Cyclist Analysis

Braden Leonard

July 16, 2020

Cyclist Full Year Analysis

This is the analysis for the Google Data Analytics Certificate capstone Project

The scenario presented was the marketing team from Cyclist, a bike-sharing company in Chicago, wants to maximize their annual memberships. They wanted to understand how casual riders and membership riders use the bike share differently. Their goal is to learn how to convert casual riders to annual members. I will follow the Ask, Prepare, Process, Analyze, Share and Act steps to the data analysis.

Ask

Now if this were an actual business task, I would ask the questions to their representatives to better understand what they are hoping to get presented to them, what metrics we are working with(possible errors), and if they would like suggestions. Since this is a capstone project, we will have to try and answer these questions ourselves.

Prepare

They provided us a divvy-data bases of all their trip data since they started in 2014. I since we require the most recent data for our analysis to be relevant, We loaded the most recent data(2019-2020).

Most of these files are too large for programs like excel, so R is the perfect fit for cleaning and manipulating this large dataset.

Process

To start we will need to install the necessary packages:

```
#install.packages("tidyverse")
#("lubridate")
```

We will then have to load these packages:

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
```

```
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     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
```

Now we have to import the last 4 quarters for data to be cleaned.

```
q1 <- read.csv("Divvy_Trips_2019_Q2.csv")
q2 <- read.csv("Divvy_Trips_2019_Q3.csv")
q3 <- read.csv("Divvy_Trips_2019_Q4.csv")
q4 <- read.csv("Divvy_Trips_2020_Q1.csv")
```

Now lets take a look at these data sets so see how they are similar or different.

```
colnames(q1)

##  [1] "X01...Rental.Details.Rental.ID"
##  [2] "X01...Rental.Details.Local.Start.Time"
##  [3] "X01...Rental.Details.Local.End.Time"
##  [4] "X01...Rental.Details.Bike.ID"
##  [5] "X01...Rental.Details.Duration.In.Seconds.Uncapped"
##  [6] "X03...Rental.Start.Station.ID"
##  [7] "X03...Rental.Start.Station.Name"
##  [8] "X02...Rental.End.Station.ID"
##  [9] "X02...Rental.End.Station.Name"
## [10] "User.Type"
## [11] "Member.Gender"
## [12] "X05...Member.Details.Member.Birthday.Year"

colnames(q2)

##  [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(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"
```

```
colnames(q4)
```

```
##  [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"
```

Cleaning

Looks like the naming for each column is not consistent and will need to be changed if we want to merge these into one data set. We will use q4 (in this case the first quarter of 2020) as a baseline for their column names.

```
(q1 <- rename(q1
              ,ride_id=X01...Rental.Details.Rental.ID
              ,rideable_type=X01...Rental.Details.Bike.ID
              ,started_at=X01...Rental.Details.Local.Start.Time
              ,ended_at= X01...Rental.Details.Local.End.Time
              ,start_station_name=X03...Rental.Start.Station.Name
              ,start_station_id=X03...Rental.Start.Station.ID
              ,end_station_name=X02...Rental.End.Station.Name
              ,end_station_id=X02...Rental.End.Station.ID
              ,member_casual=User.Type))

(q2 <- rename(q2
              ,ride_id=trip_id
              ,rideable_type=bikeid
              ,started_at=start_time
              ,ended_at=end_time
              ,start_station_name=from_station_name
              ,start_station_id=from_station_id
              ,end_station_name=to_station_name
              ,end_station_id=to_station_id
              ,member_casual=usertype))

(q3 <- rename(q3
              ,ride_id=trip_id
              ,rideable_type=bikeid
              ,started_at=start_time
              ,ended_at=end_time
              ,start_station_name=from_station_name
              ,start_station_id=from_station_id
              ,end_station_name=to_station_name
```

```
              ,end_station_id=to_station_id
              ,member_casual=usertype))

colnames(q1)
```

```
##  [1] "ride_id"
##  [2] "started_at"
##  [3] "ended_at"
##  [4] "rideable_type"
##  [5] "X01...Rental.Details.Duration.In.Seconds.Uncapped"
##  [6] "start_station_id"
##  [7] "start_station_name"
##  [8] "end_station_id"
##  [9] "end_station_name"
## [10] "member_casual"
## [11] "Member.Gender"
## [12] "X05...Member.Details.Member.Birthday.Year"
```

```
colnames(q2)
```

```
##  [1] "ride_id"            "started_at"         "ended_at"
##  [4] "rideable_type"      "tripduration"       "start_station_id"
##  [7] "start_station_name" "end_station_id"     "end_station_name"
## [10] "member_casual"      "gender"             "birthyear"
```

```
colnames(q3)
```

```
##  [1] "ride_id"            "started_at"         "ended_at"
##  [4] "rideable_type"      "tripduration"       "start_station_id"
##  [7] "start_station_name" "end_station_id"     "end_station_name"
## [10] "member_casual"      "gender"             "birthyear"
```

```
colnames(q4)
```

```
##  [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"
```

Now lets inspect these dataframe to see what columns will be useful

```
str(q1)
```

```
## 'data.frame':    1108163 obs. of  12 variables:
##  $ ride_id                                           : int  22178529
22178530 22178531 22178532 22178533 22178534 22178535 22178536 22178537
22178538 ...
##  $ started_at                                        : chr  "2019-04-01
00:02:22" "2019-04-01 00:03:02" "2019-04-01 00:11:07" "2019-04-01 00:13:01"
...
```

```
##  $ ended_at                                          : chr  "2019-04-01
00:09:48" "2019-04-01 00:20:30" "2019-04-01 00:15:19" "2019-04-01 00:18:58"
...
##  $ rideable_type                                     : int  6251 6226 5649
4151 3270 3123 6418 4513 3280 5534 ...
##  $ X01...Rental.Details.Duration.In.Seconds.Uncapped: chr  "446.0"
"1,048.0" "252.0" "357.0" ...
##  $ start_station_id                                  : int  81 317 283 26
202 420 503 260 211 211 ...
##  $ start_station_name                                : chr  "Daley Center
Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct &
Illinois St" ...
##  $ end_station_id                                    : int  56 59 174 133
129 426 500 499 211 211 ...
##  $ end_station_name                                  : chr  "Desplaines St
& Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury
St & Kinzie St" ...
##  $ member_casual                                     : chr  "Subscriber"
"Subscriber" "Subscriber" "Subscriber" ...
##  $ Member.Gender                                     : chr  "Male" "Female"
"Male" "Male" ...
##  $ X05...Member.Details.Member.Birthday.Year         : int  1975 1984 1990
1993 1992 1999 1969 1991 NA NA ...
```

```
str(q2)
```

```
## 'data.frame':    1640718 obs. of  12 variables:
##  $ ride_id           : int  23479388 23479389 23479390 23479391 23479392
23479393 23479394 23479395 23479396 23479397 ...
##  $ started_at        : chr  "2019-07-01 00:00:27" "2019-07-01 00:01:16"
"2019-07-01 00:01:48" "2019-07-01 00:02:07" ...
##  $ ended_at          : chr  "2019-07-01 00:20:41" "2019-07-01 00:18:44"
"2019-07-01 00:27:42" "2019-07-01 00:27:10" ...
##  $ rideable_type     : int  3591 5353 6180 5540 6014 4941 3770 5442 2957
6091 ...
##  $ tripduration      : chr  "1,214.0" "1,048.0" "1,554.0" "1,503.0" ...
##  $ start_station_id  : int  117 381 313 313 168 300 168 313 43 43 ...
##  $ start_station_name: chr  "Wilton Ave & Belmont Ave" "Western Ave &
Monroe St" "Lakeview Ave & Fullerton Pkwy" "Lakeview Ave & Fullerton Pkwy"
...
##  $ end_station_id    : int  497 203 144 144 62 232 62 144 195 195 ...
##  $ end_station_name  : chr  "Kimball Ave & Belmont Ave" "Western Ave &
21st St" "Larrabee St & Webster Ave" "Larrabee St & Webster Ave" ...
##  $ member_casual     : chr  "Subscriber" "Customer" "Customer" "Customer"
...
##  $ gender            : chr  "Male" "" "" "" ...
##  $ birthyear         : int  1992 NA NA NA NA 1990 NA NA NA NA ...
```

```
str(q3)
```

```
## 'data.frame':    704054 obs. of  12 variables:
##  $ ride_id           : int  25223640 25223641 25223642 25223643 25223644
25223645 25223646 25223647 25223648 25223649 ...
##  $ started_at        : chr  "2019-10-01 00:01:39" "2019-10-01 00:02:16"
"2019-10-01 00:04:32" "2019-10-01 00:04:32" ...
##  $ ended_at          : chr  "2019-10-01 00:17:20" "2019-10-01 00:06:34"
"2019-10-01 00:18:43" "2019-10-01 00:43:43" ...
##  $ rideable_type     : int  2215 6328 3003 3275 5294 1891 1061 1274 6011
2957 ...
##  $ tripduration      : chr  "940.0" "258.0" "850.0" "2,350.0" ...
##  $ start_station_id  : int  20 19 84 313 210 156 84 156 156 336 ...
##  $ start_station_name: chr  "Sheffield Ave & Kingsbury St" "Throop
(Loomis) St & Taylor St" "Milwaukee Ave & Grand Ave" "Lakeview Ave &
Fullerton Pkwy" ...
##  $ end_station_id    : int  309 241 199 290 382 226 142 463 463 336 ...
##  $ end_station_name  : chr  "Leavitt St & Armitage Ave" "Morgan St & Polk
St" "Wabash Ave & Grand Ave" "Kedzie Ave & Palmer Ct" ...
##  $ member_casual     : chr  "Subscriber" "Subscriber" "Subscriber"
"Subscriber" ...
##  $ gender            : chr  "Male" "Male" "Female" "Male" ...
##  $ birthyear         : int  1987 1998 1991 1990 1987 1994 1991 1995 1993
NA ...
```

```
str(q4)
```

```
## 'data.frame':    426887 obs. of  13 variables:
##  $ ride_id           : chr  "EACB19130B0CDA4A" "8FED874C809DC021"
"789F3C21E472CA96" "C9A388DAC6ABF313" ...
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike"
"docked_bike" ...
##  $ started_at        : chr  "2020-01-21 20:06:59" "2020-01-30 14:22:39"
"2020-01-09 19:29:26" "2020-01-06 16:17:07" ...
##  $ ended_at          : chr  "2020-01-21 20:14:30" "2020-01-30 14:26:22"
"2020-01-09 19:32:17" "2020-01-06 16:25:56" ...
##  $ start_station_name: chr  "Western Ave & Leland Ave" "Clark St &
Montrose Ave" "Broadway & Belmont Ave" "Clark St & Randolph St" ...
##  $ start_station_id  : int  239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr  "Clark St & Leland Ave" "Southport Ave &
Irving Park Rd" "Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave" ...
##  $ end_station_id    : int  326 318 117 24 212 96 212 212 96 100 ...
##  $ start_lat         : num  42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

Need to convert ride_id and rideable_id to characters so they can stack correctly.

```
q4<-  mutate(q4, ride_id = as.character(ride_id)
                  ,rideable_type = as.character(rideable_type))
q3<-  mutate(q3, ride_id = as.character(ride_id)
                  ,rideable_type = as.character(rideable_type))
q2<-  mutate(q2, ride_id = as.character(ride_id)
                  ,rideable_type = as.character(rideable_type))
q1<-  mutate(q1, ride_id = as.character(ride_id)
                  ,rideable_type= as.character(rideable_type))
```

Now lets stack the data frames into one so it is easy to clean

```
all_trips <- bind_rows(q1,q2,q3,q4)
```

Lets take a look at all the columns in our new data frame

```
colnames(all_trips)
```

```
##  [1] "ride_id"
##  [2] "started_at"
##  [3] "ended_at"
##  [4] "rideable_type"
##  [5] "X01...Rental.Details.Duration.In.Seconds.Uncapped"
##  [6] "start_station_id"
##  [7] "start_station_name"
##  [8] "end_station_id"
##  [9] "end_station_name"
## [10] "member_casual"
## [11] "Member.Gender"
## [12] "X05...Member.Details.Member.Birthday.Year"
## [13] "tripduration"
## [14] "gender"
## [15] "birthyear"
## [16] "start_lat"
## [17] "start_lng"
## [18] "end_lat"
## [19] "end_lng"
```

Looks great we just need to trim the columns that are not very useful for us. Lets clean it up

```
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, birthyear, gender,
tripduration, Member.Gender, X05...Member.Details.Member.Birthday.Year,
X01...Rental.Details.Duration.In.Seconds.Uncapped ))
```

Now lets check the data frame

```
colnames(all_trips)
```

```
## [1] "ride_id"            "started_at"         "ended_at"
## [4] "rideable_type"      "start_station_id"   "start_station_name"
## [7] "end_station_id"     "end_station_name"   "member_casual"
```

Great,we now have one data frame with all columns useful and we can now start to clean the data.

```
nrow(all_trips)
```

```
## [1] 3879822
```

We have lots of entries, that is great our sample size is large!

```
dim(all_trips)
```

```
## [1] 3879822       9
```

Lets check the data types to make sure they are what we want before calculation.

```
str(all_trips)
```

```
## 'data.frame':    3879822 obs. of  9 variables:
##  $ ride_id           : chr  "22178529" "22178530" "22178531" "22178532"
...
##  $ started_at        : chr  "2019-04-01 00:02:22" "2019-04-01 00:03:02"
"2019-04-01 00:11:07" "2019-04-01 00:13:01" ...
##  $ ended_at          : chr  "2019-04-01 00:09:48" "2019-04-01 00:20:30"
"2019-04-01 00:15:19" "2019-04-01 00:18:58" ...
##  $ rideable_type     : chr  "6251" "6226" "5649" "4151" ...
##  $ start_station_id  : int  81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name: chr  "Daley Center Plaza" "Wood St & Taylor St"
"LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : int  56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name  : chr  "Desplaines St & Kinzie St" "Wabash Ave &
Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr  "Subscriber" "Subscriber" "Subscriber"
"Subscriber" ...
```

```
summary(all_trips)
```

```
##    ride_id           started_at          ended_at         rideable_type
##  Length:3879822     Length:3879822     Length:3879822     Length:3879822
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  start_station_id start_station_name end_station_id  end_station_name
##  Min.   :  1.0    Length:3879822     Min.   :  1.0   Length:3879822
##  1st Qu.: 77.0    Class :character   1st Qu.: 77.0   Class :character
```

```
##  Median :174.0    Mode  :character   Median :174.0   Mode  :character
##  Mean   :202.9                       Mean   :203.8
##  3rd Qu.:291.0                       3rd Qu.:291.0
##  Max.   :675.0                       Max.   :675.0
##                                      NA's   :1
##  member_casual
##  Length:3879822
##  Class :character
##  Mode  :character
##
##
##
##
```

Lots of interesting things to consider.

Looks like the started_at and ended_at columns are characters, we will want these in a date-time to do some calculation later. In the member_casual column, there theoretically should only have two entries, either casual or member. It seems there is many different types of entries like subscriber, Subscriber, casual, or member. We will need it to be in two categories in order to do our analysis. Rideable type is also some variance in numbers which must represent something like a code). We will need to look into that too.

We will have to add columns to the data frame to find things like trip duration, day of the week etc.

First lets see a table of values from the member_casual column

```
table(all_trips$member_casual)
```

```
##
##     casual   Customer     member Subscriber
##      48480     857474     378407    2595461
```

4 different entries. Easy fix. Since the conversations with Cyclist they referred to the different types of customers as casual and members. So we will change Subscriber to member and Customer to casual

```
all_trips <- all_trips %>%
  mutate(member_casual= recode(member_casual
                               ,"Subscriber"="member"
                               ,"Customer"="casual"))
```

Lets check to see if it changed

```
table(all_trips$member_casual)
```

```
##
##  casual  member
##  905954 2973868
```

Great! Now there is only two different types of customers, easy for comparing the two.

Now lets add columns for date, month, day,and year for each ride.

```
all_trips$date <- as.Date(all_trips$started_at)
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
```

Lets take a quick look to see how it looks.

```
head(all_trips)
```

```
##    ride_id          started_at            ended_at rideable_type
## 1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48          6251
## 2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30          6226
## 3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19          5649
## 4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58          4151
## 5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13          3270
## 6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56          3123
##   start_station_id        start_station_name end_station_id
## 1               81        Daley Center Plaza             56
## 2              317       Wood St & Taylor St             59
## 3              283 LaSalle St & Jackson Blvd            174
## 4               26  McClurg Ct & Illinois St            133
## 5              202      Halsted St & 18th St            129
## 6              420       Ellis Ave & 55th St            426
##            end_station_name member_casual       date month day year
day_of_week
## 1 Desplaines St & Kinzie St        member 2019-04-01    04  01 2019
Monday
## 2 Wabash Ave & Roosevelt Rd        member 2019-04-01    04  01 2019
Monday
## 3     Canal St & Madison St        member 2019-04-01    04  01 2019
Monday
## 4  Kingsbury St & Kinzie St        member 2019-04-01    04  01 2019
Monday
## 5 Blue Island Ave & 18th St        member 2019-04-01    04  01 2019
Monday
## 6       Ellis Ave & 60th St        member 2019-04-01    04  01 2019
Monday
```

Now lets add a ride length column

```
all_trips$ride_length <- difftime(all_trips$ended_at, all_trips$started_at)
```

Now lets take a look at data types

```
str(all_trips)
```

```
## 'data.frame':    3879822 obs. of  15 variables:
##  $ ride_id           : chr  "22178529" "22178530" "22178531" "22178532"
...
##  $ started_at        : chr  "2019-04-01 00:02:22" "2019-04-01 00:03:02"
"2019-04-01 00:11:07" "2019-04-01 00:13:01" ...
##  $ ended_at          : chr  "2019-04-01 00:09:48" "2019-04-01 00:20:30"
"2019-04-01 00:15:19" "2019-04-01 00:18:58" ...
##  $ rideable_type     : chr  "6251" "6226" "5649" "4151" ...
##  $ start_station_id  : int  81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name: chr  "Daley Center Plaza" "Wood St & Taylor St"
"LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : int  56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name  : chr  "Desplaines St & Kinzie St" "Wabash Ave &
Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
##  $ date              : Date, format: "2019-04-01" "2019-04-01" ...
##  $ month             : chr  "04" "04" "04" "04" ...
##  $ day               : chr  "01" "01" "01" "01" ...
##  $ year              : chr  "2019" "2019" "2019" "2019" ...
##  $ day_of_week       : chr  "Monday" "Monday" "Monday" "Monday" ...
##  $ ride_length       : 'difftime' num  446 1048 252 357 ...
##   ..- attr(*, "units")= chr "secs"
```

Looks like we have to convert the ride length to numeric from a factor.

```
is.factor(all_trips$ride_length)
```

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

```
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
```

```
is.numeric(all_trips$ride_length)
```

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

Lets check a summary again.

```
summary(all_trips)
```

```
##    ride_id           started_at          ended_at         rideable_type
##  Length:3879822     Length:3879822     Length:3879822     Length:3879822
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##
##  start_station_id start_station_name end_station_id  end_station_name
##  Min.   :  1.0    Length:3879822     Min.   :  1.0   Length:3879822
##  1st Qu.: 77.0    Class :character   1st Qu.: 77.0   Class :character
##  Median :174.0    Mode  :character   Median :174.0   Mode  :character
##  Mean   :202.9                       Mean   :203.8
##  3rd Qu.:291.0                       3rd Qu.:291.0
##  Max.   :675.0                       Max.   :675.0
##                                      NA's   :1
##  member_casual           date               month               day
##  Length:3879822     Min.   :2019-04-01   Length:3879822     Length:3879822
##  Class :character   1st Qu.:2019-06-23   Class :character   Class
:character
##  Mode  :character   Median :2019-08-14   Mode  :character   Mode
:character
##                     Mean   :2019-08-25
##                     3rd Qu.:2019-10-12
##                     Max.   :2020-03-31
##
##      year           day_of_week         ride_length
##  Length:3879822     Length:3879822     Min.   :  -6982
##  Class :character   Class :character   1st Qu.:    411
##  Mode  :character   Mode  :character   Median :    711
##                                        Mean   :   1478
##                                        3rd Qu.:   1288
##                                        Max.   :9383424
##
```

Interesting. Looks like there are some values for ride length are negative. We can remove the negative values.

There are also some trips where the bikes are taken out for maintenance where the start station name is HQ QR. We can remove these from the data so it does not affect the analysis.

```
all_trips_v2 <- all_trips[!(all_trips$start_station_name=="HQ QR"|
all_trips$ride_length<0),]
```

Now our data should be clean for some analysis. Since some rows were removed I renamed the data frame so it's easier to go back and check the data that was removed.

In saying that. Lets get looking at what the data is telling us.

```
is.Date(all_trips_v2$started_at)
```

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

Lets make it a datetime format

```
all_trips_v2$started_at <- ymd_hms(all_trips_v2$started_at)

all_trips_v2$ended_at <- ymd_hms(all_trips_v2$ended_at)

class(all_trips_v2$started_at)

## [1] "POSIXct" "POSIXt"

class(all_trips_v2$ended_at)

## [1] "POSIXct" "POSIXt"
```

Now lets make a 'start_time' and 'end_time' column.

```
all_trips_v2$start_time <- format(all_trips_v2$started_at, format =
"%H:%M:%S")
all_trips_v2$start_time <- as.POSIXct(all_trips_v2$start_time, format =
"%H:%M:%S")

all_trips_v2$end_time <- format(all_trips_v2$ended_at, format = "%H:%M:%S")
all_trips_v2$end_time <- as.POSIXct(all_trips_v2$end_time, format =
"%H:%M:%S")
```

Analysis

Lets get an idea how the ride lengths average, min and max values.

```
summary(all_trips_v2$ride_length)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     412     712    1479    1289 9383424
```

Note this is in seconds. So on mean trip length is 1479 seconds or just over 24 and a half minutes.

Lets see how the times are different bases on the membership type.

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)

##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                3552.7941
## 2                     member                 850.0783

aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN =
median)

##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                      1546
## 2                     member                       589

aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                  9383424
## 2                     member                  9056634
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                        2
## 2                     member                        1
```

Looks like casuals have longer average and median ride times compared to members. Interesting, lets see how this interacts with day of the week.

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual +
all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week
all_trips_v2$ride_length
## 1                      casual                   Friday
3773.8351
## 2                      member                   Friday
824.5385
## 3                      casual                   Monday
3372.2869
## 4                      member                   Monday
842.5649
## 5                      casual                 Saturday
3331.8795
## 6                      member                 Saturday
968.9962
## 7                      casual                   Sunday
3581.5047
## 8                      member                   Sunday
920.0284
## 9                      casual                 Thursday
3683.0548
## 10                     member                 Thursday
823.9278
## 11                     casual                  Tuesday
3596.3599
## 12                     member                  Tuesday
826.1498
## 13                     casual                Wednesday
3718.8955
## 14                     member                Wednesday
823.9996
```

Looks like the days are out of order, lets fix that.

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week,
levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday"))

aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual +
all_trips_v2$day_of_week, FUN = mean)

##    all_trips_v2$member_casual all_trips_v2$day_of_week
all_trips_v2$ride_length
## 1                      casual                   Sunday
3581.5047
## 2                      member                   Sunday
920.0284
## 3                      casual                   Monday
3372.2869
## 4                      member                   Monday
842.5649
## 5                      casual                  Tuesday
3596.3599
## 6                      member                  Tuesday
826.1498
## 7                      casual                Wednesday
3718.8955
## 8                      member                Wednesday
823.9996
## 9                      casual                 Thursday
3683.0548
## 10                     member                 Thursday
823.9278
## 11                     casual                   Friday
3773.8351
## 12                     member                   Friday
824.5385
## 13                     casual                 Saturday
3331.8795
## 14                     member                 Saturday
968.9962
```

On average, the trip length is longer for everyday for the casual riders.

Lets continue to investigate.

```
all_trips_v2 %>%
  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)
```

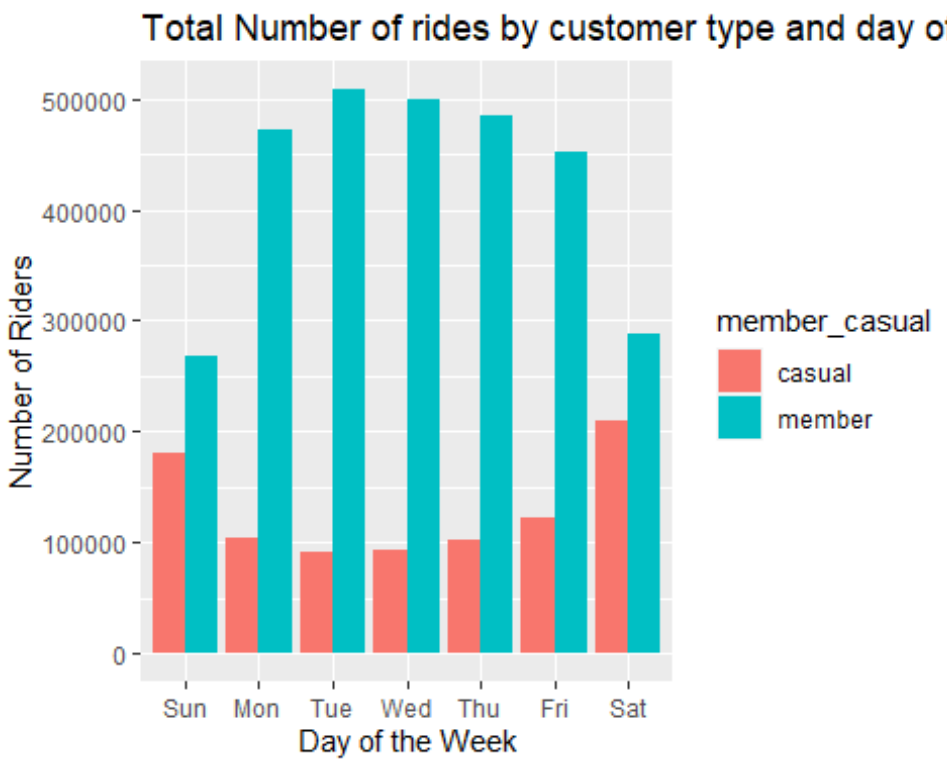
```
## `summarise()` has grouped output by 'member_casual'. You can override
using the `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              181293            3582.
##  2 casual        Mon              103296            3372.
##  3 casual        Tue               90510            3596.
##  4 casual        Wed               92457            3719.
##  5 casual        Thu              102679            3683.
##  6 casual        Fri              122404            3774.
##  7 casual        Sat              209543            3332.
##  8 member        Sun              267965             920.
##  9 member        Mon              472196             843.
## 10 member        Tue              508445             826.
## 11 member        Wed              500329             824.
## 12 member        Thu              484177             824.
## 13 member        Fri              452790             825.
## 14 member        Sat              287958             969.
```

Lets make a quick visualization to get an idea of whats going on before we take the visualization to tableau.

```
all_trips_v2 %>%
  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)) +
  geom_col(position = "dodge") +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))+
  labs(title = "Total Number of rides by customer type and day of the week",
y= "Number of Riders", x= "Day of the Week")
```

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

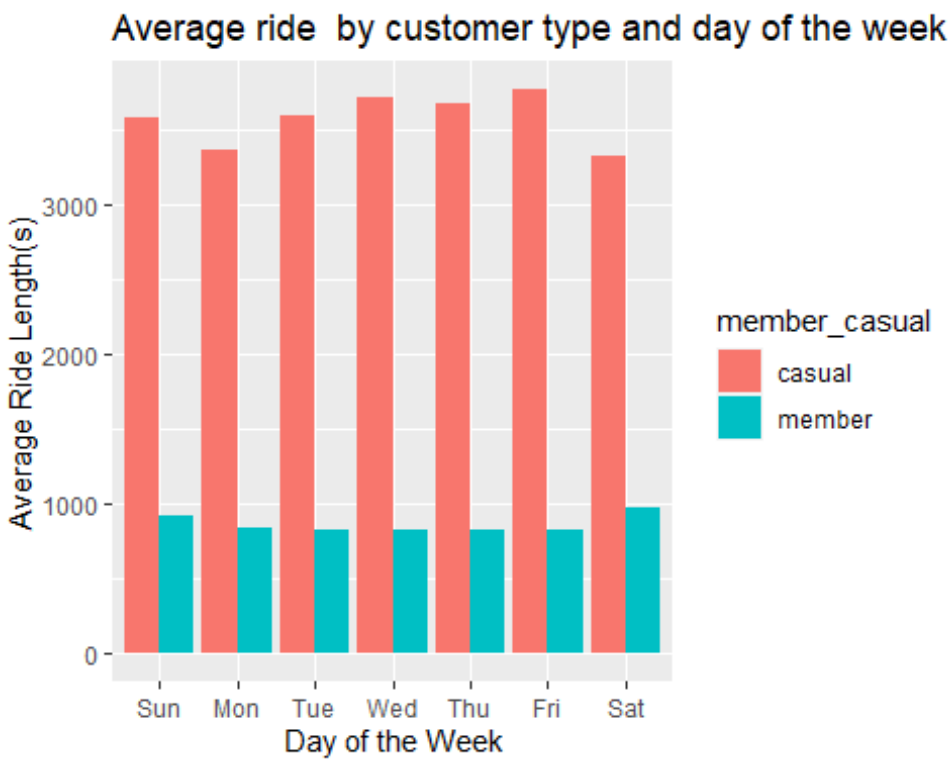


Seems like members use the bikes during the week, less on weekends and causal are the exact opposite.

Lets create a visual based on average ride duration

```
all_trips_v2 %>%
  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 = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(title = "Average ride  by customer type and day of the week", x= "Day
of the Week", y= "Average Ride Length(s)")
```

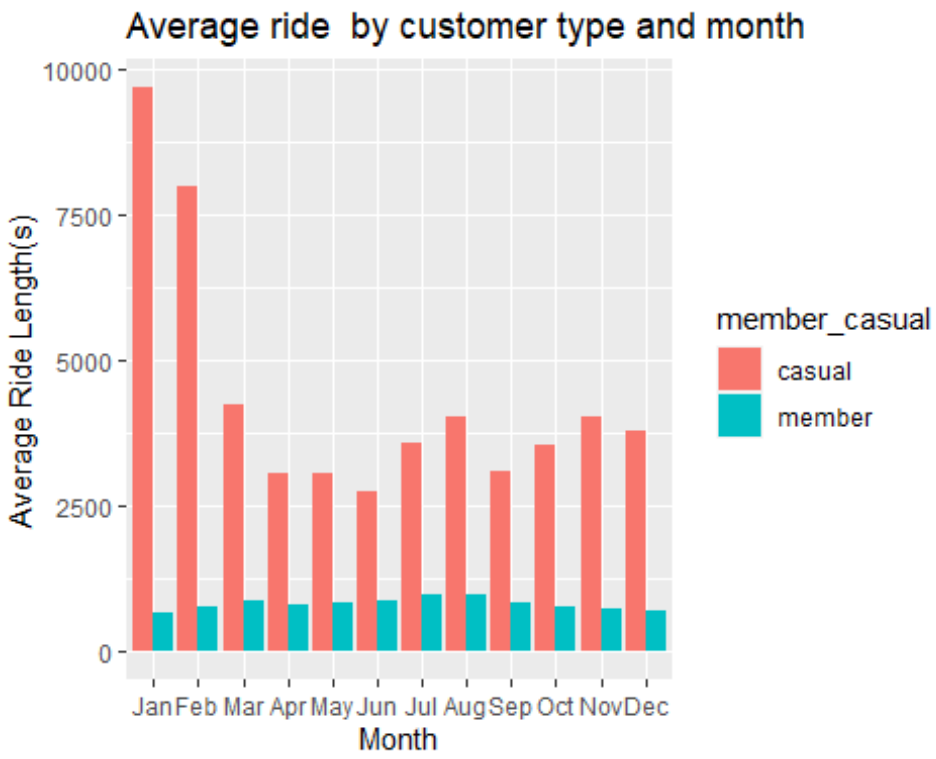
```
## `summarise()` has grouped output by 'member_casual'. You can override
using the `.groups` argument.
```



So it's clear the average ride time for casuals is higher than members on every day of the week. There could be reasons for this, but lets consider the month of the year and see if anything else comes to light.

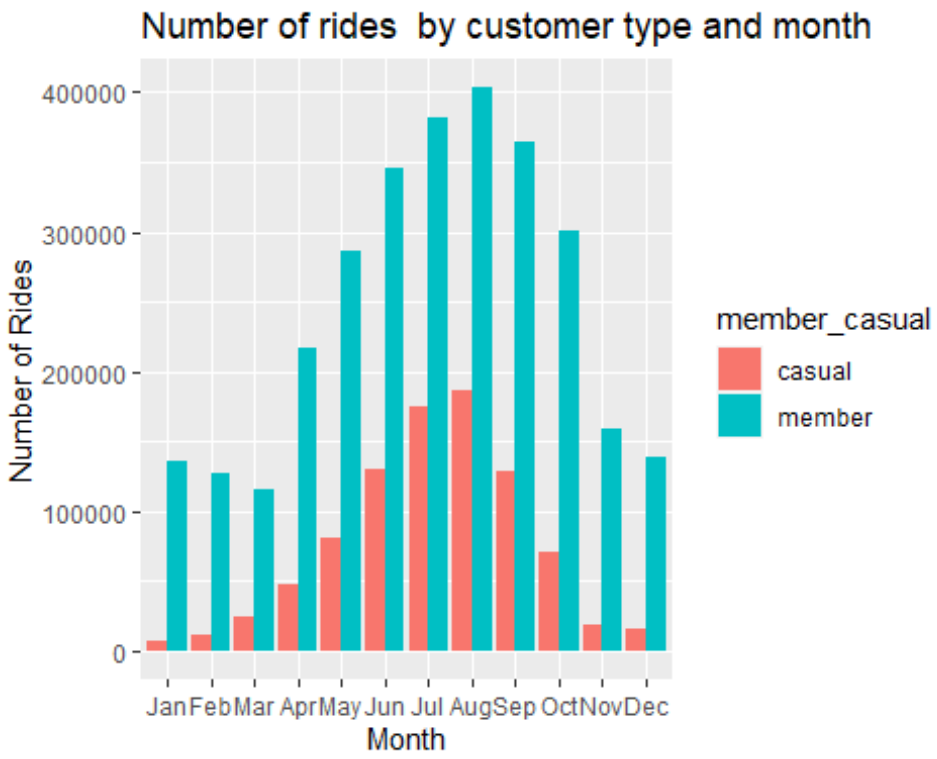
```
all_trips_v2 %>%
  mutate(month = month(started_at, label = TRUE)) %>%
  group_by(member_casual, month) %>%
  summarise(average_duration = mean(ride_length)) %>%
  arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")+
  labs(title = "Average ride  by customer type and month", x= "Month", y=
"Average Ride Length(s)")

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




```
all_trips_v2 %>%
  mutate(month = month(started_at, label=TRUE)) %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()) %>%
    arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))+
  labs(title = "Number of rides  by customer type and month", x= "Month", y=
"Number of Rides")
```

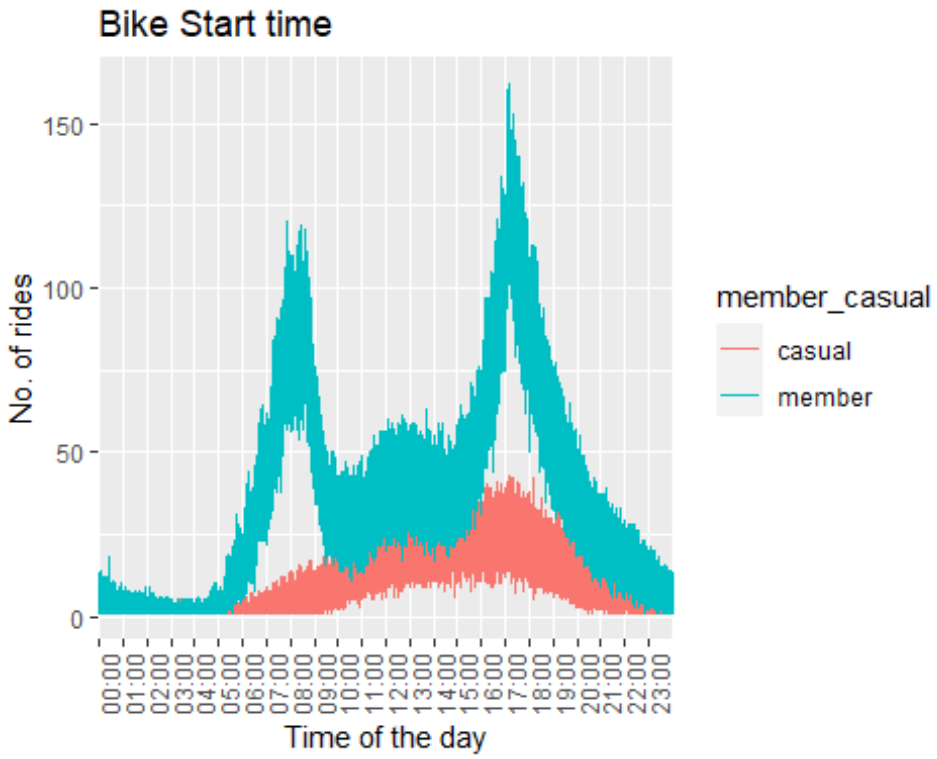
```
## `summarise()` has grouped output by 'member_casual'. You can override
using the `.groups` argument.
```



So it seems the average ride duration is higher for the casuals all year round, but their is more consistent number of rides for members year round. With the summer months being more popular for both customer types, which makes with the snow in the winter months. It also seems to be the case that members are more consistent throughout the year which may indicate them using the bikes to commute to work. Lets the the average start times and see if this is the case.

```
all_trips_v2 %>%
  group_by(member_casual, start_time) %>%
  summarise(number_of_rides = n()) %>%
ggplot(aes(x = start_time, y = number_of_rides, color = member_casual, group
= member_casual))+
  geom_line() +
  scale_x_datetime(date_breaks = "1 hour", minor_breaks = NULL,
                   date_labels = "%H:%M", expand = c(0,0))+
  labs(title ="Bike Start time", x = "Time of the day", y = "No. of rides")+
  theme(axis.text.x = element_text(angle = 90))
```

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



#Act It appears members are taking out bikes mostly from 7-9 am and the again from 4-7 pm. This would back up the prediction of members using the bikes to commute to work. The casuals use the bikes mostly from 4-7 pm, similar to members, but no influx in the morning. This may suggest casuals using the bikes as an outlet for exercise. THe members may be doing this as well.

So there was a lot of information we have gathered here.

Casuals- Rides tend to be longer and during the warmer months. Hot times for casuals are from 4-7pm. This hints to using the bikes for exercise and for leisure.

Members- Rides are shorter than casuals and are consistent throughout the year (yes it is still higher during the warmer months). Hot times for the members are from 7-9 am and 4-7 pm. This indicates members are using the bikes as a means of transportation to commute to work.

Since the goal of the stakeholders was to understand how to convert casuals to members in order to maximize their profit. This could be done in numerous ways. Perhaps advertising in high density areas like subways and bus stops to show how people can use the bikes as a means to commute to work. Perhaps its saving money, like it's cheaper than public transit or cheaper than a gym membership will convert the casual.

These options would be presented to the stakeholders but the analysis is primarily for them to understand how casuals and members use the bikes differently.

#Extension Analaysis for Stake Holders

Possibility for further exploration would be to check the stations that are popular with members, with casuals, and the ones that are not. This may give the stake holders a better idea where their members are using their bikes and to see if there are high traffic areas with more casuals than members.

Here is the amount of trips between each station to see for a heat map in future. This was not asked by the stake holders but if I was in communication with them, I would definitely ask if this would help them target advertising in certain places/stations. I would also require a map of Chicago and the points of their stations to help show traffic flow and most popular stations based on the membership type.

```
all_trips_v2 %>%
 group_by( start_station_name, end_station_name, member_casual) %>%
  summarise(number_of_rides = n()   ) %>%
  arrange (start_station_name, end_station_name, member_casual)
```

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

```
 all_trips_v2 %>%
 group_by( start_station_name, end_station_name ) %>%
  summarise(number_of_rides = n()   ) %>%
  arrange (desc(number_of_rides))
```

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

```
#write.csv(station_data, file=
"C:/Users/baleo/OneDrive/Documents/Cyclist/station_traffic_data.csv")
```