MyCaseStudy_Notebook

Abdoul Fall

05/09/2021

Case Study 1: How Does a Bike-Share Navigate Speedy Success?

Step 1: ASK

Mission Statement

Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago.the 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.

Key Objectives

- **Business Task**

  Identifying patterns, trends and connections in Cyclistic's historical data in order to gain and generate key insights into how casual bike riders differ from annual members of Cyclistic to attract those casual riders and convert them to annual members by showing the potential benefits of an annual memberships , which will improve the growth of Cyclistic.

- **Key Stakeholders**

  Cyclistic executive team and Lily Moreno, the director of marketing

Step 2: PREPARE

Key Objectives

- **organization and credibility of the data**

The data is organized in long format. There are no issues with bias or credibility with the data as it is reliable, coming from Motivate International Inc with a data license, the data is original as it comes from a second party source, it is also comprehensible as the data contains the necessary information we need to do the analysis. It is also current as the data were collected from April 2020 to March 2021. The data is cited as among the content, the source of the data is stated with the dates in which the survey was performed. Therefore, the data is credible and there is no bias as the data is a vetted public data.The data include information about the riders' ride time when they started and when they ended their rides in the week.

- **Sort and filter the data**

The this part of the analysis will consist of making the type of the columns of the data the same so we could bind and merge them together and look for any incongruencies in the data.

```
library("tidyverse")
## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
library("ggplot2")
library("lubridate")
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
q2_2019 <- read_csv("Divvy_Trips_2019_Q2.csv")
##
## -- Column specification --------------------------------------------------
------
## cols(
##   `01 - Rental Details Rental ID` = col_double(),
##   `01 - Rental Details Local Start Time` = col_datetime(format = ""),
##   `01 - Rental Details Local End Time` = col_datetime(format = ""),
##   `01 - Rental Details Bike ID` = col_double(),
##   `01 - Rental Details Duration In Seconds Uncapped` = col_number(),
##   `03 - Rental Start Station ID` = col_double(),
##   `03 - Rental Start Station Name` = col_character(),
##   `02 - Rental End Station ID` = col_double(),
##   `02 - Rental End Station Name` = col_character(),
##   `User Type` = col_character(),
##   `Member Gender` = col_character(),
##   `05 - Member Details Member Birthday Year` = col_double()
## )
q3_2019 <- read_csv("Divvy_Trips_2019_Q3.csv")
##
```

```
## -- Column specification --------------------------------------------------
------
## 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()
## )
q4_2019 <- read_csv("Divvy_Trips_2019_Q4.csv")
##
## -- Column specification --------------------------------------------------
------
## 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()
## )
q1_2020 <- read_csv("Divvy_Trips_2020_Q1.csv")
##
## -- Column specification --------------------------------------------------
------
## cols(
##   ride_id = col_character(),
##   rideable_type = col_character(),
##   started_at = col_datetime(format = ""),
##   ended_at = col_datetime(format = ""),
##   start_station_name = col_character(),
##   start_station_id = col_double(),
##   end_station_name = col_character(),
##   end_station_id = col_double(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
colnames(q2_2019)
##  [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"
colnames(q3_2019)
##  [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_2019)
##  [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(q1_2020)
##  [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"
```

Renaming columns to make them consistent with q1_2020

```
q4_2019 <- q4_2019 %>%
  rename(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_2019 <- q3_2019 %>%
  rename(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)

q2_2019 <- q2_2019 %>%
  rename(ride_id = "01 - Rental Details Rental ID"
        ,rideable_type = "01 - Rental Details Bike ID"
        ,started_at = "01 - Rental Details Local Start Time"
        ,ended_at = "01 - Rental Details Local End Time"
        ,start_station_name = "03 - Rental Start Station Name"
        ,start_station_id = "03 - Rental Start Station ID"
        ,end_station_name = "02 - Rental End Station Name"
        ,end_station_id = "02 - Rental End Station ID"
        ,member_casual = "User Type")
```

Converting data type to character to be consistent with q1_2020

```
q4_2019 <- q2_2019 %>%
  mutate(ride_id = as.character(ride_id)
        ,rideable_type = as.character(rideable_type))
q3_2019 <- q3_2019 %>%
  mutate(q3_2019, ride_id = as.character(ride_id)
        ,rideable_type = as.character(rideable_type))
q2_2019 <- q2_2019 %>%
  mutate(q2_2019, ride_id = as.character(ride_id)
        ,rideable_type = as.character(rideable_type))
```

Inspecting the data and look for incongruencies and consistency

```
str(q1_2020)
## spec_tbl_df[,13] [426,887 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426887] "EACB19130B0CDA4A"
"8FED874C809DC021" "789F3C21E472CA96" "C9A388DAC6ABF313" ...
##  $ rideable_type     : chr [1:426887] "docked_bike" "docked_bike"
"docked_bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:426887], format: "2020-01-21 20:06:59"
"2020-01-30 14:22:39" ...
##  $ ended_at          : POSIXct[1:426887], format: "2020-01-21 20:14:30"
"2020-01-30 14:26:22" ...
##  $ start_station_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St
& Montrose Ave" "Broadway & Belmont Ave" "Clark St & Randolph St" ...
##  $ start_station_id  : num [1:426887] 239 234 296 51 66 212 96 96 212 38
...
##  $ end_station_name  : chr [1:426887] "Clark St & Leland Ave" "Southport
Ave & Irving Park Rd" "Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave"
...
##  $ end_station_id    : num [1:426887] 326 318 117 24 212 96 212 212 96 100
...
##  $ start_lat         : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:426887] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:426887] -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:426887] "member" "member" "member" "member"
...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
str(q4_2019)
## spec_tbl_df[,12] [1,108,163 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id                                         : chr [1:1108163]
"22178529" "22178530" "22178531" "22178532" ...
##  $ started_at                                      : POSIXct[1:1108163],
format: "2019-04-01 00:02:22" "2019-04-01 00:03:02" ...
##  $ ended_at                                        : POSIXct[1:1108163],
format: "2019-04-01 00:09:48" "2019-04-01 00:20:30" ...
##  $ rideable_type                                   : chr [1:1108163]
"6251" "6226" "5649" "4151" ...
##  $ 01 - Rental Details Duration In Seconds Uncapped: num [1:1108163] 446
1048 252 357 1007 ...
##  $ start_station_id                                : num [1:1108163] 81
317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name                              : chr [1:1108163]
"Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd"
"McClurg Ct & Illinois St" ...
##  $ end_station_id                                  : num [1:1108163] 56 59
174 133 129 426 500 499 211 211 ...
##  $ end_station_name                                : chr [1:1108163]
"Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison
St" "Kingsbury St & Kinzie St" ...
##  $ member_casual                                   : chr [1:1108163]
"Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ Member Gender                                   : chr [1:1108163]
"Male" "Female" "Male" "Male" ...
##  $ 05 - Member Details Member Birthday Year        : num [1:1108163] 1975
1984 1990 1993 1992 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `01 - Rental Details Rental ID` = col_double(),
##   ..   `01 - Rental Details Local Start Time` = col_datetime(format = ""),
##   ..   `01 - Rental Details Local End Time` = col_datetime(format = ""),
##   ..   `01 - Rental Details Bike ID` = col_double(),
##   ..   `01 - Rental Details Duration In Seconds Uncapped` = col_number(),
##   ..   `03 - Rental Start Station ID` = col_double(),
##   ..   `03 - Rental Start Station Name` = col_character(),
##   ..   `02 - Rental End Station ID` = col_double(),
##   ..   `02 - Rental End Station Name` = col_character(),
##   ..   `User Type` = col_character(),
##   ..   `Member Gender` = col_character(),
##   ..   `05 - Member Details Member Birthday Year` = col_double()
##   .. )
str(q3_2019)
## spec_tbl_df[,12] [1,640,718 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1640718] "23479388" "23479389" "23479390"
"23479391" ...
##  $ started_at        : POSIXct[1:1640718], format: "2019-07-01 00:00:27"
"2019-07-01 00:01:16" ...
##  $ ended_at          : POSIXct[1:1640718], format: "2019-07-01 00:20:41"
"2019-07-01 00:18:44" ...
##  $ rideable_type     : chr [1:1640718] "3591" "5353" "6180" "5540" ...
##  $ tripduration      : num [1:1640718] 1214 1048 1554 1503 1213 ...
##  $ start_station_id  : num [1:1640718] 117 381 313 313 168 300 168 313 43
43 ...
##  $ start_station_name: chr [1:1640718] "Wilton Ave & Belmont Ave" "Western
Ave & Monroe St" "Lakeview Ave & Fullerton Pkwy" "Lakeview Ave & Fullerton
Pkwy" ...
##  $ end_station_id    : num [1:1640718] 497 203 144 144 62 232 62 144 195
195 ...
##  $ end_station_name  : chr [1:1640718] "Kimball Ave & Belmont Ave"
"Western Ave & 21st St" "Larrabee St & Webster Ave" "Larrabee St & Webster
Ave" ...
##  $ member_casual     : chr [1:1640718] "Subscriber" "Customer" "Customer"
"Customer" ...
##  $ gender            : chr [1:1640718] "Male" NA NA NA ...
##  $ birthyear         : num [1:1640718] 1992 NA NA NA NA ...
##  - 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()
##   .. )
str(q2_2019)
## spec_tbl_df[,12] [1,108,163 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id                                        : chr [1:1108163]
"22178529" "22178530" "22178531" "22178532" ...
##  $ started_at                                     : POSIXct[1:1108163],
format: "2019-04-01 00:02:22" "2019-04-01 00:03:02" ...
##  $ ended_at                                       : POSIXct[1:1108163],
format: "2019-04-01 00:09:48" "2019-04-01 00:20:30" ...
##  $ rideable_type                                  : chr [1:1108163]
"6251" "6226" "5649" "4151" ...
##  $ 01 - Rental Details Duration In Seconds Uncapped: num [1:1108163] 446
1048 252 357 1007 ...
##  $ start_station_id                               : num [1:1108163] 81
317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name                             : chr [1:1108163]
"Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd"
"McClurg Ct & Illinois St" ...
##  $ end_station_id                                 : num [1:1108163] 56 59
174 133 129 426 500 499 211 211 ...
##  $ end_station_name                               : chr [1:1108163]
"Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison
St" "Kingsbury St & Kinzie St" ...
##  $ member_casual                                  : chr [1:1108163]
"Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ Member Gender                                  : chr [1:1108163]
"Male" "Female" "Male" "Male" ...
```

```
##  $ 05 - Member Details Member Birthday Year        : num [1:1108163] 1975
1984 1990 1993 1992 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `01 - Rental Details Rental ID` = col_double(),
##   ..   `01 - Rental Details Local Start Time` = col_datetime(format = ""),
##   ..   `01 - Rental Details Local End Time` = col_datetime(format = ""),
##   ..   `01 - Rental Details Bike ID` = col_double(),
##   ..   `01 - Rental Details Duration In Seconds Uncapped` = col_number(),
##   ..   `03 - Rental Start Station ID` = col_double(),
##   ..   `03 - Rental Start Station Name` = col_character(),
##   ..   `02 - Rental End Station ID` = col_double(),
##   ..   `02 - Rental End Station Name` = col_character(),
##   ..   `User Type` = col_character(),
##   ..   `Member Gender` = col_character(),
##   ..   `05 - Member Details Member Birthday Year` = col_double()
##   .. )
```

Stacking individual quarter’s data frames into one big data frame

```
all_bike_trips <- bind_rows(q2_2019,q3_2019, q4_2019, q1_2020)
```

Removing columns are not needed for the analyze phase (Remove lat, long, birthyear, and gender fields)

```
all_bike_trips <- all_bike_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, birthyear, gender
            , "01 - Rental Details Duration In Seconds Uncapped"
            , "05 - Member Details Member Birthday Year", "Member Gender"
            , "tripduration"))
```

Step 3: PROCESS

Inspecting the new table that has been created

```
colnames(all_bike_trips)   #column names are listed
## [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"
nrow(all_bike_trips)       #number of rows in the data frame
## [1] 4283931
dim(all_bike_trips)        #dimensions of the data frame
## [1] 4283931       9
head(all_bike_trips)       #the six rows of the data frame are listed
## # A tibble: 6 x 9
##   ride_id started_at          ended_at            rideable_type
start_station_id
##   <chr>   <dttm>              <dttm>              <chr>
<dbl>
## 1 221785~ 2019-04-01 00:02:22 2019-04-01 00:09:48 6251
81
## 2 221785~ 2019-04-01 00:03:02 2019-04-01 00:20:30 6226
317
```

```
## 3 221785~ 2019-04-01 00:11:07 2019-04-01 00:15:19 5649
283
## 4 221785~ 2019-04-01 00:13:01 2019-04-01 00:18:58 4151
26
## 5 221785~ 2019-04-01 00:19:26 2019-04-01 00:36:13 3270
202
## 6 221785~ 2019-04-01 00:19:39 2019-04-01 00:23:56 3123
420
## # ... with 4 more variables: start_station_name <chr>, end_station_id
<dbl>,
## #   end_station_name <chr>, member_casual <chr>
str(all_bike_trips)            #columns and data types are listed
## tibble[,9] [4,283,931 x 9] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:4283931] "22178529" "22178530" "22178531"
"22178532" ...
##  $ started_at        : POSIXct[1:4283931], format: "2019-04-01 00:02:22"
"2019-04-01 00:03:02" ...
##  $ ended_at          : POSIXct[1:4283931], format: "2019-04-01 00:09:48"
"2019-04-01 00:20:30" ...
##  $ rideable_type     : chr [1:4283931] "6251" "6226" "5649" "4151" ...
##  $ start_station_id  : num [1:4283931] 81 317 283 26 202 420 503 260 211
211 ...
##  $ start_station_name: chr [1:4283931] "Daley Center Plaza" "Wood St &
Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : num [1:4283931] 56 59 174 133 129 426 500 499 211
211 ...
##  $ end_station_name  : chr [1:4283931] "Desplaines St & Kinzie St" "Wabash
Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr [1:4283931] "Subscriber" "Subscriber"
"Subscriber" "Subscriber" ...
summary(all_bike_trips)    #statistical summary of the data
##    ride_id            started_at                     ended_at
##  Length:4283931     Min.   :2019-04-01 00:02:22   Min.   :2019-04-01
00:09:48
##  Class :character   1st Qu.:2019-05-25 04:11:37   1st Qu.:2019-05-25
06:01:22
##  Mode  :character   Median :2019-06-28 20:10:29   Median :2019-06-28
20:33:07
##                     Mean   :2019-07-20 18:37:19   Mean   :2019-07-20
19:02:05
##                     3rd Qu.:2019-08-23 17:10:58   3rd Qu.:2019-08-23
17:32:21
##                     Max.   :2020-03-31 23:51:34   Max.   :2020-05-19
20:10:34
##
##  rideable_type      start_station_id start_station_name end_station_id
##  Length:4283931     Min.   :  1.0    Length:4283931     Min.   :  1.0
##  Class :character   1st Qu.: 77.0    Class :character   1st Qu.: 77.0
##  Mode  :character   Median :174.0    Mode  :character   Median :174.0
##                     Mean   :202.1                       Mean   :203.1
##                     3rd Qu.:289.0                       3rd Qu.:291.0
##                     Max.   :675.0                       Max.   :675.0
##                                                         NA's   :1
##  end_station_name   member_casual
##  Length:4283931     Length:4283931
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

```
##
##
##
##
```

In the “member_casual” column, there are two names for members (“member” and “Subscriber”) and two names for casual riders (“Customer” and “casual”). We will need to replace “Subscriber” with “member” and “Customer” with “casual”.

```
table(all_bike_trips$member_casual)
##
##     casual   Customer     member Subscriber
##      48480    1010866     378407    2846178
```

Reassigning the values to their correct usertype: Changing Subsriber to member & Customer to casual

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

Check to make sure the proper number of observations were reassigned

```
table(all_bike_trips$member_casual)
##
##  casual  member
## 1059346 3224585
```

adding some additional columns of data such as day, month, year to provide additional opportunities to aggregate the data.

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

Adding a ride_length column that will calculate the length of the ride for each bikers for all trips in seconds

```
all_bike_trips$ride_length <- difftime(all_bike_trips$ended_at
                                       ,all_bike_trips$started_at)
str(all_bike_trips)
## tibble[,15] [4,283,931 x 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id          : chr [1:4283931] "22178529" "22178530" "22178531"
"22178532" ...
##  $ started_at       : POSIXct[1:4283931], format: "2019-04-01 00:02:22"
"2019-04-01 00:03:02" ...
##  $ ended_at         : POSIXct[1:4283931], format: "2019-04-01 00:09:48"
"2019-04-01 00:20:30" ...
##  $ rideable_type    : chr [1:4283931] "6251" "6226" "5649" "4151" ...
##  $ start_station_id  : num [1:4283931] 81 317 283 26 202 420 503 260 211
211 ...
##  $ start_station_name: chr [1:4283931] "Daley Center Plaza" "Wood St &
Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...
##  $ end_station_id    : num [1:4283931] 56 59 174 133 129 426 500 499 211
211 ...
##  $ end_station_name  : chr [1:4283931] "Desplaines St & Kinzie St" "Wabash
Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual     : chr [1:4283931] "member" "member" "member" "member"
...
##  $ date              : Date[1:4283931], format: "2019-04-01" "2019-04-01"
...
##  $ month             : chr [1:4283931] "04" "04" "04" "04" ...
##  $ day               : chr [1:4283931] "01" "01" "01" "01" ...
##  $ year              : chr [1:4283931] "2019" "2019" "2019" "2019" ...
##  $ day_of_week       : chr [1:4283931] "Monday" "Monday" "Monday" "Monday"
...
##  $ ride_length       : 'difftime' num [1:4283931] 446 1048 252 357 ...
##   ..- attr(*, "units")= chr "secs"
```

Converting the column “ride_length” from factor to numeric in order to do some calculations on the data

```
is.factor(all_bike_trips$ride_length)
## [1] FALSE
all_bike_trips$ride_length <-
as.numeric(as.character(all_bike_trips$ride_length))
is.numeric(all_bike_trips$ride_length)
## [1] TRUE
```

The data frame includes a few hundred entries when bikes were taken out of docks and checked for quality by Divvy or ride_length was negative. Removing maintenance trips and trips less with no length from the dataset. Also, create a new version of the data frame (v2) since data is being removed.

```
all_bike_trips_v2 <- all_bike_trips[!(all_bike_trips$start_station_name ==
"HQ QR" | all_bike_trips$ride_length<0),]
```

Step 4: ANALYZE

Performing a descriptive analysis on the ride_length column to determine the mean (average total ride length/rides), median (midpoint number of ride lengths), the maximum ride or longest ride and the minimum ride or shortest ride.

```
mean(all_bike_trips_v2$ride_length)
## [1] 1487.465
median(all_bike_trips_v2$ride_length)
## [1] 745
max(all_bike_trips_v2$ride_length)
## [1] 9387024
min(all_bike_trips_v2$ride_length)
## [1] 1
```

```
summary(all_bike_trips_v2$ride_length)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1     428     745    1487    1353 9387024
```

This step will consist of aggregating the mean, median, max and min of the ride length for both members and casual users and compare them.

```
aggregate(all_bike_trips_v2$ride_length ~ all_bike_trips_v2$member_casual,
FUN = mean)
##   all_bike_trips_v2$member_casual all_bike_trips_v2$ride_length
## 1                          casual                     3383.6429
## 2                          member                      866.7436
aggregate(all_bike_trips_v2$ride_length ~ all_bike_trips_v2$member_casual,
FUN = median)
##   all_bike_trips_v2$member_casual all_bike_trips_v2$ride_length
## 1                          casual                          1595
## 2                          member                           607
aggregate(all_bike_trips_v2$ride_length ~ all_bike_trips_v2$member_casual,
FUN = max)
##   all_bike_trips_v2$member_casual all_bike_trips_v2$ride_length
## 1                          casual                       9387024
## 2                          member                       9056634
aggregate(all_bike_trips_v2$ride_length ~ all_bike_trips_v2$member_casual,
FUN = min)
##   all_bike_trips_v2$member_casual all_bike_trips_v2$ride_length
## 1                          casual                             2
## 2                          member                             1
```

Next up, we will analyze the length of ride by days of the week, first with the average ride time by each day for members compared to casual users.

```
aggregate(all_bike_trips_v2$ride_length ~ all_bike_trips_v2$member_casual +
all_bike_trips_v2$day_of_week, FUN = mean)
##    all_bike_trips_v2$member_casual all_bike_trips_v2$day_of_week
## 1                           casual                        Friday
## 2                           member                        Friday
## 3                           casual                        Monday
## 4                           member                        Monday
## 5                           casual                      Saturday
## 6                           member                      Saturday
## 7                           casual                        Sunday
## 8                           member                        Sunday
## 9                           casual                      Thursday
## 10                          member                      Thursday
## 11                          casual                       Tuesday
## 12                          member                       Tuesday
## 13                          casual                     Wednesday
## 14                          member                     Wednesday
##    all_bike_trips_v2$ride_length
## 1                      3539.5701
## 2                       847.3562
## 3                      3280.1548
## 4                       869.5307
## 5                      3233.5718
## 6                       963.6353
## 7                       3391.5910
## 8                        952.1544
## 9                       3544.2240
## 10                       843.1404
## 11                      3363.5863
## 12                       827.0632
## 13                      3471.4699
## 14                       841.2950
```

The days of the week are out of order, so a fix needs to be made in order to make the data more readable and consistent.

```
all_bike_trips_v2$day_of_week <- ordered(all_bike_trips_v2$day_of_week,
levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday"))
```

Then we will run the code again to make sure that the fix has be made and that the days of the week are now in order.

```
aggregate(all_bike_trips_v2$ride_length ~ all_bike_trips_v2$member_casual +
all_bike_trips_v2$day_of_week, FUN = mean)
##    all_bike_trips_v2$member_casual all_bike_trips_v2$day_of_week
## 1                           casual                        Sunday
## 2                           member                        Sunday
## 3                           casual                        Monday
## 4                           member                        Monday
## 5                           casual                       Tuesday
## 6                           member                       Tuesday
## 7                           casual                     Wednesday
## 8                           member                     Wednesday
## 9                           casual                      Thursday
## 10                          member                      Thursday
## 11                          casual                        Friday
## 12                          member                        Friday
## 13                          casual                      Saturday
## 14                          member                      Saturday
##    all_bike_trips_v2$ride_length
## 1                      3391.5910
## 2                       952.1544
## 3                      3280.1548
## 4                       869.5307
## 5                      3363.5863
## 6                       827.0632
## 7                      3471.4699
## 8                       841.2950
## 9                      3544.2240
## 10                      843.1404
## 11                     3539.5701
## 12                      847.3562
## 13                     3233.5718
## 14                      963.6353
```

From this aggregation of length of ride consisting of the average ride length for both members and casual users for each day of the week, we can see that casual users have a considerable higher average ride length each day of the week compared to the annual members.

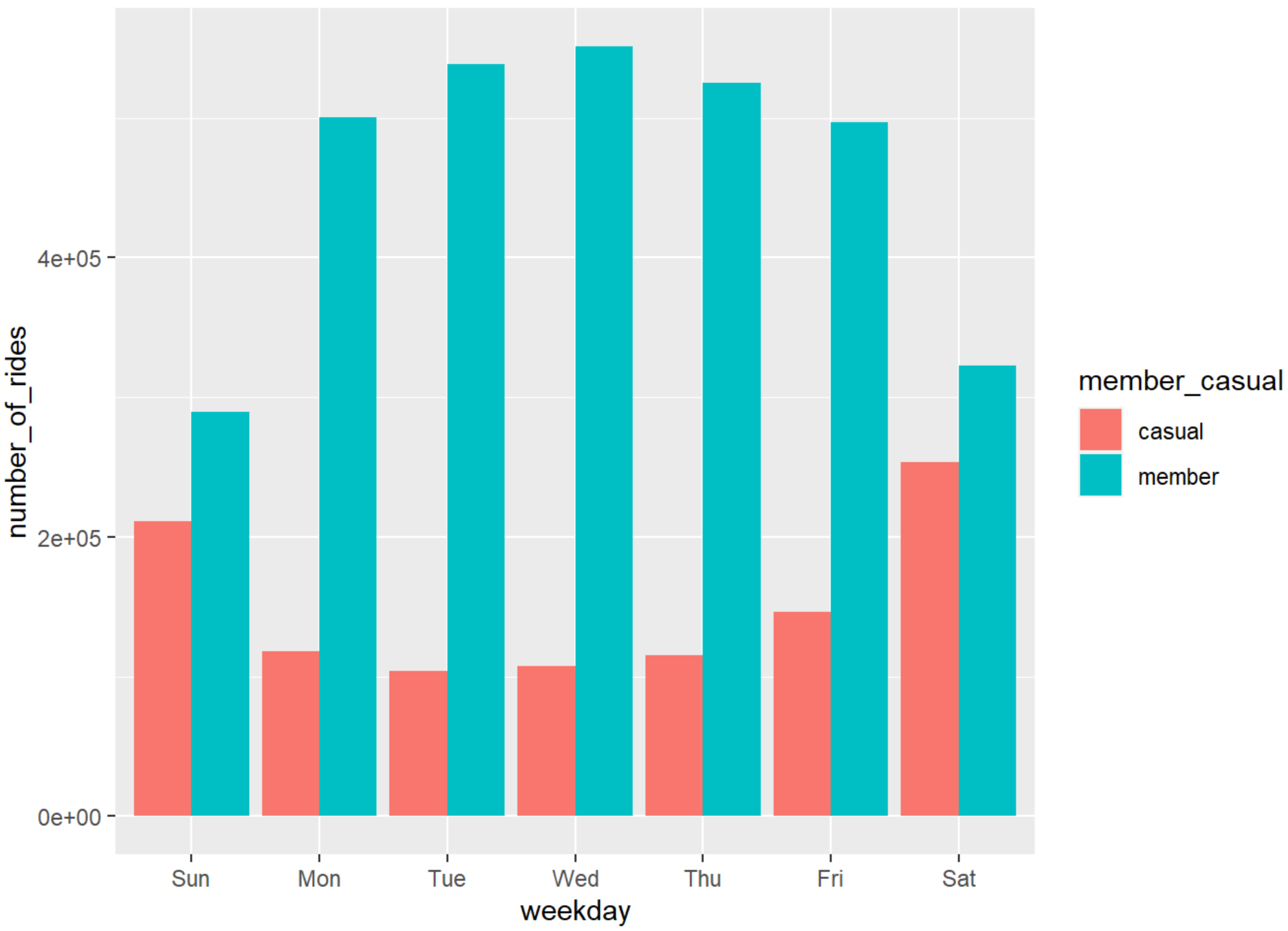
The next step in this analyze phase will consist of analyzing ridership data by type and weekday.

```
all_bike_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)
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              211298            3392.
##  2 casual        Mon              118083            3280.
##  3 casual        Tue              104035            3364.
##  4 casual        Wed              107131            3471.
##  5 casual        Thu              115244            3544.
##  6 casual        Fri              146109            3540.
##  7 casual        Sat              253680            3234.
##  8 member        Sun              289531             952.
##  9 member        Mon              500639             870.
## 10 member        Tue              538188             827.
## 11 member        Wed              551020             841.
## 12 member        Thu              525160             843.
## 13 member        Fri              497134             847.
## 14 member        Sat              322912             964.
```

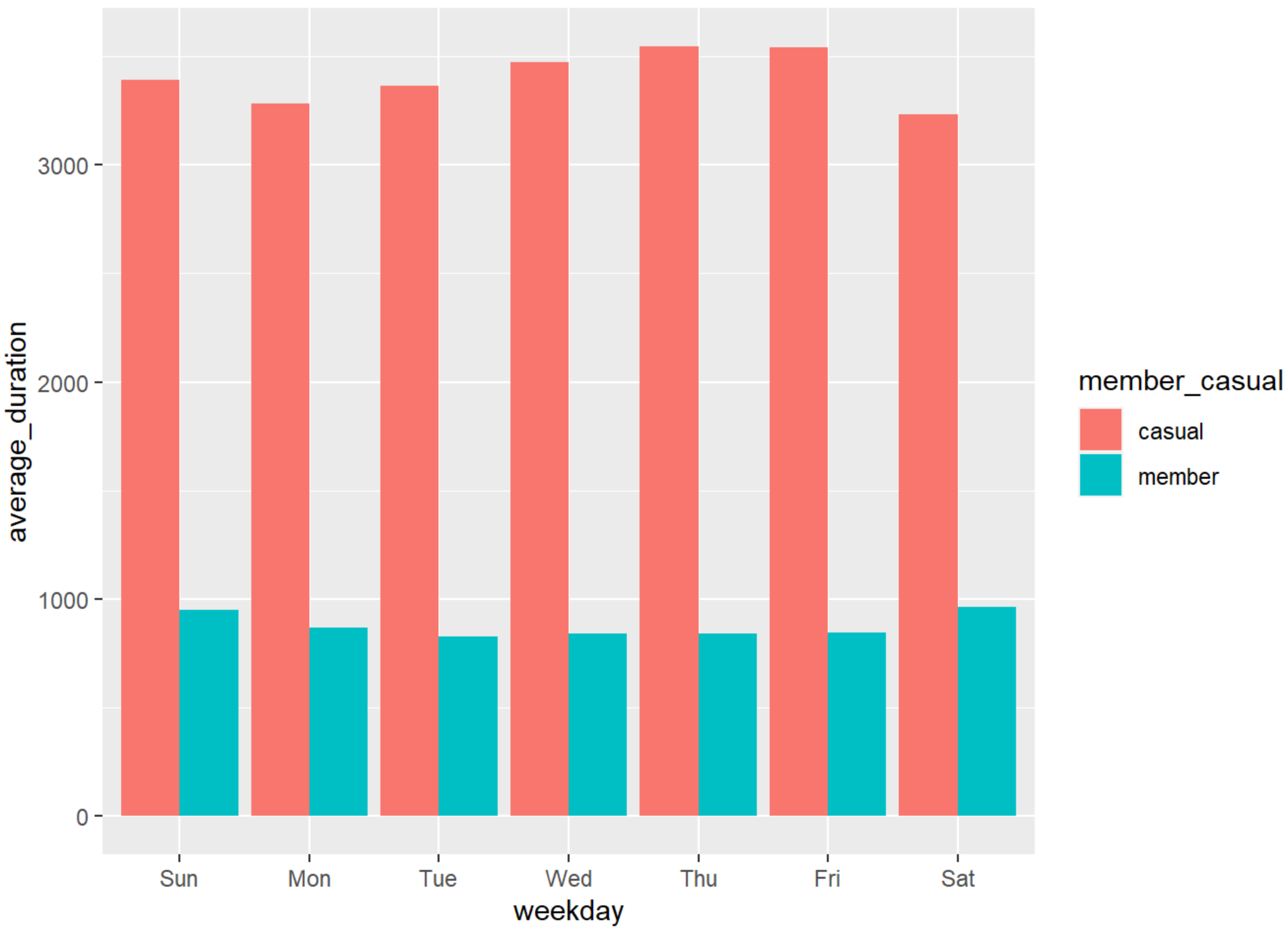
Looking at this result regarding the number of rides and average duration during the days of the week, we can notice that casual users have higher average length of ride during each day of the week than annual members but annual members take significantly higher number of rides during each day of the week than casual users.

Next we will visualize those findings with first, the number of rides by rider type (annual member vs casual riders) and then we will visualize the average duration of rides by rider type (annual member vs casual riders)

```
all_bike_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")
```



```
all_bike_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")
```



Step 5: SHARE

From both graphs and the interesting insights we got throughout the analyze phase, we can observe: 1. From the first graph that, annual members take longer trips during the week which is considerably higher than the number of rides that casual users take during the week, however,casual users take longer rides during the weekend. 2. From the second graph, we notice that casual users have considerably higher average trip duration compared to annual members during the week, which is above 3000 seconds duration for each day of the week.

- To conclude with this analysis, we can say that there is clear difference between casual users and annual members as casual users average higher ride duration duration the week compared to annual members, however, annual members take a considerably higher number of rides each day of week especially during the weekday compared to casual users who are more active during the weekend.