Bike\_trip\_Chicago\_2019

Neeraj

2023-01-12

# Problem Statement

In this case study, I use Divvy trips data set which has data of a bike-sharing company in Chicago during 2019. Analyzing the given data set we aim to design marketing strategies aimed at converting casual riders into annual members.

Data collected from [Divvy\_trips](https://divvy-tripdata.s3.amazonaws.com/index.html)

# Importing necessary libraries

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.0   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.5.0   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(lubridate)

## Loading required package: timechange  
##   
## Attaching package: 'lubridate'  
##   
## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(ggplot2)

# Reading datasets of Bike trips in 2019 over 4 quarters

df1=read\_csv("D:/Capstone 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.

df2=read\_csv("D:/Capstone 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.

df3=read\_csv("D:/Capstone 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.

df4=read\_csv("D:/Capstone 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.

# Making data consistent across all 4 datasets

colnames(df1)

## [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(df2)

## [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(df3)

## [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(df4)

## [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"

df2 <- df2 %>%   
 rename(  
 trip\_id = "01 - Rental Details Rental ID"  
 ,bikeid = "01 - Rental Details Bike ID"   
 ,start\_time = "01 - Rental Details Local Start Time"   
 ,end\_time = "01 - Rental Details Local End Time"   
 ,tripduration= "01 - Rental Details Duration In Seconds Uncapped"  
 ,from\_station\_name = "03 - Rental Start Station Name"   
 ,from\_station\_id = "03 - Rental Start Station ID"  
 ,to\_station\_name = "02 - Rental End Station Name"   
 ,to\_station\_id = "02 - Rental End Station ID"  
 ,usertype = "User Type"  
 ,gender="Member Gender"  
 ,birthyear= "05 - Member Details Member Birthday Year"  
 )  
df1 <- mutate(df1, trip\_id = as.character(trip\_id)  
 ,bikeid = as.character(bikeid))  
df2 <- mutate(df2, trip\_id = as.character(trip\_id)  
 ,bikeid = as.character(bikeid))  
df3 <- mutate(df3, trip\_id = as.character(trip\_id)  
 ,bikeid = as.character(bikeid))  
df4 <- mutate(df4, trip\_id = as.character(trip\_id)  
 ,bikeid = as.character(bikeid))

# Merging the dataset together

all\_trips= bind\_rows(df1,df2,df3,df4)  
colnames(all\_trips)

## [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"

dim(all\_trips)

## [1] 3818004 12

summary(all\_trips)

## trip\_id start\_time   
## Length:3818004 Min. :2019-01-01 00:04:37.00   
## Class :character 1st Qu.:2019-05-29 15:49:26.50   
## Mode :character Median :2019-07-25 17:50:54.00   
## Mean :2019-07-19 21:47:37.11   
## 3rd Qu.:2019-09-15 06:48:05.75   
## Max. :2019-12-31 23:57:17.00   
##   
## end\_time bikeid tripduration   
## Min. :2019-01-01 00:11:07.00 Length:3818004 Min. : 61   
## 1st Qu.:2019-05-29 16:09:28.25 Class :character 1st Qu.: 411   
## Median :2019-07-25 18:12:23.00 Mode :character Median : 709   
## Mean :2019-07-19 22:11:47.56 Mean : 1450   
## 3rd Qu.:2019-09-15 08:30:13.25 3rd Qu.: 1283   
## Max. :2020-01-21 13:54:35.00 Max. :10628400   
##   
## from\_station\_id from\_station\_name to\_station\_id to\_station\_name   
## Min. : 1.0 Length:3818004 Min. : 1.0 Length:3818004   
## 1st Qu.: 77.0 Class :character 1st Qu.: 77.0 Class :character   
## Median :174.0 Mode :character Median :174.0 Mode :character   
## Mean :201.7 Mean :202.6   
## 3rd Qu.:289.0 3rd Qu.:291.0   
## Max. :673.0 Max. :673.0   
##   
## usertype gender birthyear   
## Length:3818004 Length:3818004 Min. :1759   
## Class :character Class :character 1st Qu.:1979   
## Mode :character Mode :character Median :1987   
## Mean :1984   
## 3rd Qu.:1992   
## Max. :2014   
## NA's :538751

# Data cleaning

all\_trips$date <- as.Date(all\_trips$start\_time)  
all\_trips$time <- format(as\_datetime(all\_trips$start\_time),"%H")  
all\_trips$month <- format(as.Date(all\_trips$start\_time),"%b")  
all\_trips$day <- format(as.Date(all\_trips$start\_time),"%d")  
all\_trips$day\_of\_week <- format(as.Date(all\_trips$start\_time),"%A")  
str(all\_trips)

## tibble [3,818,004 × 17] (S3: tbl\_df/tbl/data.frame)  
## $ trip\_id : chr [1:3818004] "21742443" "21742444" "21742445" "21742446" ...  
## $ 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 : chr [1:3818004] "2167" "4386" "1524" "252" ...  
## $ 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 & 18th St" "California Ave & Milwaukee 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 (\*)" "Western Ave & Fillmore St (\*)" "Clark St & Elm St" ...  
## $ 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 ...  
## $ date : Date[1:3818004], format: "2019-01-01" "2019-01-01" ...  
## $ time : chr [1:3818004] "00" "00" "00" "00" ...  
## $ month : chr [1:3818004] "Jan" "Jan" "Jan" "Jan" ...  
## $ day : chr [1:3818004] "01" "01" "01" "01" ...  
## $ day\_of\_week : chr [1:3818004] "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...

#selecting only relevant features

df <- select(all\_trips,-c(start\_time,end\_time,bikeid,from\_station\_id,  
 to\_station\_id,birthyear))

# COMPARE

## trip duration with usertype

aggregate(df$tripduration~df$usertype,FUN = mean)

## df$usertype df$tripduration  
## 1 Customer 3420.8857  
## 2 Subscriber 859.3833

aggregate(df$tripduration~df$usertype,FUN = length)

## df$usertype df$tripduration  
## 1 Customer 880637  
## 2 Subscriber 2937367

## trip duration with gender

aggregate(df$tripduration~df$gender,FUN = mean)

## df$gender df$tripduration  
## 1 Female 1304.1320  
## 2 Male 987.4869

aggregate(df$tripduration~df$gender,FUN = length)

## df$gender df$tripduration  
## 1 Female 857978  
## 2 Male 2400820

## trip duration with day\_week

#fixing the order  
df$day\_of\_week <- ordered(df$day\_of\_week, levels=c("Sunday",  
 "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))  
aggregate(df$tripduration~df$day\_of\_week,FUN = mean)

## df$day\_of\_week df$tripduration  
## 1 Sunday 1900.408  
## 2 Monday 1292.127  
## 3 Tuesday 1241.785  
## 4 Wednesday 1257.296  
## 5 Thursday 1303.871  
## 6 Friday 1415.446  
## 7 Saturday 1929.782

aggregate(df$tripduration~df$day\_of\_week,FUN = length)

## df$day\_of\_week df$tripduration  
## 1 Sunday 426420  
## 2 Monday 560269  
## 3 Tuesday 585680  
## 4 Wednesday 584022  
## 5 Thursday 588287  
## 6 Friday 578107  
## 7 Saturday 495219

## trip duration with month

df$month <- ordered(df$month, levels=c("Jan","Feb", "Mar", "Apr", "May", "Jun",  
 "Jul","Aug","Sep","Oct","Nov","Dec"))  
aggregate(df$tripduration~df$month,FUN = mean)

## df$month df$tripduration  
## 1 Jan 1023.657  
## 2 Feb 1021.613  
## 3 Mar 1008.720  
## 4 Apr 1214.908  
## 5 May 1329.303  
## 6 Jun 1388.453  
## 7 Jul 1805.295  
## 8 Aug 1936.238  
## 9 Sep 1437.189  
## 10 Oct 1308.527  
## 11 Nov 1091.974  
## 12 Dec 1014.410

aggregate(df$tripduration~df$month,FUN = length)

## df$month df$tripduration  
## 1 Jan 103272  
## 2 Feb 96186  
## 3 Mar 165611  
## 4 Apr 265310  
## 5 May 367458  
## 6 Jun 475395  
## 7 Jul 557315  
## 8 Aug 590184  
## 9 Sep 493219  
## 10 Oct 371786  
## 11 Nov 177176  
## 12 Dec 155092

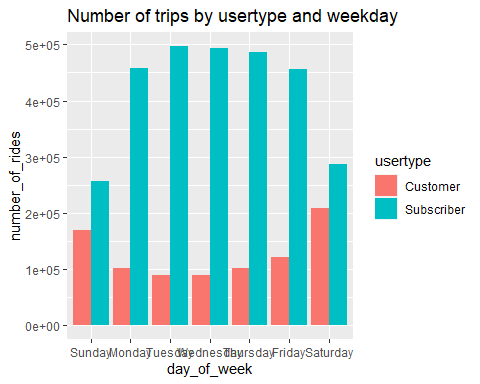
# ANALYSIS

## ridership data by usertype and weekday

graph\_1 <- df %>%  
 group\_by(usertype,day\_of\_week) %>%   
 summarise(number\_of\_rides = n(), avg\_duration=mean(tripduration)) %>%   
 arrange(usertype,day\_of\_week)

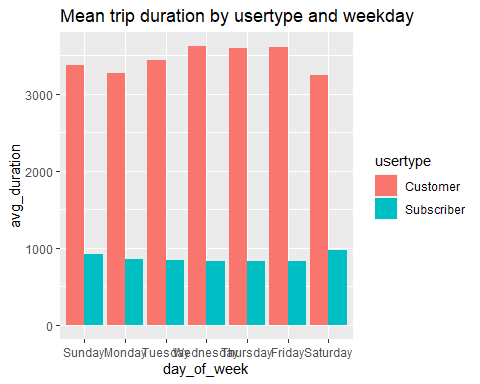
#### Number of trips by usertype and weekday

ggplot(data = graph\_1, aes(x=day\_of\_week,y=number\_of\_rides,fill = usertype)) +  
 geom\_col(position = "dodge")+labs(title = "Number of trips by usertype and weekday")



#### Mean trip duration by usertype and weekday

ggplot(data = graph\_1, aes(x=day\_of\_week,y=avg\_duration,fill = usertype)) +  
 geom\_col(position = "dodge")+labs(title = "Mean trip duration by usertype and weekday")

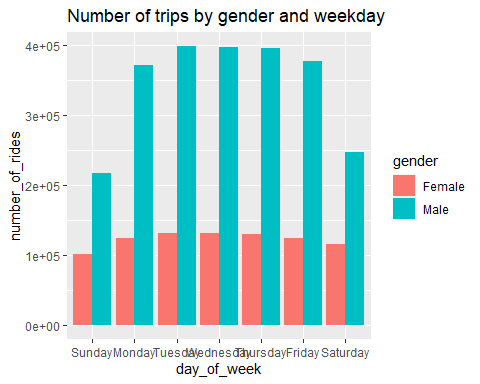


## ridership data by gender and weekday

graph\_2 <- df %>%  
 drop\_na(gender) %>%   
 group\_by(gender,day\_of\_week) %>%   
 summarise(number\_of\_rides = n(), avg\_duration=mean(tripduration)) %>%   
 arrange(gender,day\_of\_week)

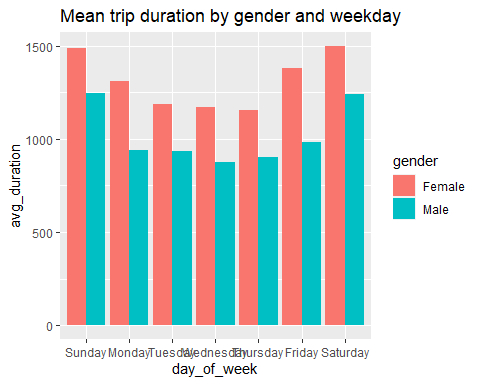
#### Number of trips by gender and weekday

ggplot(data = graph\_2, aes(x=day\_of\_week,y=number\_of\_rides,fill = gender)) +  
 geom\_col(position = "dodge")+labs(title = "Number of trips by gender and weekday")



#### Mean trip duration by gender and weekday

ggplot(data = graph\_2, aes(x=day\_of\_week,y=avg\_duration,fill = gender)) +  
 geom\_col(position = "dodge")+labs(title = "Mean trip duration by gender and weekday")

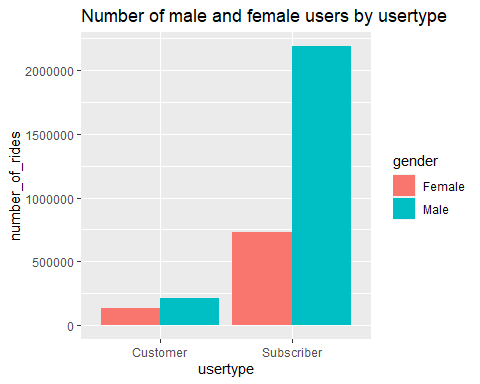


## ridership data by gender and usertype

graph\_3 <- df %>%  
 drop\_na(gender) %>%   
 group\_by(gender,usertype) %>%   
 summarise(number\_of\_rides = n(), avg\_duration=mean(tripduration)) %>%   
 arrange(gender,usertype)

#### Number of male and female users by usertype

ggplot(data = graph\_3, aes(x=usertype,y=number\_of\_rides,fill = gender)) +  
 geom\_col(position = "dodge")+labs(title = "Number of male and female users by usertype")

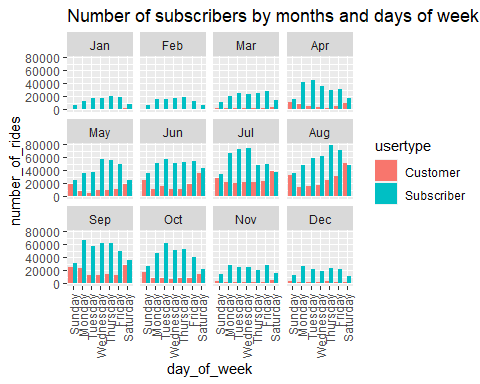


## ridership data by week days and months

df$month <- ordered(df$month, levels=c("Jan","Feb", "Mar", "Apr", "May", "Jun",  
 "Jul","Aug","Sep","Oct","Nov","Dec"))  
df$day\_of\_week <- ordered(df$day\_of\_week, levels=c("Sunday", "Monday", "Tuesday",  
 "Wednesday", "Thursday", "Friday", "Saturday"))  
graph\_4 <- df %>%  
 group\_by(day\_of\_week,month,usertype) %>%   
 summarise(number\_of\_rides = n(), avg\_duration=mean(tripduration)) %>%   
 arrange(day\_of\_week,month)

#### Number of subscribers by months and days of week

ggplot(data = graph\_4, aes(x=day\_of\_week,y=number\_of\_rides, fill = usertype)) +  
 geom\_col(position = "dodge")+  
 facet\_wrap(~month)+  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))+  
 labs(title = "Number of subscribers by months and days of week")

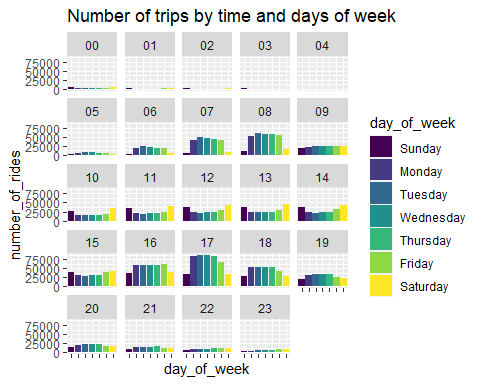


## ridership data by week days and time

df$day\_of\_week <- ordered(df$day\_of\_week, levels=c("Sunday",  
 "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))  
graph\_5 <- df %>%  
 group\_by(day\_of\_week,time) %>%   
 summarise(number\_of\_rides = n(), avg\_duration=mean(tripduration)) %>%   
 arrange(day\_of\_week,time)

#### Number of trips by time and days of week

ggplot(data = graph\_5, aes(x=day\_of\_week,y=number\_of\_rides, fill = day\_of\_week)) +  
 geom\_col(position = "dodge")+  
 facet\_wrap(~time)+  
 theme(axis.text.x = element\_blank())+  
 labs(title = "Number of trips by time and days of week")



# Observations

* [From this plot](#number-of-trips-by-usertype-and-weekday) , we can see that number of rides for subscribers increase through working days and decreases on the weekends. While the customer number is high on weekends.
* [From this plot](#X9560d1658ff59d59b2c89336f7bac4dad703b60) , we can see that ride duration is high for customers than subscribers.
* [From this plot](#number-of-trips-by-gender-and-weekday) , we can see that number of male riders increases through workdays and is at the lowest on weekends. Number of females riders are low compared to male riders and is fairly consistent throughout the week.
* [Here](#mean-trip-duration-by-gender-and-weekday) , Average ride duration is higher for females compared to males and it is high on weekends and decreases through the working days.
* [This](#X711894ab796e5f50274e6f80b9b7f04bb5ef26a) plot shows that there are less number of female subscribers for the company.
* [From this plot](#Xe8df2beda008fb0c891e431ab5774359832953f) , we can see that subscriptions start to increase from the month of April to October and is low on the months from November to March.
* [Here](#number-of-trips-by-time-and-days-of-week) , we can see that the number of riders start to increase from morning 6 am, reaches a peak at 8 am, then decrease mid day and then again reaches peak at 4pm to 6pm, then decreases again.

# Conclusion

* To increase the subscribers on weekends, weekend specific offers and discounts for subscribers taking bikes from/to entertainment centers or recreational activities should be introduced.
* Provide special subscription offers to customers who have ride duration over a certain limit.
* Advertisements and campaigns to increase the number of female users should be introduced.
* Advertisements and discounts should be increased on the months from November to march to increase the number of users during this period.