Cyclistic Marketing Strategy

Howard Masekesa

2025-08-05

Introduction

This report will document how Cyclistic can increase its subscriber numbers by looking into its past data, and generating insights into consumer behaviour that can be translated into increased subscriber count.

As a junior data analyst, I will begin by looking at the data and going through the steps in the data analysis process. Those steps are:

- Ask
- Prepare
- Process
- Analyze
- Share, and
- Act.

Ask

Three guiding questions to help steer the objective in the right way are:

- How do annual members and casual riders use Cyclistic bikes differently?
- Why would casual riders buy Cyclistic annual memberships?
- How can Cyclistic use digital media to influence casual riders to become members?

The question I will be answering is: How do annual members and casual riders use Cyclistic bikes differently?

In answering this question, we will observe the following:

- A description of all data sources used.
- Process used for data cleaning, wrangling, and manipulation.
- A summary of the data
- Visualizations of the data.
- A summary of my analysis
- Recommendations for increasing subscriber count.

Prepare

To prepare the site i was going to be working in, RStudio, I began by installing the relevant packages I would need:

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##
## Attaching package: 'janitor'
##
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
##
##
## i Google's Terms of Service: <https://mapsplatform.google.com>
##   Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service>
##   OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles>
## i Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

For the project, I used Cyclistic's historical trip data.

To begin the process, I imported the datasets for the Q1, Q2, Q3, Q4 2019 bike data I would be analysing (first-hand data). The data was collected, stored, and protected by Cyclistic in a secure company database.

```
data_Q1 <- read_csv("original_dataset/Divvy_Trips_2019_Q1.csv")
```

```
## Rows: 365069 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## 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.
```

```
data_Q2 <- read_csv("original_dataset/Divvy_Trips_2019_Q2.csv")
```

```
## Rows: 1108163 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): 03 - Rental Start Station Name, 02 - Rental End Station Name, User...
## dbl  (5): 01 - Rental Details Rental ID, 01 - Rental Details Bike ID, 03 - R...
## num  (1): 01 - Rental Details Duration In Seconds Uncapped
## dttm (2): 01 - Rental Details Local Start Time, 01 - Rental Details Local En...
##
## 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.
```

```
data_Q3 <- read_csv("original_dataset/Divvy_Trips_2019_Q3.csv")
```

```
## Rows: 1640718 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## 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.
```

```
data_Q4 <- read_csv("original_dataset/Divvy_Trips_2019_Q4.csv")
```

```
## Rows: 704054 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): from_station_name, to_station_name, usertype, gender
## dbl  (5): trip_id, bikeid, from_station_id, to_station_id, birthyear
## num  (1): tripduration
## dttm (2): start_time, end_time
##
## 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.
```

From the column specification, I noticed that the column names for the datasets were not all identical. I can confirm this by pulling out the column names for each dataset and calling the structure of each column. I will use this to identify if there are any inconsistencies with the column name and the formatting of each column name

```
colnames(data_Q1)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```
colnames(data_Q2)
```

```
##  [1] "01 - Rental Details Rental ID"
##  [2] "01 - Rental Details Local Start Time"
##  [3] "01 - Rental Details Local End Time"
##  [4] "01 - Rental Details Bike ID"
##  [5] "01 - Rental Details Duration In Seconds Uncapped"
##  [6] "03 - Rental Start Station ID"
##  [7] "03 - Rental Start Station Name"
##  [8] "02 - Rental End Station ID"
##  [9] "02 - Rental End Station Name"
## [10] "User Type"
## [11] "Member Gender"
## [12] "05 - Member Details Member Birthday Year"
```

```r
colnames(data_Q3)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```r
colnames(data_Q4)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```r
str(data_Q1)
str(data_Q2)
str(data_Q3)
str(data_Q4)
```

From the output, I would need to change the Q2 data column names before I can merge them into one table.

Process

The process stage will begin with changing the Q2 column names as described above, and merging all four datasets to create a yearly dataset for the number of trips taken.

```r
colnames(data_Q2) <- c("trip_id", "start_time", "end_time", "bikeid", "tripduration", "from_station_id"

trip_data <- bind_rows(data_Q1, data_Q2, data_Q3, data_Q4)
```

I can now verify the accuracy of my new table by calling its column names and structure as previously done as well as obtaining a statistical summary of the new table

```r
colnames(trip_data)
```

```
##  [1] "trip_id"           "start_time"        "end_time"
##  [4] "bikeid"            "tripduration"      "from_station_id"
##  [7] "from_station_name" "to_station_id"     "to_station_name"
## [10] "usertype"          "gender"            "birthyear"
```

```r
str(trip_data)
```

```
## spc_tbl_ [3,818,004 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ trip_id          : num [1:3818004] 21742443 21742444 21742445 21742446 21742447 ...
##  $ start_time       : POSIXct[1:3818004], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ end_time         : POSIXct[1:3818004], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ 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" ...
##  $ gender           : chr [1:3818004] "Male" "Female" "Female" "Male" ...
##  $ birthyear        : num [1:3818004] 1989 1990 1994 1993 1994 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   trip_id = col_double(),
##   ..   start_time = col_datetime(format = ""),
##   ..   end_time = col_datetime(format = ""),
##   ..   bikeid = col_double(),
##   ..   tripduration = col_number(),
##   ..   from_station_id = col_double(),
##   ..   from_station_name = col_character(),
##   ..   to_station_id = col_double(),
##   ..   to_station_name = col_character(),
##   ..   usertype = col_character(),
##   ..   gender = col_character(),
##   ..   birthyear = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(trip_data)
```

```
##     trip_id           start_time                    end_time
##  Min.   :21742443   Min.   :2019-01-01 00:04:37   Min.   :2019-01-01 00:11:07
##  1st Qu.:22873787   1st Qu.:2019-05-29 15:49:26   1st Qu.:2019-05-29 16:09:28
##  Median :23962320   Median :2019-07-25 17:50:54   Median :2019-07-25 18:12:23
##  Mean   :23915629   Mean   :2019-07-19 21:47:37   Mean   :2019-07-19 22:11:47
##  3rd Qu.:24963703   3rd Qu.:2019-09-15 06:48:05   3rd Qu.:2019-09-15 08:30:13
##  Max.   :25962904   Max.   :2019-12-31 23:57:17   Max.   :2020-01-21 13:54:35
##
##      bikeid      tripduration      from_station_id from_station_name
##  Min.   :   1   Min.   :      61   Min.   :  1.0   Length:3818004
##  1st Qu.:1727   1st Qu.:     411   1st Qu.: 77.0   Class :character
##  Median :3451   Median :     709   Median :174.0   Mode  :character
##  Mean   :3380   Mean   :    1450   Mean   :201.7
##  3rd Qu.:5046   3rd Qu.:    1283   3rd Qu.:289.0
##  Max.   :6946   Max.   :10628400   Max.   :673.0
##
##  to_station_id   to_station_name      usertype            gender
##  Min.   :  1.0   Length:3818004     Length:3818004     Length:3818004
##  1st Qu.: 77.0   Class :character   Class :character   Class :character
##  Median :174.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :202.6
##  3rd Qu.:291.0
##  Max.   :673.0
##
##    birthyear
##  Min.   :1759
##  1st Qu.:1979
##  Median :1987
##  Mean   :1984
```

```
##  3rd Qu.:1992
##  Max.   :2014
##  NA's   :538751
```

These are the descriptions of the 12 columns:

1. trip_id: Unique identifier for each trip taken
2. start_time: Time stamp indicating when the ride started.
3. end_time: Time stamp indicating when the ride ended.
4. bikeid: Unique identifier for each bike.
5. tripduration: Duration of the ride in seconds.
6. from_station_id: Unique identifier for the starting station.
7. from_station_name: Name of the station where the ride originated.
8. to_station_id: Unique identifier for the ending station.
9. to_station_name: Name of the station where the ride concluded.
10. usertype: Classification of the rider as a subscriber or a customer.
11. gender: Classification of the rider as either male or female.
12. birthyear: Year of birth of each rider.

From the new table description, 3,818,004 rows and 12 unique columns were identified, which equal to the sum of the rows from Q1, Q2, Q3, and Q4 datasets. This confirms that all rows were imported into the new table. The new table also has 12 column, confirming that no entry mistakes were made when changing the Q2 column names.

The next phase of the process step can take place, which begins by ensuring that:

- No entries have whitespaces,
- All ride id's had exactly 8 characters.
- Removing any rows with NULL values

These checks were completed in Google Sheets using data cleanup functions before merging of the datasets. However, as a final check to ensure all rows entirely with NUll, " ", and 0 values were filtered out I utilised the follwing:

```
trip_data_clean <- trip_data[!(trip_data$tripduration <= 0),]
trip_data_cleaned_2 <- trip_data_clean[!(trip_data_clean$tripduration >= 86400),]
sum(apply(trip_data_cleaned_2, 1, function(row) all(is.na(row) | row == "" | row == 0)))
```

```
## [1] 0
```

```
is_empty_row <- apply(trip_data_cleaned_2, 1, function(row) {
  row_char <- as.character(row)
  all(is.na(row) | row_char == "" | row_char == "0" | tolower(row_char) %in% c("na", "null"))
})

sum(is_empty_row)
```

```
## [1] 0
```

```
str(trip_data_cleaned_2)
```

```
## tibble [3,816,155 x 12] (S3: tbl_df/tbl/data.frame)
##  $ trip_id          : num [1:3816155] 21742443 21742444 21742445 21742446 21742447 ...
##  $ start_time       : POSIXct[1:3816155], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ end_time         : POSIXct[1:3816155], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ bikeid           : num [1:3816155] 2167 4386 1524 252 1170 ...
##  $ tripduration     : num [1:3816155] 390 441 829 1783 364 ...
##  $ from_station_id  : num [1:3816155] 199 44 15 123 173 98 98 211 150 268 ...
##  $ from_station_name: chr [1:3816155] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ to_station_id    : num [1:3816155] 84 624 644 176 35 49 49 142 148 141 ...
##  $ to_station_name  : chr [1:3816155] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ usertype         : chr [1:3816155] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender           : chr [1:3816155] "Male" "Female" "Female" "Male" ...
##  $ birthyear        : num [1:3816155] 1989 1990 1994 1993 1994 ...
```

Since all codes executed 0, I was confident that my table did not contain any empty rows.

The next phase is to remove trips taken that are negative and trips taken that are more than 24 hours:

```
trip_data_clean <- trip_data[!(trip_data$tripduration <= 0),]
```

```
trip_data_cleaned_2 <- trip_data_clean[!(trip_data_clean$tripduration >= 86400),]
```

Following these steps, it was necessary to ascertain ride information on particular days, months, and times for a more thorough analysis. Thus, it was imperative to:

- Remove duplicates,
- Create a new column for 'day_of_week'
- Create a new column for 'month'
- Create a new column for 'time'
- Create a new column for 'season'

```
trip_data_cleaned_2 <- trip_data_cleaned_2 %>%
  distinct() %>%
  mutate(tripduration = tripduration/60) %>%
  mutate(year = format(as.Date(start_time), "%Y")) %>%
  mutate(month = format(as.Date(start_time), "%B")) %>%
  mutate(day_of_week = format(as.Date(start_time), "%A")) %>%
  mutate(time = strftime(start_time, "%H")) %>%
  mutate(age=2019 - birthyear) %>%
  mutate(
    month_num = month(start_time),  # extract numeric month
    season = case_when(
      month_num %in% c(12, 1, 2)  ~ "Winter",
      month_num %in% c(3, 4, 5)   ~ "Spring",
      month_num %in% c(6, 7, 8)   ~ "Summer",
      month_num %in% c(9, 10, 11) ~ "Autumn"
    )
  ) %>%
  select(-month_num)
```

The last step on the process stage is to remove any outliers in the remaining data. This was done by using the z score to filter out tripduration data that fell outside of 3 standard deviations from the mean.

```
no_outliers <- trip_data_cleaned_2[sapply(trip_data_cleaned_2, is.numeric)]
z_scores <- scale(trip_data_cleaned_2$tripduration)
no_outliers_data <- abs(z_scores) <= 3
trip_data_cleaned_3 <- trip_data_cleaned_2[no_outliers_data, ]
```

From the processing stage, we are now able to analyse 98.9% of the original data; data that has been cleaned, manipiulated, transformed, organised, formatted, and wrangled.

Analysis

Now that the data is ready we can now analyse by:

- Conducting descriptive analysis.
- Performing calculations.
- Identifying trends and relationships
- Concluding with a summary of the analysis.

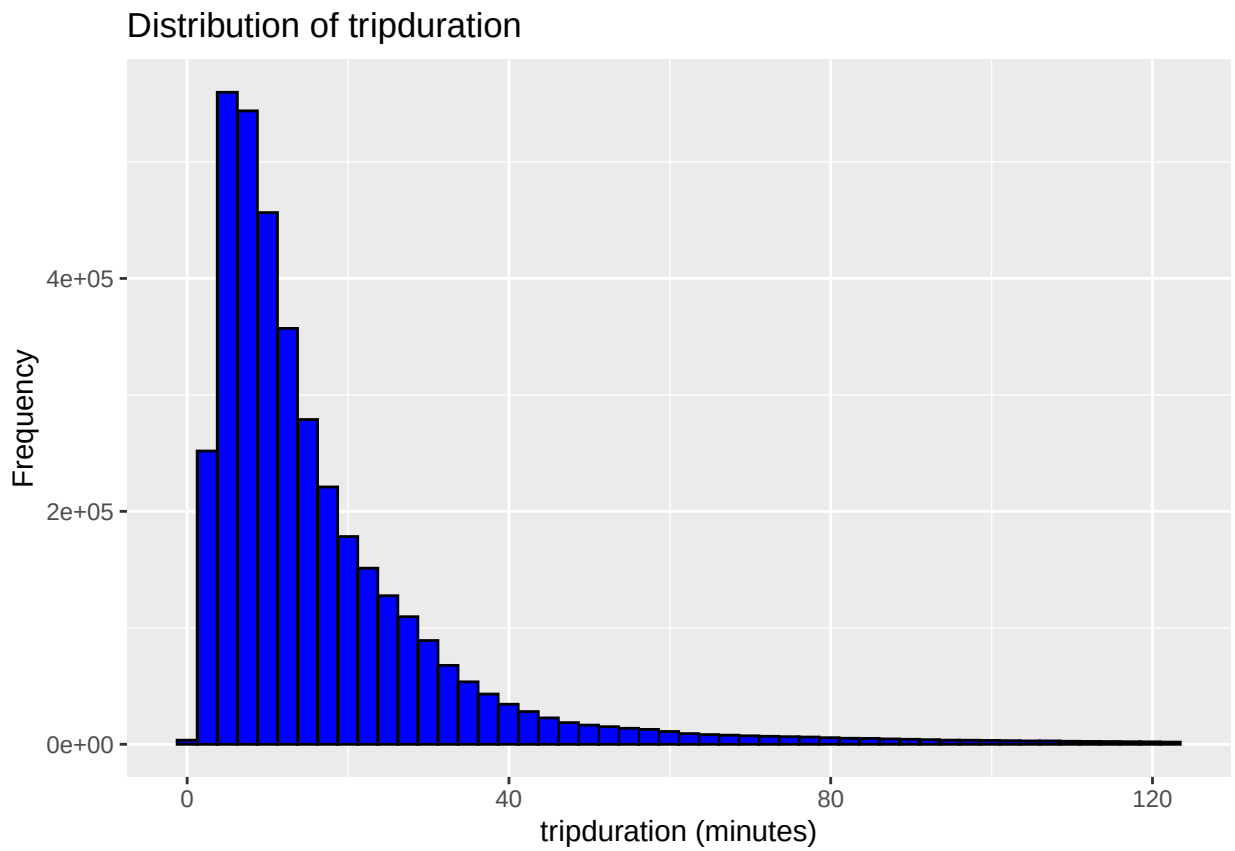
We will begin by conducting descriptive analysis and ascertain the mean, maximum, minimum, and median trip durations.

```
trip_data_cleaned_3 %>%
  summarise(average_tripduration=mean(tripduration), median_tripduration=median(tripduration),
            max_tripduration=max(tripduration), min_tripduration=min(tripduration))
```

```
## # A tibble: 1 x 4
##   average_tripduration median_tripduration max_tripduration min_tripduration
##                  <dbl>               <dbl>            <dbl>            <dbl>
## 1                 16.9                11.7             123.             1.02
```

NOTE: trip duration are in minutes

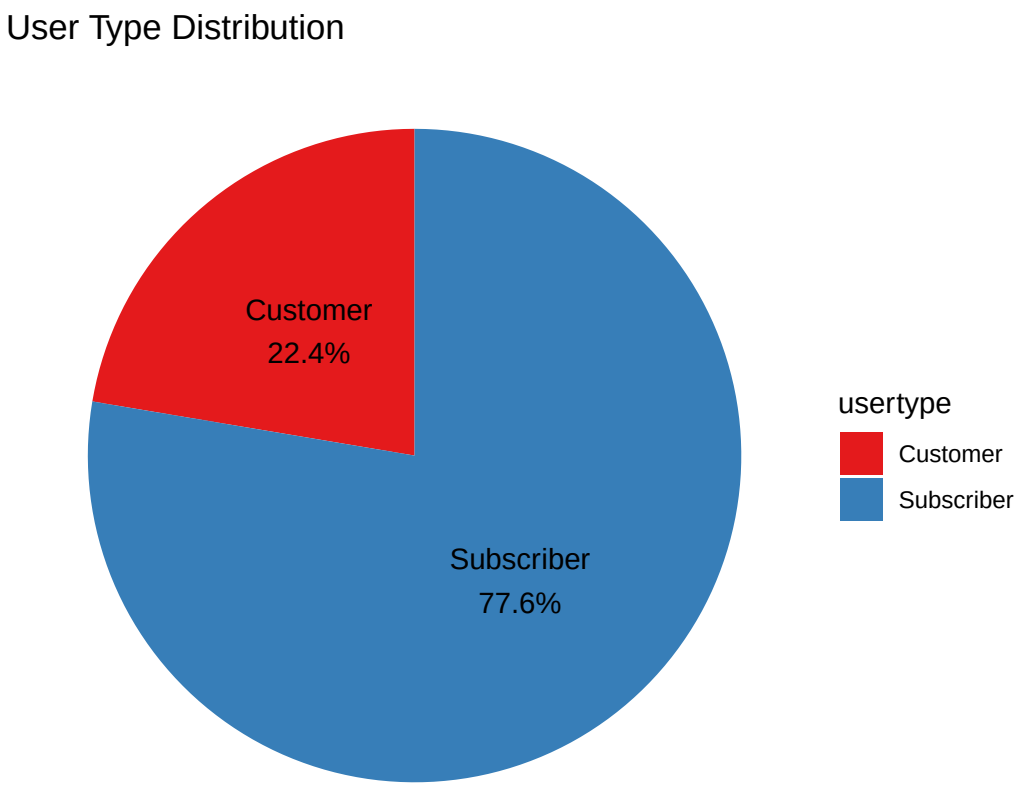
The data above shows trip duration for the year 2019. The average trip duration lasted approx. 17 minutes, while the median trip duration lasted approx. 12 minutes. This suggests a right-skewed data (as shown below), and as such, further analysis may be done in comparison with the median and not the mean.



Even after filtering out outliers, the maximum trip duration lasted just over 2 hours. Further analysis will need to be conducted to determine the the reason for this.

Ride Analysis

Usertypes For the first analysis, we will analyze the distribution of how many trips each usertype has taken.



From the pie chart, we can see that during 2019, subscribers accounted for 77.6% of all rides, while customers accounted for 22.4%.

Next we analyse the descriptive statistics by user type to gain a better understanding of the duration of trips taken by each user type

```
## # A tibble: 2 x 3
##   usertype   average_tripduration median_tripduration
##   <chr>                     <dbl>               <dbl>
## 1 Customer                   32.1                24.9
## 2 Subscriber                 12.5                 9.78
```

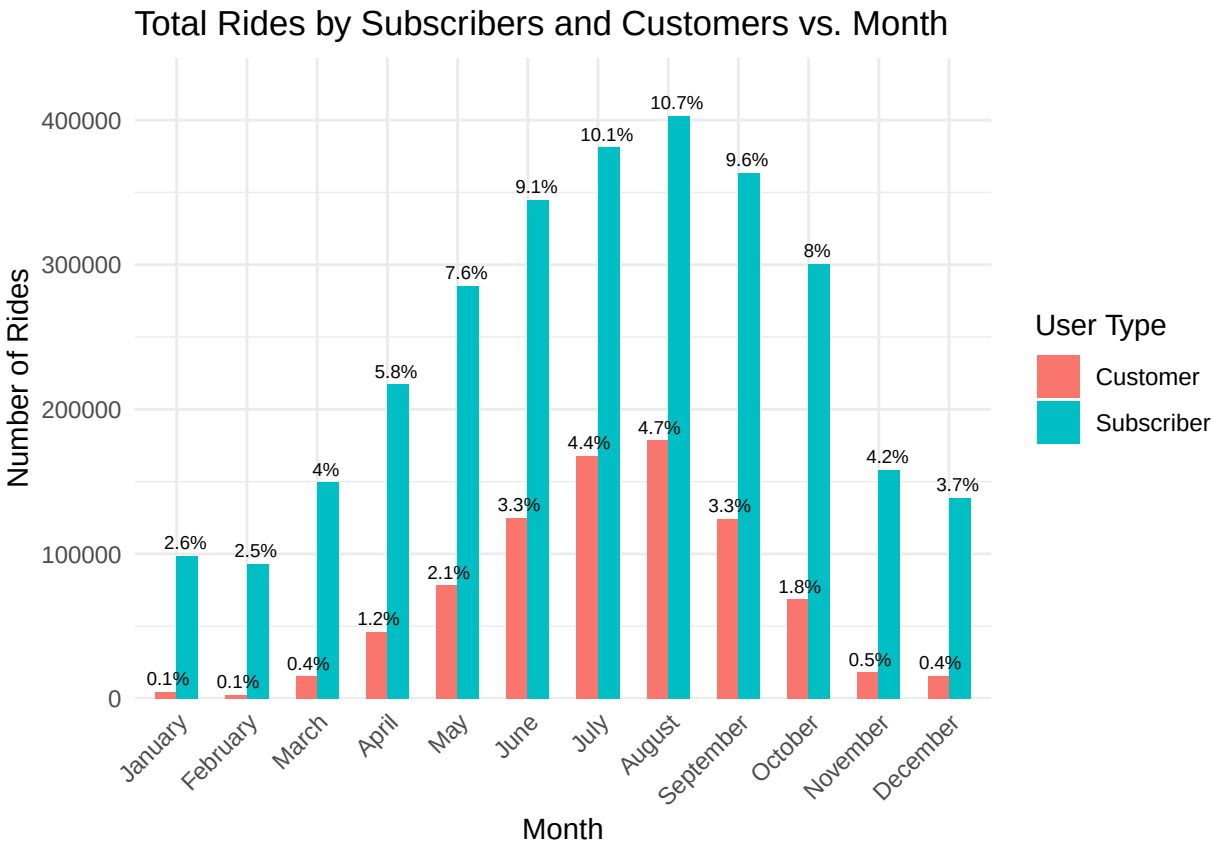
From the table above, we notice that regular customers took bikes for longer rides (25 minutes) than subscribers (10 minutes), as the median trip duration for customers is lower than the median trip duration for subscribers.

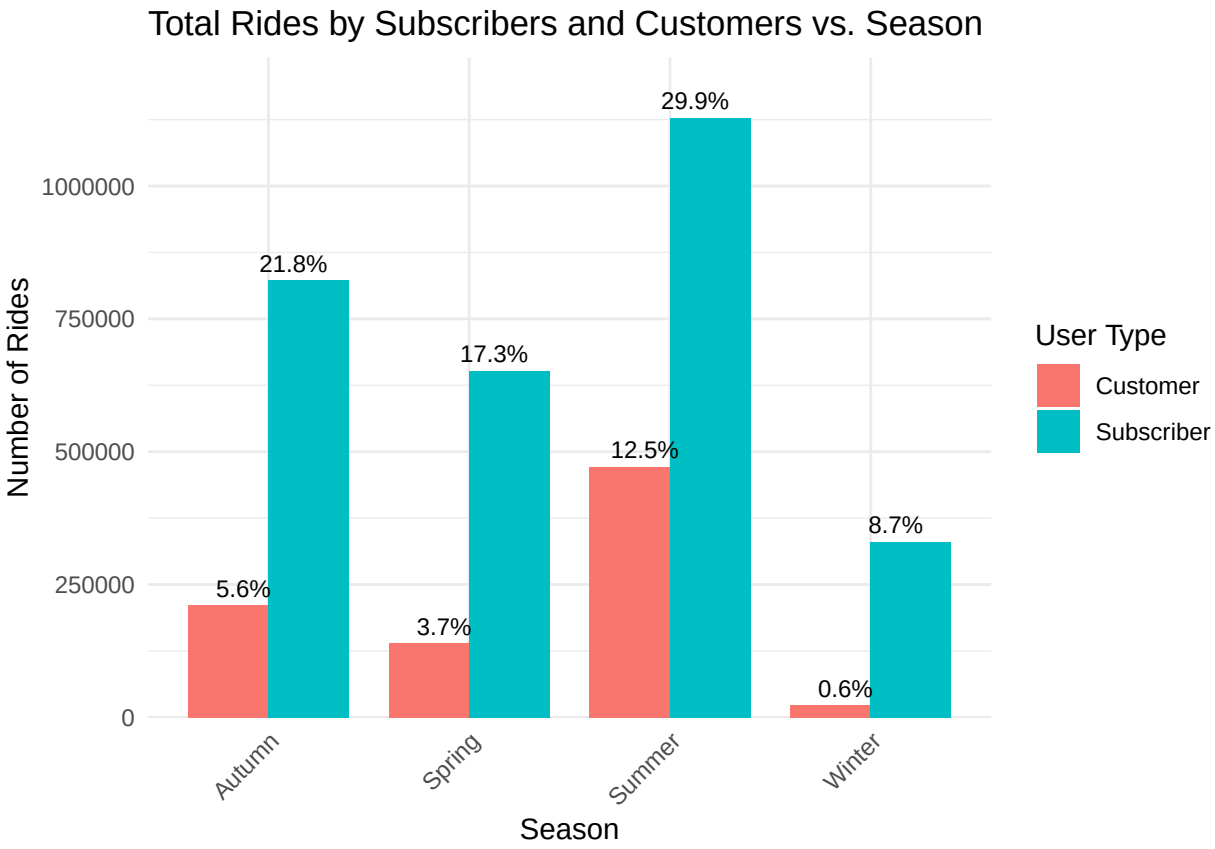
The median trip duration for each usertype is much lower than the average trip duration, suggesting that the data for each usertype is right-skewed - confirming that the overall data is right-skewed.

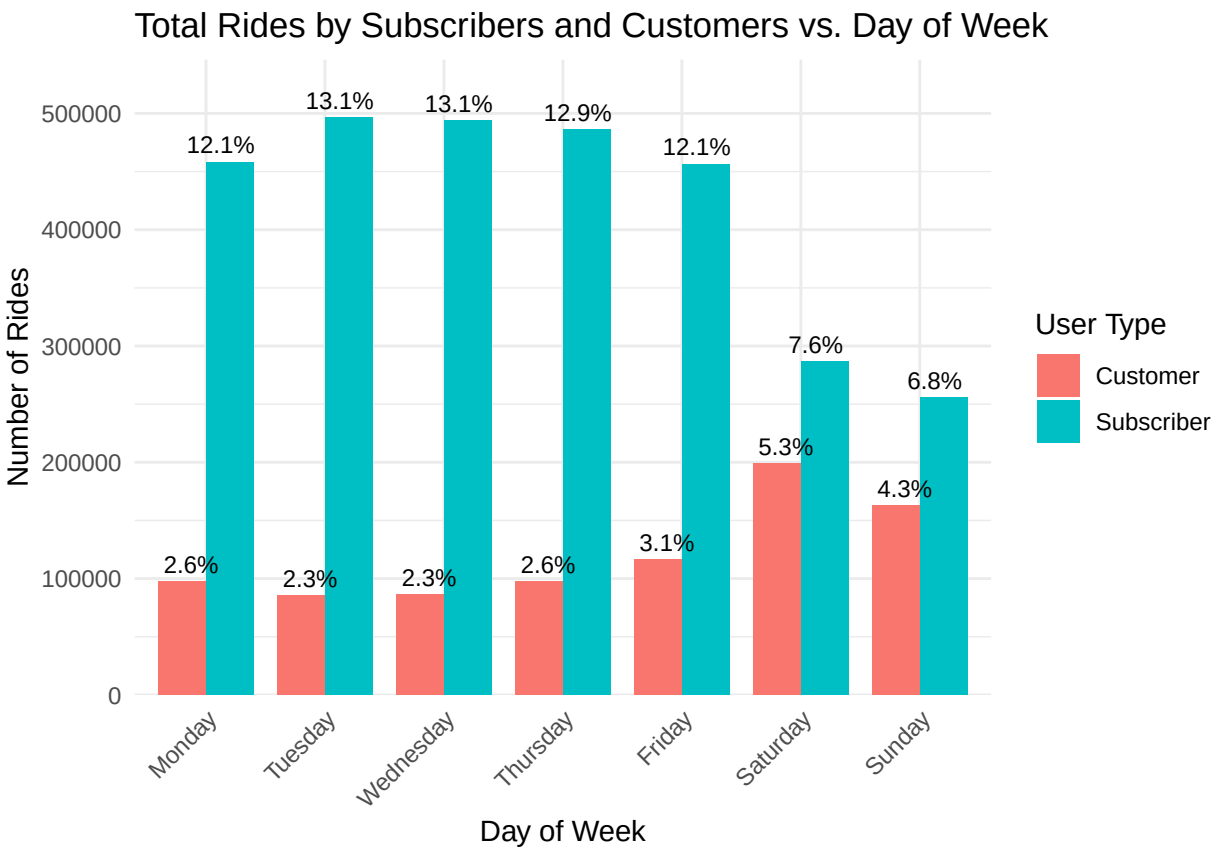
Total Trip Duration Analysis

Due to the vast number of trips taken (over 3 million in 2019), we will find out the ride activity for both usertypes across different timelines. We will group each time-frame by usertype.

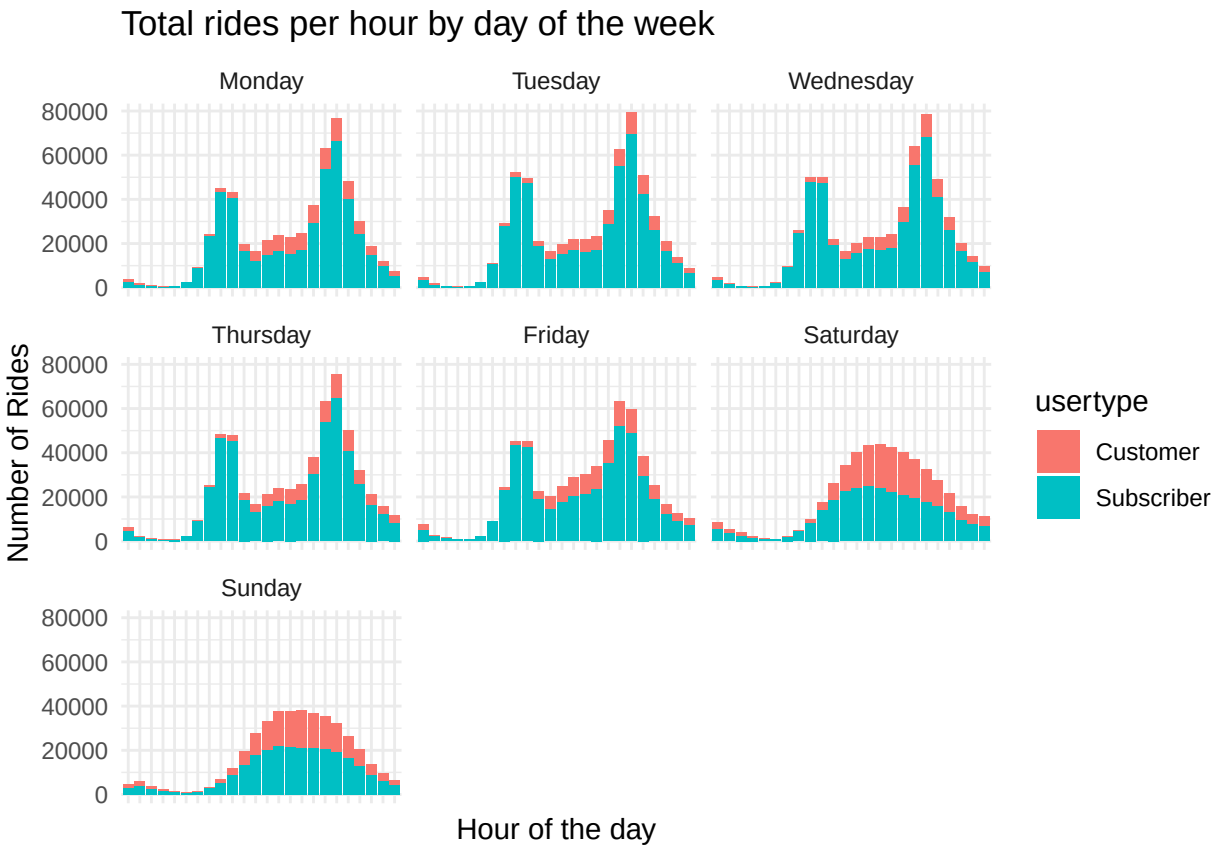
NOTE: For these analyses, we will use the median trip duration as it is a better measure of central tendency for skewed data.


Total Rides by Subscribers and Customers vs. Month
Number of Rides
400000
300000
200000
100000
0
0.1%
2.6%
0.1%
2.5%
0.4%
4%
1.2%
5.8%
2.1%
7.6%
3.3%
9.1%
4.4%
10.1%
4.7%
10.7%
3.3%
9.6%
1.8%
8%
0.5%
4.2%
0.4%
3.7%
January
February
March
April
May
June
July
August
September
October
November
December
Month
User Type
Customer
Subscriber


Total Rides by Subscribers and Customers vs. Season
Number of Rides
1000000
750000
500000
250000
0
5.6%
21.8%
3.7%
17.3%
12.5%
29.9%
0.6%
8.7%
Autumn
Spring
Summer
Winter
Season
User Type
Customer
Subscriber


Total Rides by Subscribers and Customers vs. Day of Week
Number of Rides
500000
400000
300000
200000
100000
0
12.1%
13.1%
13.1%
12.9%
12.1%
7.6%
6.8%
5.3%
4.3%
3.1%
2.6%
2.3%
2.3%
2.6%
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of Week
User Type
Customer
Subscriber

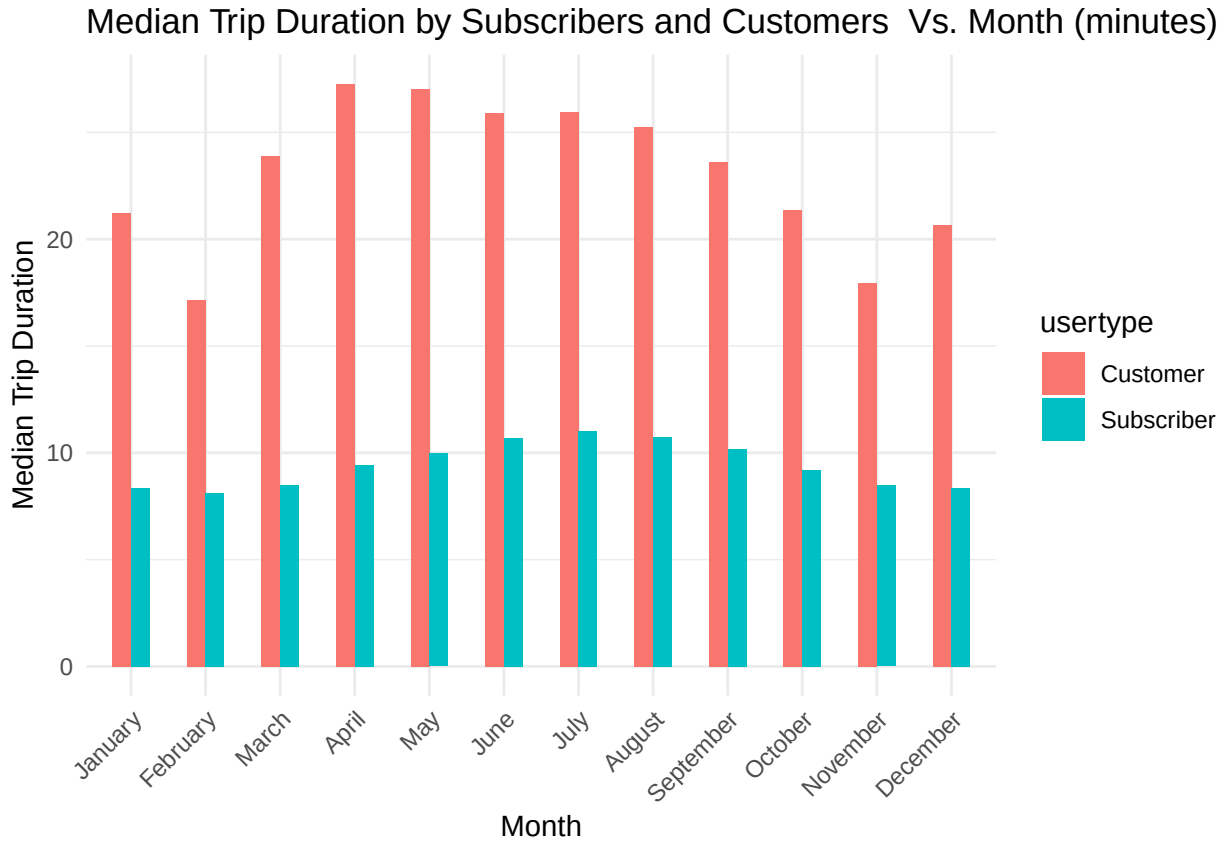

Total Rides by Subscribers and Customers vs. Time
Number of Rides
300000
200000
100000
0
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Time
User Type
Customer
Subscriber

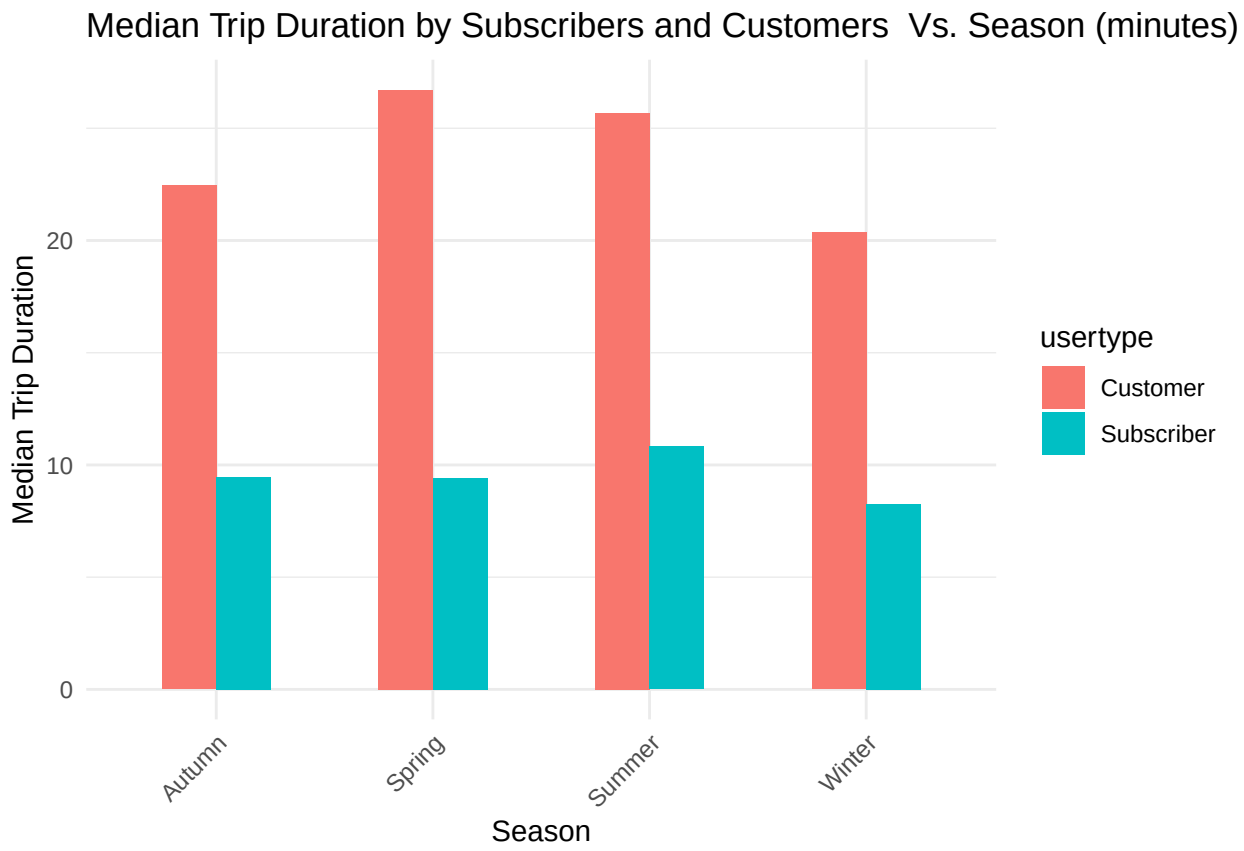


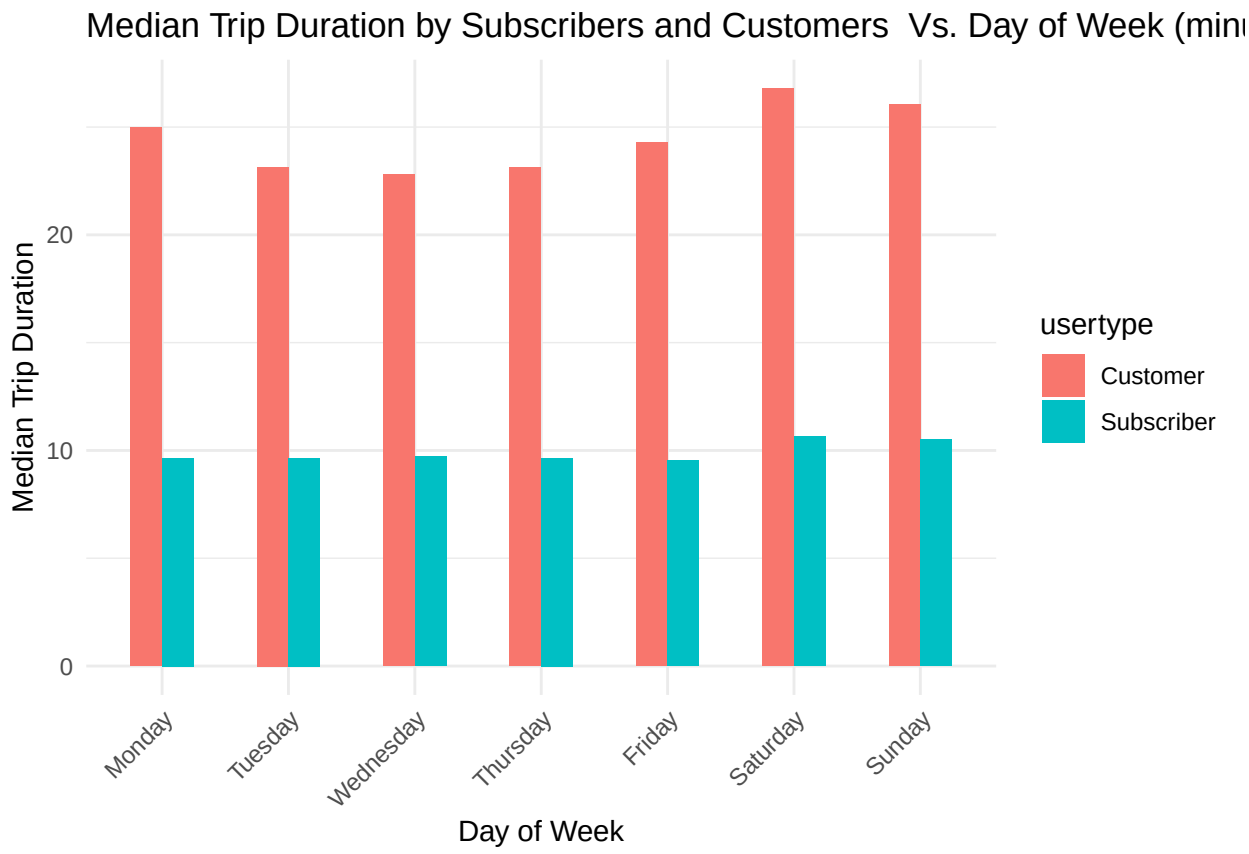
- From the data, August emerges as the busiest month, amounting to 15.4% of the yearly total. Subscribers lead this period, accounting for 69% of August trips. Conversely, February records the lowest activity with 2.6% of yearly trips. July and August stand out as Customers' preferred month, with 4.4% and 4.7% of the yearly total respectively. January and February see a significant difference in trips between Subscribers and Customers, with Subscribers accounting for about 96% of that month's trips. Intuitively, this makes sense as Customers would not opt to use bikes during winter, whereas Subscribers may feel the obligation to use bikes seeing that they already have a membership. Subscriber total monthly rides are higher throughout the year.
- Both Subscribers and Customers ride more frequently in summer and less in winter, with total rides taken in Summer accounting for 42.4% of total trips.
- Both Subscribers and Customers took consistent trips throughout the weekday. Divergence occurs as we approach the weekend, where Subscribers tend to not take as many trips and Customers ramping up their bike usage. This suggests a preference for weekday trips for Subscribers and weekend trips for Customers.
- Peak riding hours are between 16:00 and 19:00 for Subscribers and Customers, with the highest activity at 18:00. Conversely, minimal activity is recorded between 2:00 and 5:00.There is a notable increase in Subscriber activity between 6:00 and 10:00, possibly due to Subscribers mainly using Cyclistic bikes to travel to and from consistent loactions (e.g work, school, train station)
- There is a big increase in the volume of rides during the weekdays between 6:00 and 10:00 and another volume increase from 16:00 to 19:00. This likely confirms the theory that subscribers use the bikes as transportation. Weekends tell a different story. Saturday's and Sunday's see more ride volumes during the afternoon to early evening, suggesting that users (customers in particular) mostly use bikes for leisure activity.

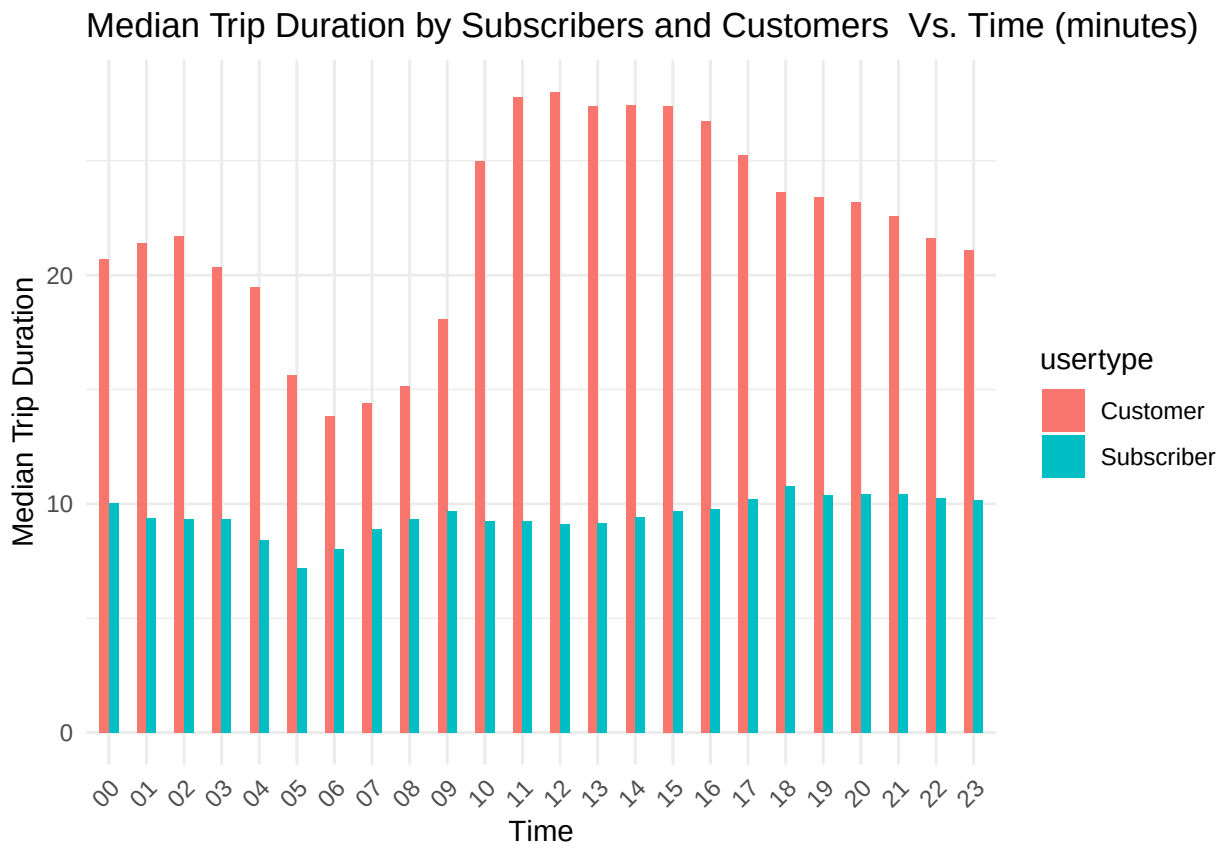
Median Trip Duration Analysis

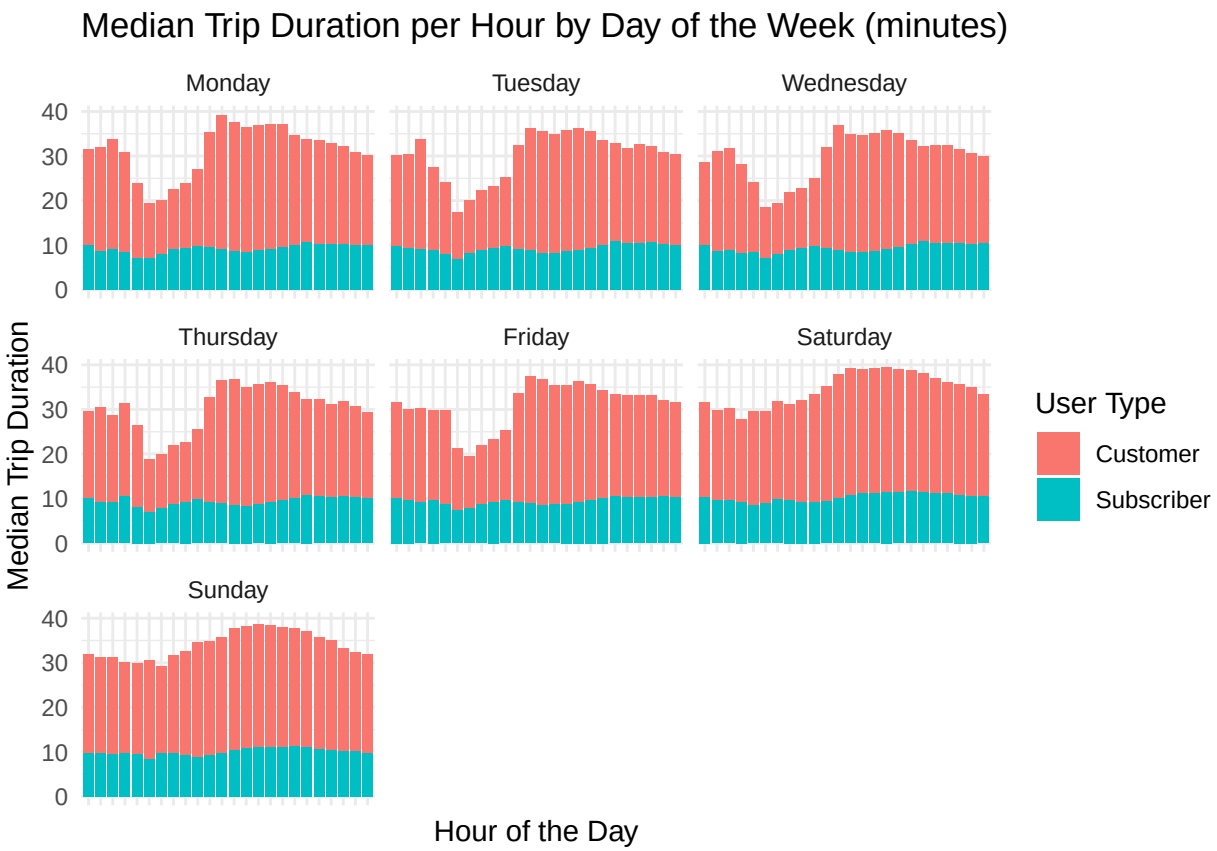
The next part of our analysis will involve delving into the median trip duration taken by subscribers and customers. This will give us insight into how long each usertype normally spends when riding. This analysis can be used on conjuction with pricing analysis to determine optimal weekday, weekend pricing for customers.




Median Trip Duration by Subscribers and Customers Vs. Season (minutes)
Median Trip Duration
20
10
0
Autumn
Spring
Summer
Winter
Season
usertype
Customer
Subscriber


Median Trip Duration by Subscribers and Customers Vs. Day of Week (min
Median Trip Duration
20
10
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of Week
usertype
Customer
Subscriber


Median Trip Duration by Subscribers and Customers Vs. Time (minutes)
Median Trip Duration
0
10
20
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Time
usertype
Customer
Subscriber

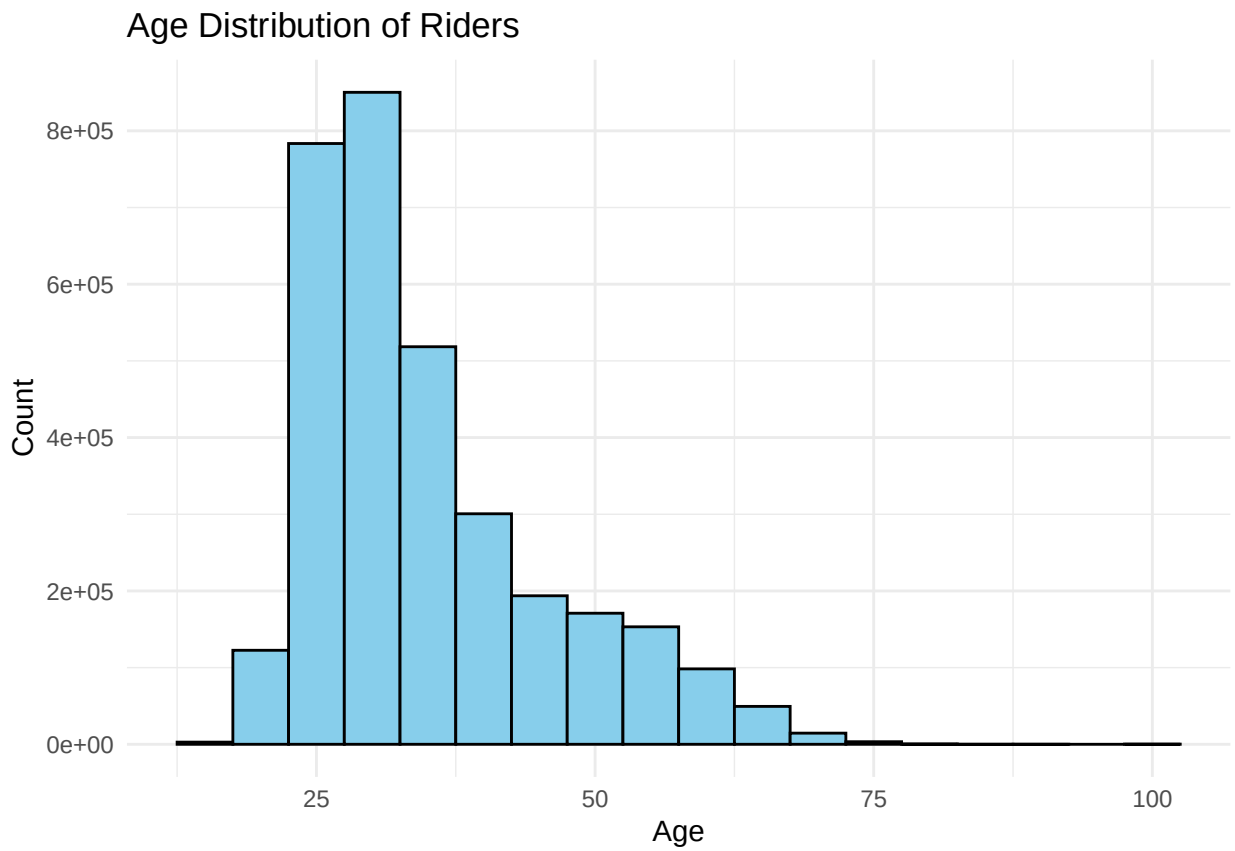


- The median trip duration for Subscribers is somewhat constant around 10 minutes throughout the year. Customers' median trip duration oscillates between 15 and 30 minutes throughout the year, exhibiting greater volatility.
- Earlier, we noticed that both subscribers and customers dramatically reduce their bike usage during winter. Looking at the median trip duration time, we see that although theres a significant drop in usage, the median time taken to complete a trip does not drop as would be expected.
- Similarly, athough we see customers' peak bike usage in the summer, the median trip duration peaks in spring.
- Weekend median trip duration is higher for both subscribers and customers.

It's evident that customers exhibit higher median values and volatility compared to subscribers throughout all time frames.

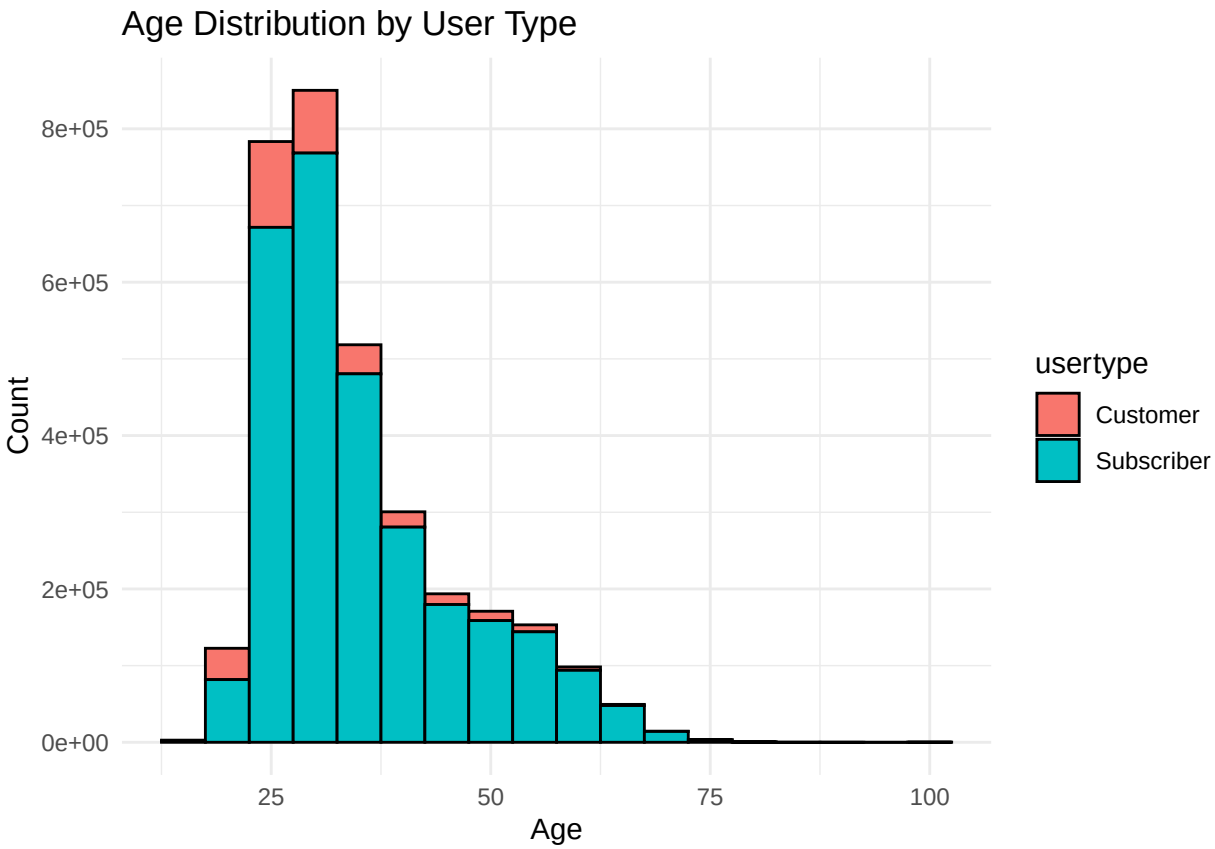
Age Analysis

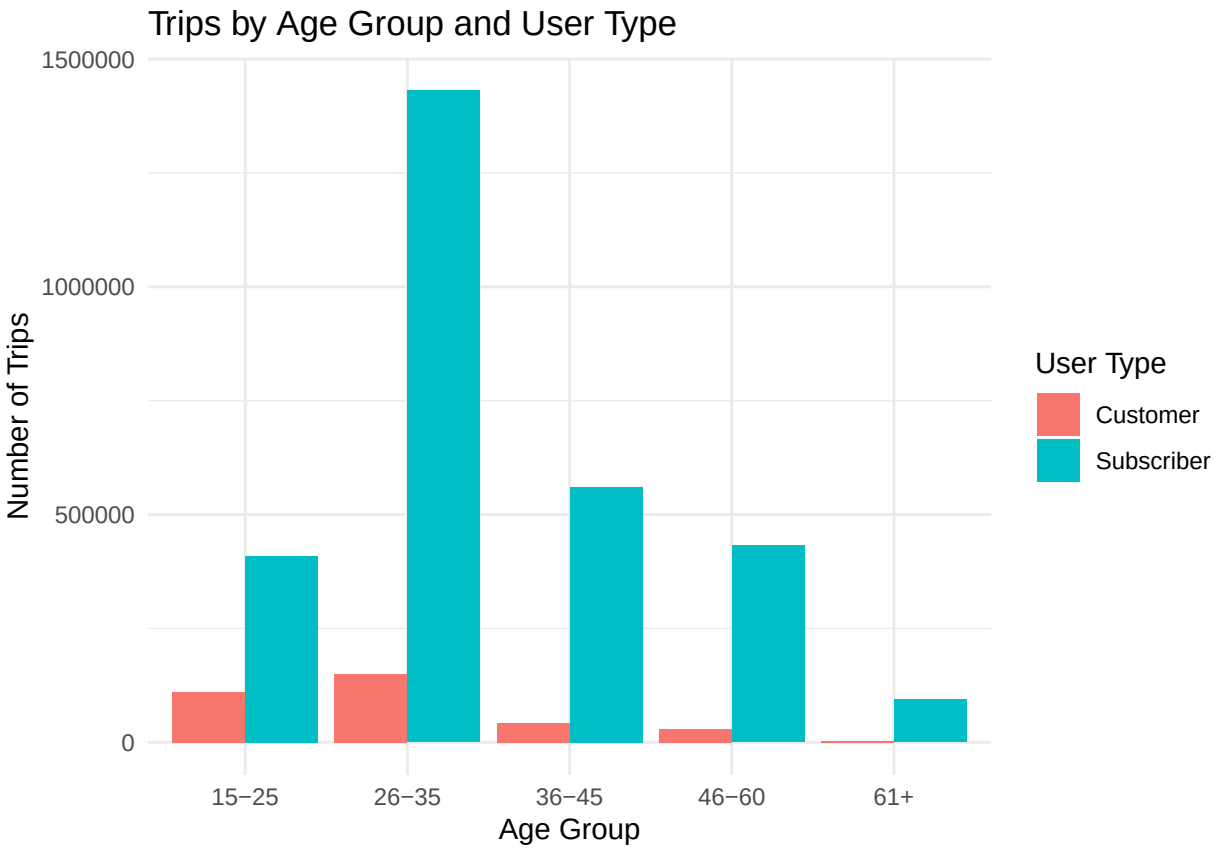
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.00   27.00   32.00   34.91   40.00   99.00
```



The majority of Cyclistic users tend to be younger with the median age being 32 (as of 2019).

```
## # A tibble: 5 x 2
##   age_group   trips
##   <fct>       <int>
## 1 15-25      518944
## 2 26-35     1581940
## 3 36-45      602582
## 4 46-60      461428
## 5 61+         97310
```


Age Distribution by User Type
8e+05
6e+05
4e+05
2e+05
0e+00
Count
25
50
75
100
Age
usertype
Customer
Subscriber



- The majority of users are aged between 15 and 45 years old. This aludes to the Cyclistic users being in education and working class (early careers and experienced professionals).
- The bulk of customers are aged between 15 and 35 years old.
- The bulk of subscribers are aged between 26 and 35 years old.

Station Analysis by Usertype

We will first identify the top 10 stations used by both subscribers and customers.

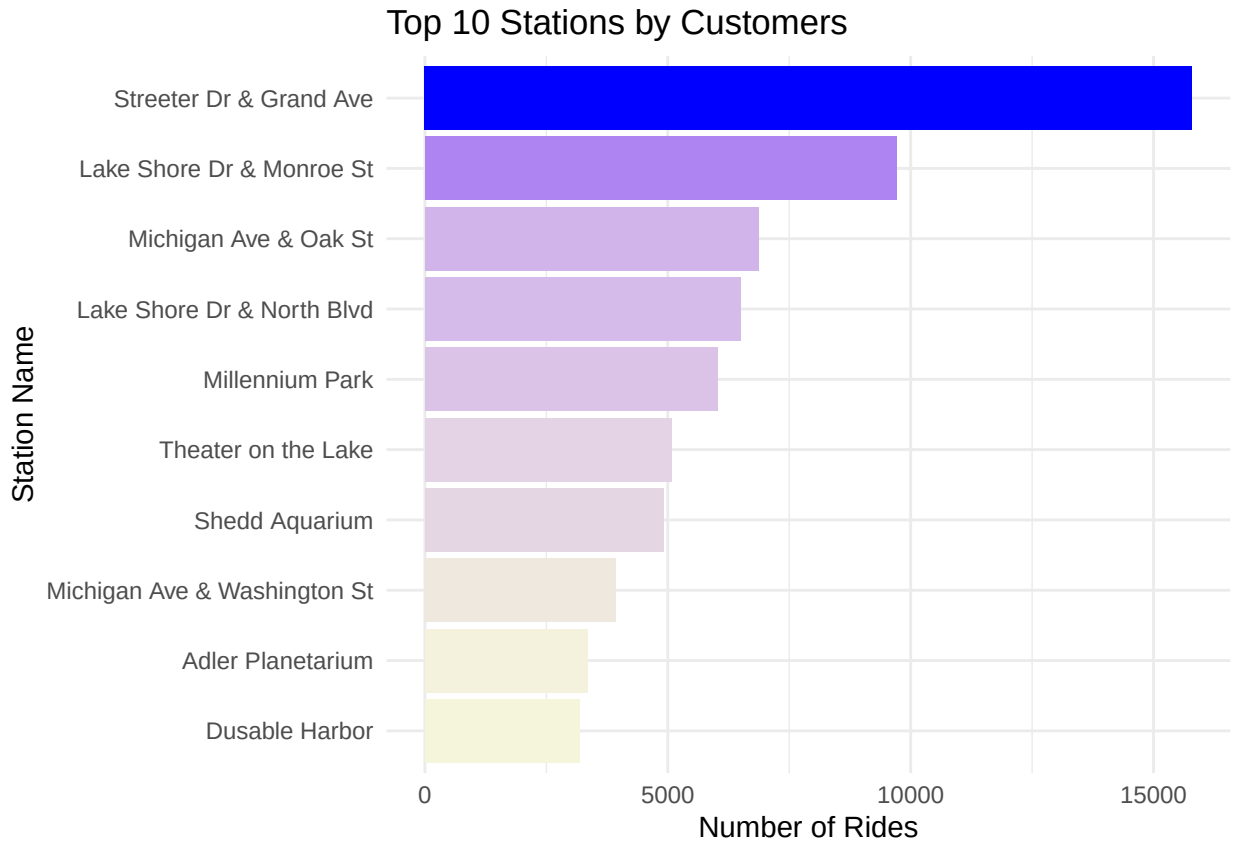
```
## # A tibble: 20 x 3
##    usertype   from_station_name            number_of_rides
##    <chr>      <chr>                                  <int>
##  1 Customer   Streeter Dr & Grand Ave                15792
##  2 Customer   Lake Shore Dr & Monroe St               9703
##  3 Customer   Michigan Ave & Oak St                   6874
##  4 Customer   Lake Shore Dr & North Blvd              6491
##  5 Customer   Millennium Park                         6029
##  6 Customer   Theater on the Lake                     5082
##  7 Customer   Shedd Aquarium                          4918
##  8 Customer   Michigan Ave & Washington St            3932
##  9 Customer   Adler Planetarium                       3354
## 10 Customer   Dusable Harbor                          3185
## 11 Subscriber Canal St & Adams St                    50496
```

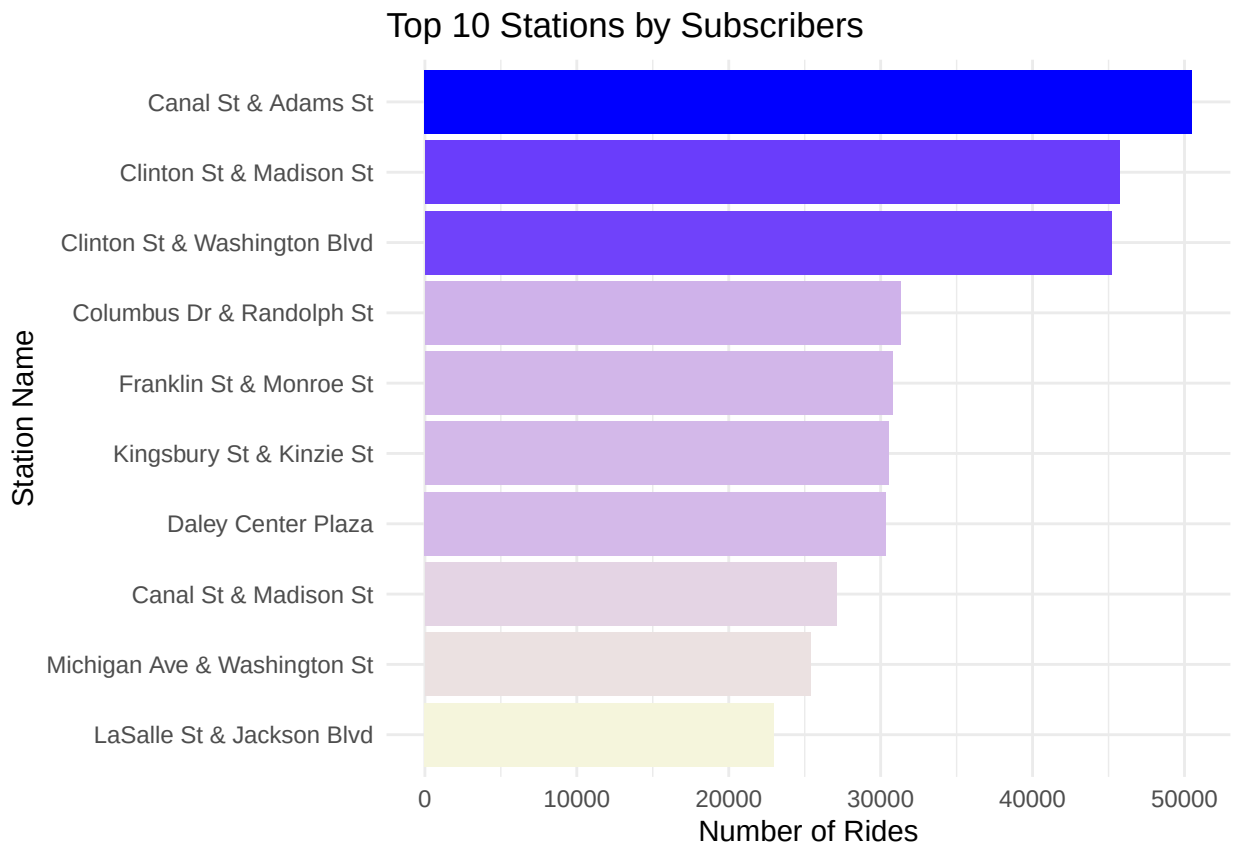
```
## 12 Subscriber Clinton St & Madison St               45723
## 13 Subscriber Clinton St & Washington Blvd          45201
## 14 Subscriber Columbus Dr & Randolph St             31309
## 15 Subscriber Franklin St & Monroe St               30759
## 16 Subscriber Kingsbury St & Kinzie St              30525
## 17 Subscriber Daley Center Plaza                    30346
## 18 Subscriber Canal St & Madison St                 27076
## 19 Subscriber Michigan Ave & Washington St          25391
## 20 Subscriber LaSalle St & Jackson Blvd             22985
```

Customers mostly start their journey from Streeter Dr & Grand Ave, then Lake Shore Dr & Monroe St, then Millenium Park.

Subscribers mostly start their journey from Canal St & Adams St, followed by Clinton St & Madison St, then Clinton St & Washington Blvd.

Graphically, the popularity of the top 10 stations for each usertype are shown below.





After visualising the top 10 stations where users start their journey from, we would next want to identify the top 10 routes that users took during the year. Once we can identify the top 10 routes for each usertype, we can begin to recommend targeted marketing strategies.

Top 10 Customer Routes

```
## # A tibble: 10 x 3
##    from_station_name         to_station_name           number_of_trips
##    <chr>                     <chr>                               <int>
##  1 Lake Shore Dr & Monroe St Streeter Dr & Grand Ave              2009
##  2 Streeter Dr & Grand Ave   Streeter Dr & Grand Ave              1724
##  3 Lake Shore Dr & Monroe St Lake Shore Dr & Monroe St            1409
##  4 Michigan Ave & Oak St     Michigan Ave & Oak St                1187
##  5 Streeter Dr & Grand Ave   Millennium Park                     1007
##  6 Millennium Park           Streeter Dr & Grand Ave               855
##  7 Shedd Aquarium            Streeter Dr & Grand Ave               761
##  8 Streeter Dr & Grand Ave   Theater on the Lake                  734
##  9 Streeter Dr & Grand Ave   Lake Shore Dr & Monroe St             691
## 10 Streeter Dr & Grand Ave   Michigan Ave & Oak St                 668
```

Top 10 Subscriber routes

```
## # A tibble: 10 x 3
##    from_station_name           to_station_name             number_of_trips
##    <chr>                       <chr>                                 <int>
```

```
##  1 Canal St & Adams St          Michigan Ave & Washington St            3537
##  2 Michigan Ave & Washington St Clinton St & Washington Blvd            2776
##  3 Columbus Dr & Randolph St    Clinton St & Washington Blvd            2638
##  4 Michigan Ave & Washington St Canal St & Adams St                     2536
##  5 Canal St & Madison St        Michigan Ave & Washington St            2326
##  6 Clinton St & Washington Blvd Michigan Ave & Washington St            2230
##  7 Columbus Dr & Randolph St    State St & Randolph St                  2187
##  8 Wacker Dr & Washington St    Michigan Ave & Washington St            2120
##  9 Columbus Dr & Randolph St    Canal St & Adams St                     2113
## 10 MLK Jr Dr & 29th St          State St & 33rd St                      2005
```

Usage Patterns by User Type:

- Subscribers took significantly more trips than Customers.

*The most popular starting stations for Subscribers recorded higher ride volumes than those for Customers.

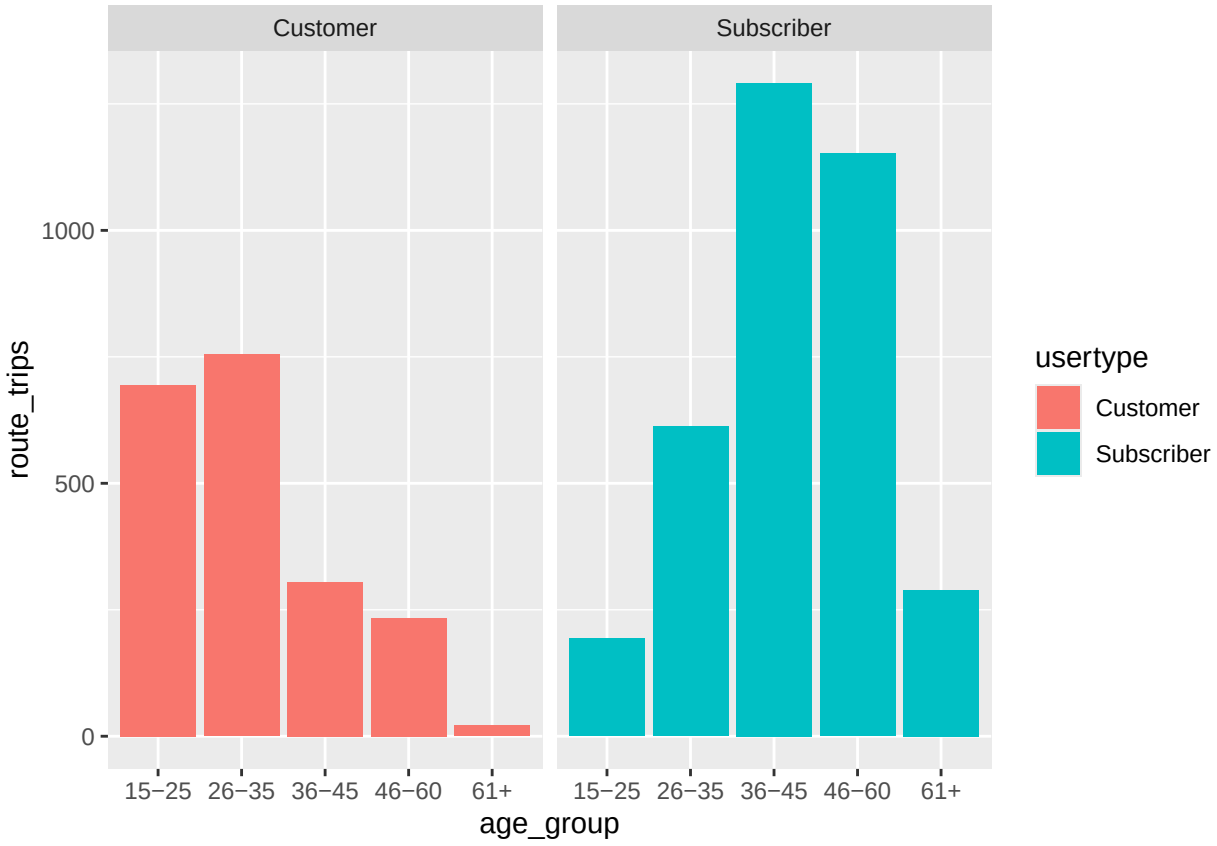
- The top 10 routes among Subscribers had more trips than the top 10 Customer routes.

This indicates that Subscribers, likely regular commuters, exhibit:

- Higher ride frequency
- Consistent usage of specific stations and routes

Top Route Geomapping I next wanted to find out why the top routes were the the top routes. In order to do that, I needed to map them out to identify if there was anything interesting I could find.

```
## # A tibble: 2 x 4
##   usertype   from_station_name         to_station_name           number_of_trips
##   <chr>      <chr>                     <chr>                               <int>
## 1 Customer   Lake Shore Dr & Monroe St Streeter Dr & Grand Ave              2009
## 2 Subscriber Canal St & Adams St       Michigan Ave & Washingto~            3537
```




```
## NULL
```

- From the graph above, we notice that the top route for customers were completed mainly by the age groups 15 - 25 and 26-35
- The top route for subscribers were mainly completed by the age groups 36 - 45 and 46 - 60.

```
## # A tibble: 4 x 4
##   station_name                address                                  lon   lat
##   <chr>                       <chr>                                  <dbl> <dbl>
## 1 Streeter Dr & Grand Ave     Streeter Dr and Grand Ave, Chicago, ~ -87.6  41.9
## 2 Lake Shore Dr & Monroe St   Lake Shore Dr and Monroe St, Chicago~ -87.6  41.9
## 3 Canal St & Adams St         Canal St and Adams St, Chicago, IL    -87.6  41.9
## 4 Michigan Ave & Washington St Michigan Ave and Washington St, Chic~ -87.6  41.9
```

- Based on the most popular route for Subscribers, it appears that riders typically travel from the train station located on Canal St & Adam St. This suggests that subscribers may be commuting to and from the work using the train and Cyclistic bikes.
- For customers, the route between Lake Shore Dr & Monroe St Ave and Streeter Dr & Grand Ave indicates travel along the lakefront in downtown Chicago. This may possibly be to enjoy the scenic views of Lake Michigan and for access to downtown tourist attractions.

Share

Main insights and finding conclusions

Findings:

- Subscribers hold the largest proportion of the total rides, ~77% of total rides.
- Subscribers contantly hold larger proportion of total rides throughout the year.
- Customers have the biggest volume of data on the weekend.
- The largest volume of riders is in the afternoon.
- Customer median trip duration is longer than subscribers.

Insights: The data suggests that subscribers use bikes for work purposes. This is confirmed by their bike usage in colder months (where there is significant drop in casual members in those months), spikes in ride usage during typical opening and closing work hours through the weekday, and the most popular route emanating from a train station.

Act

To target customers, I would recommend:

1. Offering a weekend-only subscription at a different price point than the full annual subscription
2. Creating a tiered weekend-only subscription by age group, making the 15 - 25 and 26 - 35 age group prices cheaper than the rest. This will psychologically entice those age groups into thinking they were able to "get a bargain".
3. Introducing coupons and discounts along with the weekend-only subscription.
4. Creating and advertising posters/marketing material for the new offerings in the stations that are most used by customers from June to September (most popular months for customers).

Additionally, to retain subscribers, I would recommend:

1. Offering discount on renting bikes for subscribers.
2. Partnering with the transportation council to offer joint discounts to subscribers who use trains and bikes.

To assist in pricing analysis for customers, I would suggest:

- Charging on a per minute basis.

Conclusion

Summary

We have completed our analysis for Cyclistic. We conducted various data cleaning, manipulation, and wrangling tasks, including investigation, cleaning, removal of extreme outliers, and more. Additionally, we analyzed the distributions of user types, as well as the distribution of trips and trip duration taken across all timeframes. We also examined the distributions of trips across different age groups. Finally, we conducted a geographical analysis to identify the most popular stations and routes for both user types.

Limitations of the Analysis

The analysis provided is restricted to total, trips taken, minutes and distance only, lacking weather, temperature, and humidity data that could influence trip duration.