 49 stations account for 49% of all rides from customers in 2019. These top 25 high volume stations

are significant since these stations account for 8% of all our stations but yet accrue 37.66% of all rides for customers. Many of these stations could be prime locations for increased and targeted marketing.

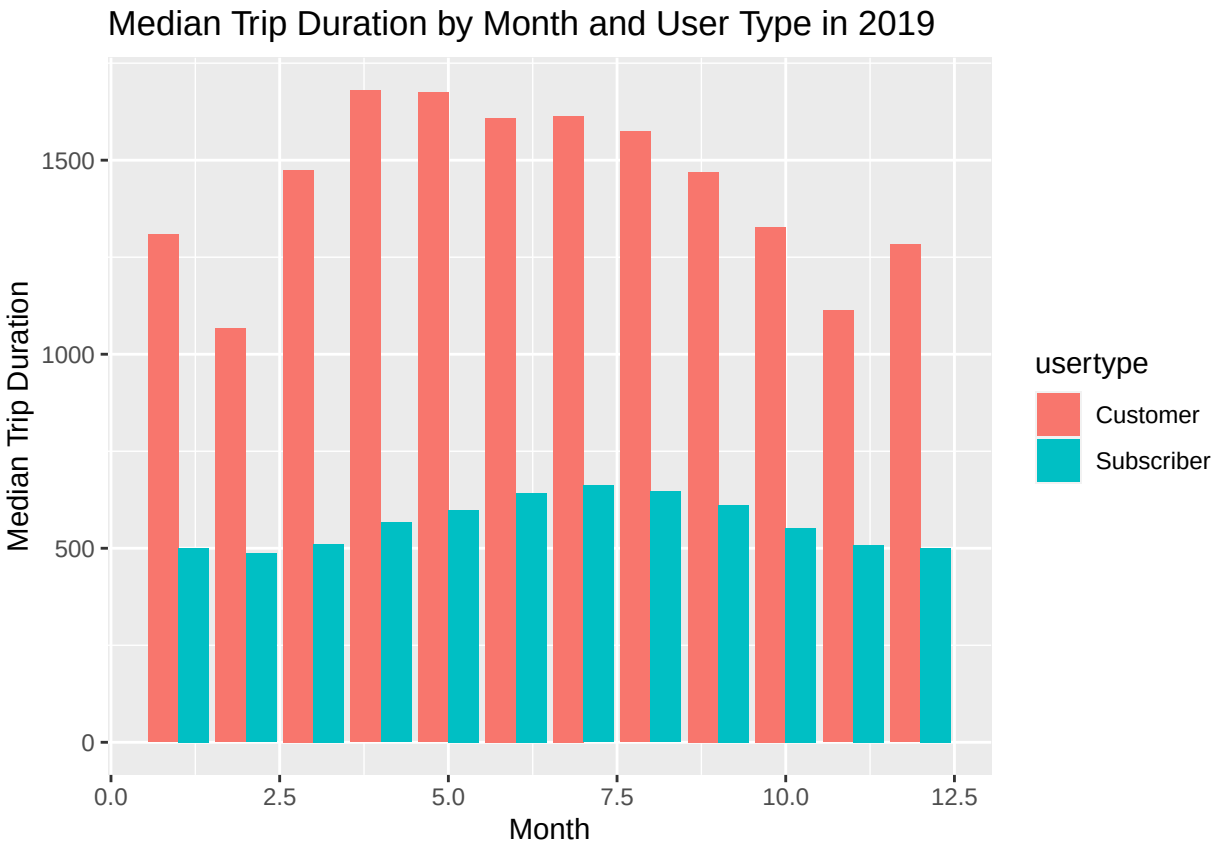
```
print( Cus_DT_df %>% arrange(desc(count)) %>% slice(1:25) %>% left_join(DT_2019_df, by = "from_station_
        distinct(from_station_name)

)
```

```
## # A tibble: 26 x 1
##    from_station_name
##    <chr>
##  1 Streeter Dr & Grand Ave
##  2 Lake Shore Dr & Monroe St
##  3 Millennium Park
##  4 Michigan Ave & Oak St
##  5 Shedd Aquarium
##  6 Lake Shore Dr & North Blvd
##  7 Theater on the Lake
##  8 Dusable Harbor
##  9 Michigan Ave & Washington St
## 10 Adler Planetarium
## # ... with 16 more rows
```

We could display a similar visual for Subscribers, but 25 high volume stations for subscribers make up 21.5% of the total number of rides. This is not as significant as Customers, which indicates that Subscribers make greater use of more stations than Customers.

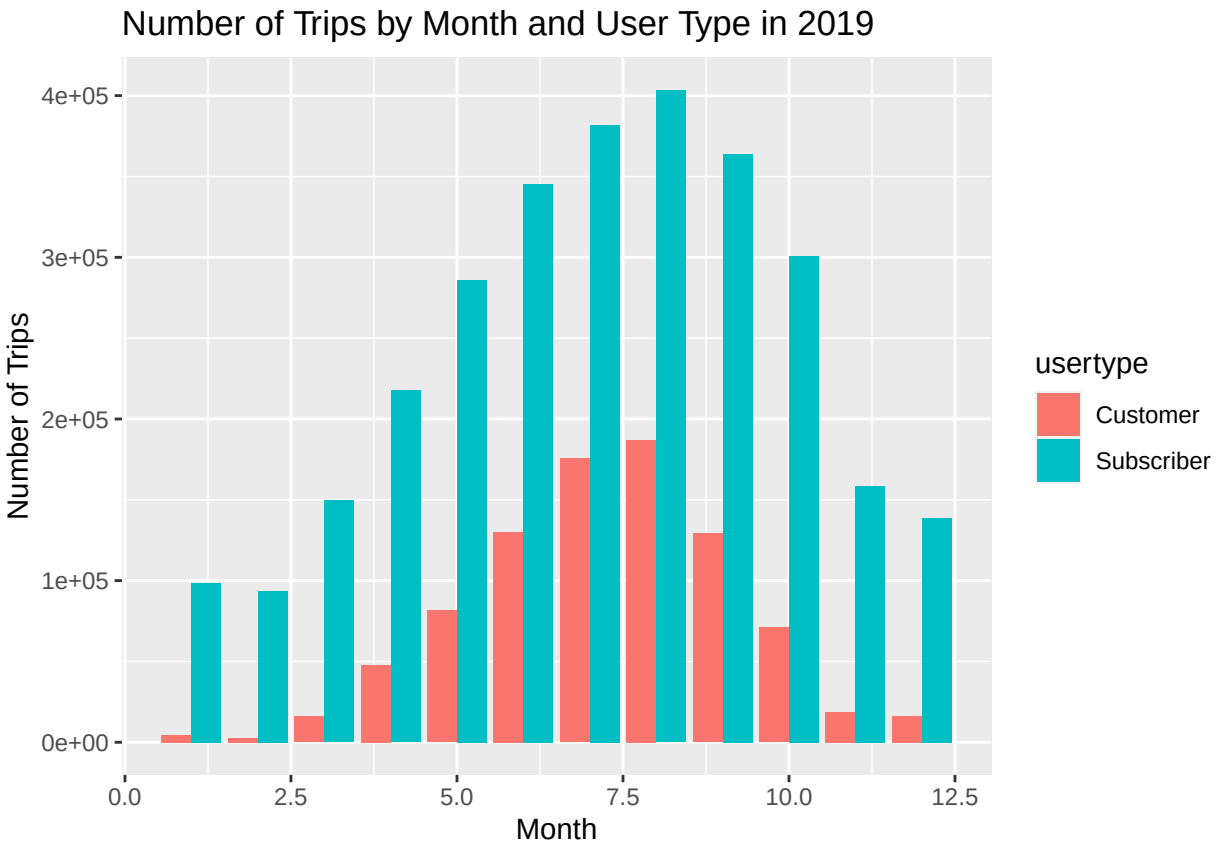
Lastly, we will visualize Customers and Subscribers median trip duration based on the month of the year.

```
## `summarise()` has grouped output by 'usertype'. You can override using the `.groups` argument.
```



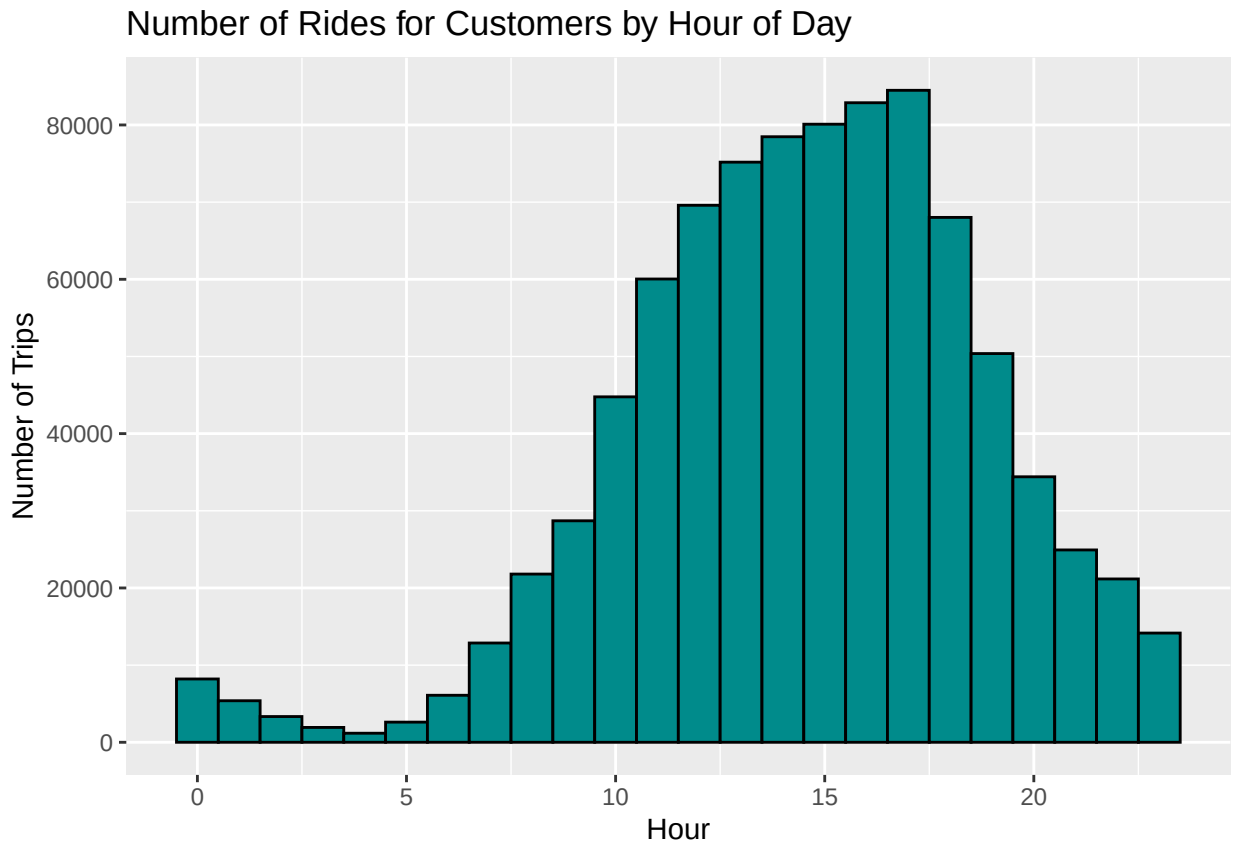

We see that Customers ride longer trips during the spring and summer months than in the fall and winter months. Subscribers follow a similar pattern, with much less variability. Next, we will see how total number of rides changes by month.

```
## `summarise()` has grouped output by 'usertype'. You can override using the `.groups` argument.
```

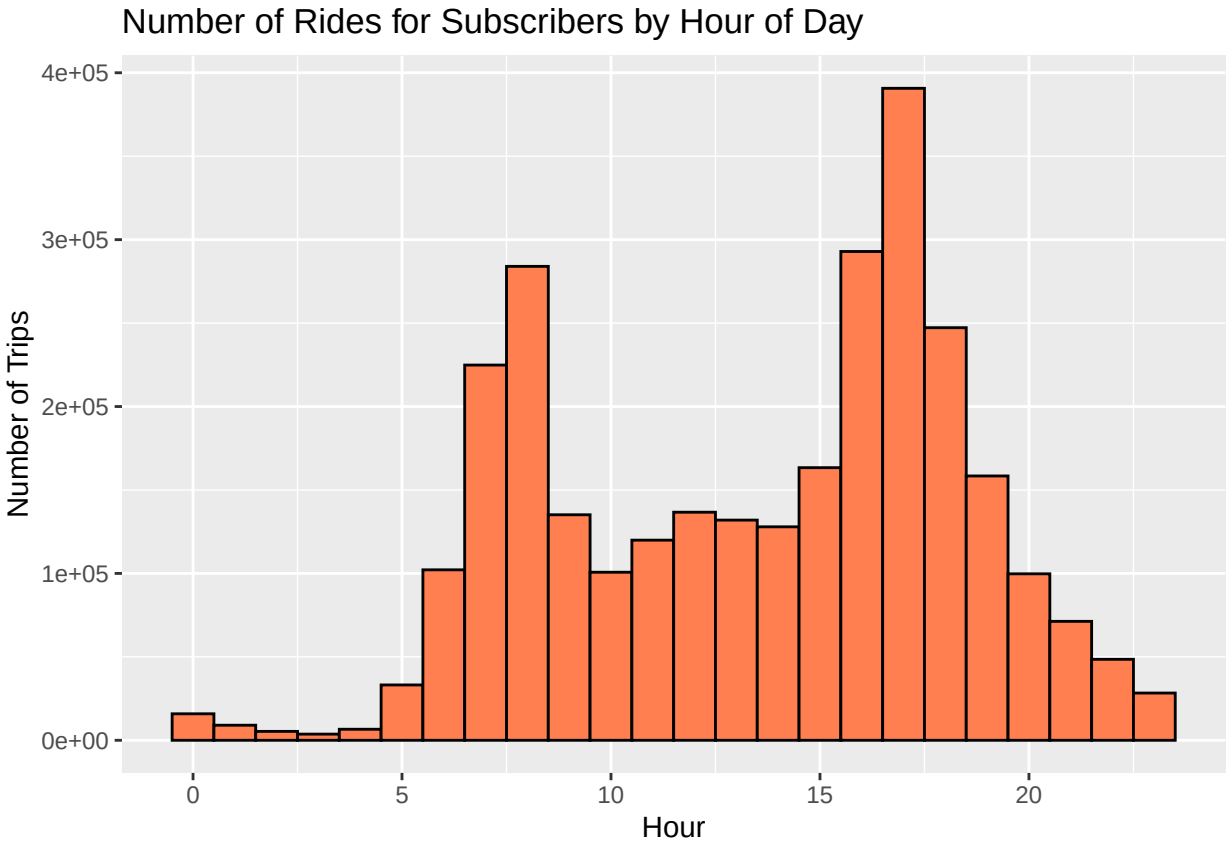



This visual shows us that Subscribers and Customers take more trips during the spring and summer months than in the fall and winter months. In fact, Customers take very few trips from November through February, whereas Subscribers still see much use of the bicycles during the winter months. Lastly, we will visualize how use of the bike share changes by time of day.

```
DT_2019_df %>% select(start_time, usertype) %>%  mutate(start_hour = hour(hms::as_hms(start_time))) %>%
  filter(usertype == "Customer") %>% ggplot() +
  geom_histogram(mapping = aes(x=start_hour), color = "black", fill = "darkcyan", bins = 24)+
  labs(title = "Number of Rides for Customers by Hour of Day",
       x = "Hour",
       y = "Number of Trips")
```




```
DT_2019_df %>% select(start_time, usertype) %>%  mutate(start_hour = hour(hms::as_hms(start_time))) %>%
  filter(usertype == "Subscriber") %>% ggplot() +
  geom_histogram(mapping = aes(x=start_hour), color = "black", fill = "coral", bins = 24)+
  labs(title = "Number of Rides for Subscribers by Hour of Day",
       x = "Hour",
       y = "Number of Trips")
```




We see that customers typically take rides between noon and early evening. On the contrary, subscribers take rides mostly during the morning and evening rush hours. This indicates that customers are taking rides for different reasons than subscribers. Subscribers use Cyclistic for commute, whereas customers use it for leisure.

We saw that Customers are less likely to take any rides during the winter months. Since customers seem to ride primarily for leisure, then weather likely impacts how customers use the bikes. Now, we'll add an additional data set from the National Climatic Data Center, taking weather data from the O'Hare airport in Chicago.

```
weather <- as_tibble(read_csv("OHare_Airport_Weather_2019.csv", col_names = TRUE))
```

```
## Rows: 365 Columns: 20
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (8): WT01, WT02, WT03, WT04, WT05, WT06, WT08, WT09
## dbl  (11): AWND, PRCP, SNOW, SNWD, TAVG, TMAX, TMIN, WDF2, WDF5, WSF2, WSF5
## date  (1): date
```

```
##
## 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.
```

```
weather_DT_df <-DT_2019_df %>%
  select(start_date, tripduration, usertype) %>%
  left_join(weather, by = c("start_date" = "date"))
```

```
bind_rows(
weather_DT_df %>% filter(PRCP < 254) %>% group_by(usertype) %>%
  summarize(condition = "Precipitation < 1 inch",
            number_of_trips = n(),
            avg_number_of_trips = n() / as.numeric(
              weather_DT_df %>% filter(PRCP < 254) %>% distinct(start_date) %>%
              summarize(number_of_days = n())
              ),
            median = median(tripduration),
            std = sd(tripduration)
            )
,
weather_DT_df %>% filter(PRCP >= 254) %>% group_by(usertype) %>%
  summarize(condition = "Precipitation >= 1 inch",
            number_of_trips = n(),
            avg_number_of_trips = n() / as.numeric(
              weather_DT_df %>% filter(PRCP > 254) %>% distinct(start_date) %>%
              summarize(number_of_days = n())
              ),
            median = median(tripduration),
            std = sd(tripduration)
            )
,

weather_DT_df %>% filter(SNOW < 12.7) %>% group_by(usertype) %>%
  summarize(condition = "Snow < 0.5 inch",
            number_of_trips = n(),
            avg_number_of_trips = n() / as.numeric(
              weather_DT_df %>% filter(SNOW < 12.7) %>% distinct(start_date) %>%
              summarize(number_of_days = n())
              ),
            median = median(tripduration),
            std = sd(tripduration)
            )
,
weather_DT_df %>% filter(SNOW >= 12.7) %>% group_by(usertype) %>%
  summarize(condition = "Snow >= 0.5 inch",
            number_of_trips = n(),
            avg_number_of_trips = n() / as.numeric(
              weather_DT_df %>% filter(SNOW > 12.7) %>% distinct(start_date) %>%
              summarize(number_of_days = n())
              ),
            median = median(tripduration),
            std = sd(tripduration)
            )
,
weather_DT_df %>% filter(AWND < 67.06) %>% group_by(usertype) %>%
  summarize(condition = "Average Wind Speed < 15mph",
            number_of_trips = n(),
```

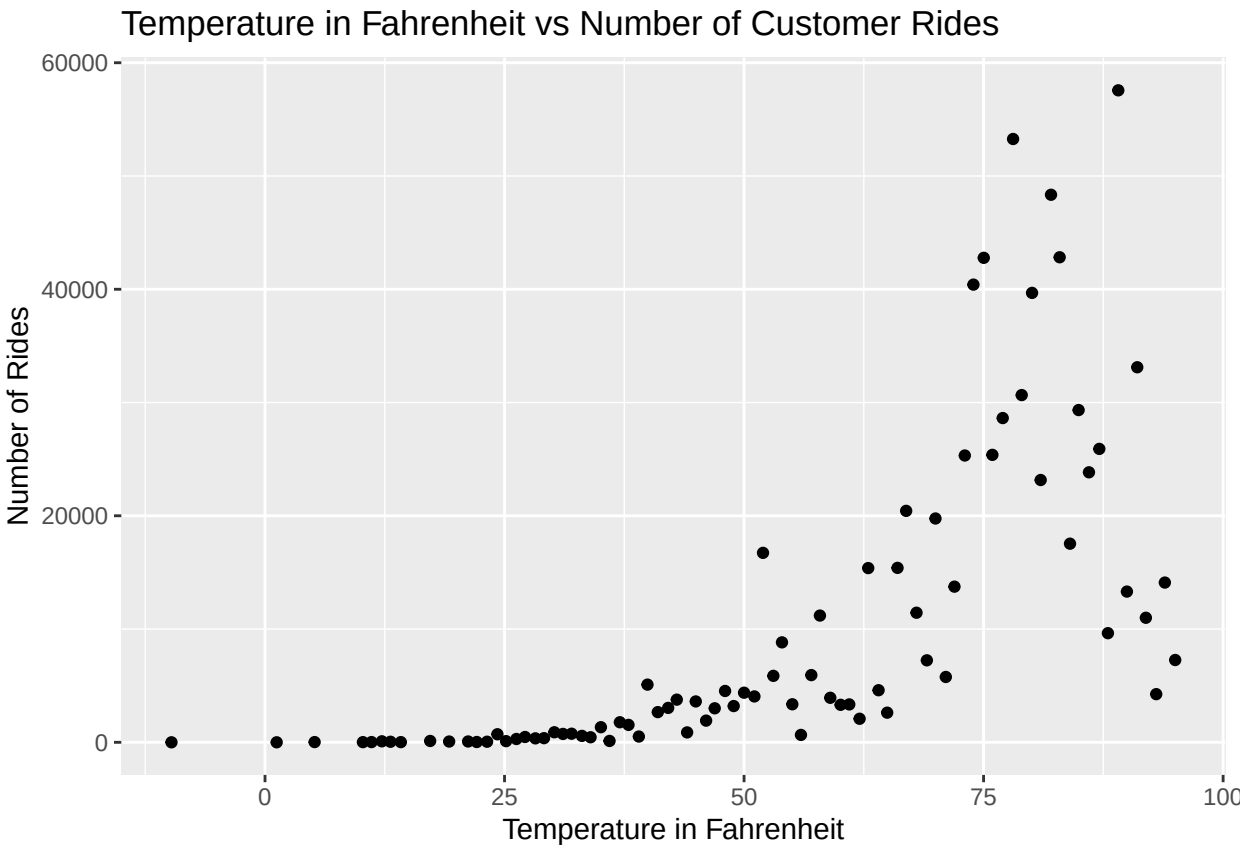
```
            avg_number_of_trips = n() / as.numeric(
              weather_DT_df %>% filter(AWND <= 67.06) %>% distinct(start_date) %>%
              summarize(number_of_days = n())
              ),
            median = median(tripduration),
            std = sd(tripduration)
            )
,
weather_DT_df %>% filter(AWND > 67.06) %>% group_by(usertype) %>%
  summarize(condition = "Average Wind Speed >= 15mph",
            number_of_trips = n(),
            avg_number_of_trips = n() / as.numeric(
              weather_DT_df %>% filter(AWND > 67.06) %>% distinct(start_date) %>%
              summarize(number_of_days = n())
              ),
            median = median(tripduration),
            std = sd(tripduration)
            )
)
```

```
## # A tibble: 12 x 6
##    usertype   condition          number_of_trips avg_number_of_tr~ median    std
##    <chr>      <chr>                        <int>             <dbl>  <dbl>  <dbl>
##  1 Customer   Precipitation < 1~          864001             2434.    1552 55557.
##  2 Subscriber Precipitation < 1~         2867917             8079.     588 15157.
##  3 Customer   Precipitation >= ~           16636             1664.    1397 59005.
##  4 Subscriber Precipitation >= ~           69450             6945      566 14615.
##  5 Customer   Snow < 0.5 inch             878957             2563.    1550 55623.
##  6 Subscriber Snow < 0.5 inch            2892038             8432.     590 15212.
##  7 Customer   Snow >= 0.5 inch              1680               76.4    971 56285.
##  8 Subscriber Snow >= 0.5 inch             45329             2060.     494  9877.
##  9 Customer   Average Wind Spee~          873795             2525.    1553 55688.
## 10 Subscriber Average Wind Spee~         2852331             8244.     591 14635.
## 11 Customer   Average Wind Spee~            6842              360.    1149 46860.
## 12 Subscriber Average Wind Spee~           85036             4476.     503 27171.
```

Here, we see that the weather conditions change how customers use Cyclistic bikes. We see that with adverse weather conditions, Customers tend to not use Cyclistic bikes. Windy and Snowy days are prime examples of how customers use the bikes less frequently. While subscribers also see a decrease in number of trips, it isnt as large as the decrease for customers.

Lastly, the number of bike rides for customers is plotted against the maximum recorded temperature to see if there is a relationship between good weather and Customer rides.

```
weather_DT_df %>% select(TMAX, usertype, tripduration) %>% mutate(temp_F = (TMAX/10)*(9/5)+32) %>%
  filter(usertype == "Customer") %>%
  group_by(temp_F) %>% summarize(number_of_rides = n()) %>%
  ggplot() + geom_point(mapping = aes(x=temp_F,y=number_of_rides)) +
  labs(title = "Temperature in Fahrenheit vs Number of Customer Rides",
       x = "Temperature in Fahrenheit",
       y = "Number of Rides")
```




We see that the number of customer rides do increase with good weather. There could be other factors at play that determine the increase in ridership though, such as more tourists riding bikes in Chicago during months with better weather.

## 5. Summary of Analysis

The most important findings from this analysis are the following.

- Customers have an average and median trip duration that is longer than the average and median trip duration of Subscribers.
- 49% of all Customer rides started at 49 high volume stations for Customers.
- Both trip duration and number of rides are higher among both Subscribers and Customers during the spring and summer months.
- Customers make far less use of the bike share from November through February.
- Adverse weather conditions greatly discourage Customers from using Cyclistic bikes.

## 6. Recommendations

Based on the summary of the analysis section, there are three recommendations that I have for Cyclistic.

1. Develop new Subscriber pricing models to appeal to Customers who use Cyclistic bikes for leisure.

2. Investigating high volume stations. 49% of customers start at 49 stations. Determining why customers tend to use these stations more than others could result in developing stronger marketing strategies.

3. Develop seasonal marketing strategies. Marketing to customers in the spring and summer could result in more subscriptions since there are more customers wanting to ride on the bike share. Additionally, marketing discounts during the fall and winter months could result in increased ridership during the fall and winter.