Google Data Analytics Capstone Project

Jason Christian Wijaya

2023-01-11

Case Study Scenario

In 2016, 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 bikes can be unlocked from one station and returned to any other station in the system anytime. Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes,and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.

Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, Moreno believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, Moreno believes there is a very good chance to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs. Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the marketing analyst team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

Goal

Better understand how annual members and casual riders differ

Data Source

Data license : https://ride.divvybikes.com/data-license-agreement

Data source : https://divvy-tripdata.s3.amazonaws.com/index.html

The data is organized in .csv formats and are grouped by month, and consists of x columns, which are :
"ride_id","rideable_type","started_at","ended_at","start_station_name","start_station_id","end_station_name","end_s

Notes : 12 months of recent data were used from this source (November 2021 - October 2022)

Data Preparation

Load the necessary libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(readr)
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.2.2
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(tidyr)
library(lubridate)
```

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

```
library(ggplot2)
```

Load the file path

Loading the directory to the datasets that have been downloaded and checking all 12 files.

```
folder <- "/Users/Jason/Downloads/Data Analyst/Capstone/Cyclistic Data/Csv files/"      # path to folde
file_list <- list.files(path=folder, pattern="*.csv") # create list of all .csv files in folder

print(file_list)
```

```
##  [1] "202110-divvy-tripdata.csv"       "202111-divvy-tripdata.csv"
##  [3] "202112-divvy-tripdata.csv"       "202201-divvy-tripdata.csv"
##  [5] "202202-divvy-tripdata.csv"       "202203-divvy-tripdata.csv"
##  [7] "202204-divvy-tripdata.csv"       "202205-divvy-tripdata.csv"
##  [9] "202206-divvy-tripdata.csv"       "202207-divvy-tripdata.csv"
## [11] "202208-divvy-tripdata.csv"       "202209-divvy-publictripdata.csv"
```

Combining all 12 files into a single dataframe

```
data <-
  do.call("rbind",
          lapply(file_list,
                 function(x)
                   read.csv(paste(folder, x, sep=''),
                            stringsAsFactors = FALSE)))
```

Data Cleaning

Cleaning the column names

Cleaning the column names with clean_names() function from janitor library. This helps us make sure that all column names follow the same syntax.

```
data<-clean_names(data)
colnames(data)
```

```
##  [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"
```

Changing ambiguous column names

I changed 'member_casual' column to be 'membership_type' to make it easier to understand.

```
colnames(data)[colnames(data) == "member_casual"] = "membership_type"
glimpse(data)
```

```
## Rows: 5,828,235
## Columns: 13
## $ ride_id            <chr> "620BC6107255BF4C", "4471C70731AB2E45", "26CA69D43D~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <chr> "2021-10-22 12:46:42", "2021-10-21 09:12:37", "2021~
## $ ended_at           <chr> "2021-10-22 12:49:50", "2021-10-21 09:14:14", "2021~
## $ start_station_name <chr> "Kingsbury St & Kinzie St", "", "", "", "", "", "",~
## $ start_station_id   <chr> "KA1503000043", "", "", "", "", "", "", "", "", "",~
## $ end_station_name   <chr> "", "", "", "", "", "", "", "", "", "", "", "", "",~
## $ end_station_id     <chr> "", "", "", "", "", "", "", "", "", "", "", "", "",~
## $ start_lat          <dbl> 41.88919, 41.93000, 41.92000, 41.92000, 41.89000, 4~
## $ start_lng          <dbl> -87.63850, -87.70000, -87.70000, -87.69000, -87.710~
## $ end_lat            <dbl> 41.89000, 41.93000, 41.94000, 41.92000, 41.89000, 4~
## $ end_lng            <dbl> -87.63000, -87.71000, -87.72000, -87.69000, -87.690~
## $ membership_type    <chr> "member", "member", "member", "member", "member", "~
```

Removing duplicate datas

```
data<-distinct(data)
```

Checking for cells with 'NA' values

```
colSums(is.na(data))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0               5844               5844
##    membership_type
##                  0
```

Checking for cells with NULL values

```
colSums(data=="")
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##             895032             895032             958227             958227
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0                 NA                 NA
##    membership_type
##                  0
```

Filling NULL values with NA and deleting rows that contains NA values

```
data[data==""] <- NA
df <- na.omit(data)
```

Converting every data in each columns to follow their respective data types.

```
df <- type.convert(df, as.is = TRUE)
sapply(df, class)
```

```
##            ride_id      rideable_type         started_at           ended_at
##        "character"        "character"        "character"        "character"
## start_station_name   start_station_id   end_station_name     end_station_id
##        "character"        "character"        "character"        "character"
```

```
##          start_lat          start_lng            end_lat            end_lng
##          "numeric"          "numeric"          "numeric"          "numeric"
##    membership_type
##        "character"
```

Checking the columns and its datatype

```
glimpse(df)
```

```
## Rows: 4,474,141
## Columns: 13
## $ ride_id            <chr> "614B15BC42810184", "ADCC6E3CF9C04688", "6184CC5724~
## $ rideable_type      <chr> "docked_bike", "classic_bike", "docked_bike", "dock~
## $ started_at         <chr> "2021-10-05 10:56:05", "2021-10-06 13:55:33", "2021~
## $ ended_at           <chr> "2021-10-05 11:38:48", "2021-10-06 13:58:16", "2021~
## $ start_station_name <chr> "Michigan Ave & Oak St", "Desplaines St & Kinzie St~
## $ start_station_id   <chr> "13042", "TA1306000003", "13042", "13042", "KA15030~
## $ end_station_name   <chr> "Michigan Ave & Oak St", "Kingsbury St & Kinzie St"~
## $ end_station_id     <chr> "13042", "KA1503000043", "13042", "13042", "TA13060~
## $ start_lat          <dbl> 41.90096, 41.88872, 41.90096, 41.90096, 41.88918, 4~
## $ start_lng          <dbl> -87.62378, -87.64445, -87.62378, -87.62378, -87.638~
## $ end_lat            <dbl> 41.90096, 41.88918, 41.90096, 41.90096, 41.88872, 4~
## $ end_lng            <dbl> -87.62378, -87.63851, -87.62378, -87.62378, -87.644~
## $ membership_type    <chr> "casual", "member", "casual", "casual", "member", "~
```

Formatting date&time column to follow date&time datatypes, and adding day and month columns

```
df2 = df %>%
  mutate(
    started_at = ymd_hms(as_datetime(started_at)),
    ended_at = ymd_hms(as_datetime(ended_at))
  )

df2 = df2 %>%
  mutate(
    day = wday(started_at, label = T, abbr = F),
    month = month(started_at, label = T, abbr =F),
  )
glimpse(df2)
```

```
## Rows: 4,474,141
## Columns: 15
## $ ride_id            <chr> "614B15BC42810184", "ADCC6E3CF9C04688", "6184CC5724~
## $ rideable_type      <chr> "docked_bike", "classic_bike", "docked_bike", "dock~
## $ started_at         <dttm> 2021-10-05 10:56:05, 2021-10-06 13:55:33, 2021-10-~
## $ ended_at           <dttm> 2021-10-05 11:38:48, 2021-10-06 13:58:16, 2021-10-~
## $ start_station_name <chr> "Michigan Ave & Oak St", "Desplaines St & Kinzie St~
## $ start_station_id   <chr> "13042", "TA1306000003", "13042", "13042", "KA15030~
```

```
## $ end_station_name   <chr> "Michigan Ave & Oak St", "Kingsbury St & Kinzie St"~
## $ end_station_id     <chr> "13042", "KA1503000043", "13042", "13042", "TA13060~
## $ start_lat          <dbl> 41.90096, 41.88872, 41.90096, 41.90096, 41.88918, 4~
## $ start_lng          <dbl> -87.62378, -87.64445, -87.62378, -87.62378, -87.638~
## $ end_lat            <dbl> 41.90096, 41.88918, 41.90096, 41.90096, 41.88872, 4~
## $ end_lng            <dbl> -87.62378, -87.63851, -87.62378, -87.62378, -87.644~
## $ membership_type    <chr> "casual", "member", "casual", "casual", "member", "~
## $ day                <ord> Tuesday, Wednesday, Saturday, Sunday, Saturday, Mon~
## $ month              <ord> October, October, October, October, October, Octobe~
```

Checking each unique values of the bike type column

```
unique(df2$rideable_type)
```

```
## [1] "docked_bike"   "classic_bike"  "electric_bike"
```

Checking each unique values of the membership type column

```
unique(df2$membership_type)
```

```
## [1] "casual" "member"
```

Data Analysis

Dataframe Summary

```
summary(df2)
```

```
##    ride_id          rideable_type        started_at
##  Length:4474141     Length:4474141     Min.   :2021-10-01 00:00:09.00
##  Class :character   Class :character   1st Qu.:2022-03-05 17:30:24.00
##  Mode  :character   Mode  :character   Median :2022-06-09 21:24:53.00
##                                        Mean   :2022-05-08 21:27:11.38
##                                        3rd Qu.:2022-08-02 08:44:21.00
##                                        Max.   :2022-09-30 23:59:56.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2021-10-01 00:03:51.00   Length:4474141     Length:4474141
##  1st Qu.:2022-03-05 17:57:15.00   Class :character   Class :character
##  Median :2022-06-09 21:42:17.00   Mode  :character   Mode  :character
##  Mean   :2022-05-08 21:44:41.74
##  3rd Qu.:2022-08-02 08:57:25.00
##  Max.   :2022-10-01 14:22:35.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:4474141     Length:4474141     Min.   :41.65   Min.   :-87.83
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
```

```
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.64
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :45.64   Max.   :-73.80
##
##     end_lat         end_lng       membership_type           day
##  Min.   :41.65   Min.   :-87.83   Length:4474141     Sunday   :616483
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   Monday   :583331
##  Median :41.90   Median :-87.64   Mode  :character   Tuesday  :631349
##  Mean   :41.90   Mean   :-87.64                      Wednesday:629556
##  3rd Qu.:41.93   3rd Qu.:-87.63                      Thursday :637192
##  Max.   :42.06   Max.   :-87.53                      Friday   :637055
##                                                      Saturday :739175
##        month
##  July     : 642680
##  June     : 620350
##  August   : 605325
##  September: 535145
##  May      : 502545
##  October  : 477972
##  (Other)  :1090124
```

Splitting the dataframes for easier analysis

```
dataframe <- df2[, c('rideable_type','membership_type','day','month')]
stationdata<- df2[, c('ride_id','start_station_name','end_station_name')]
```

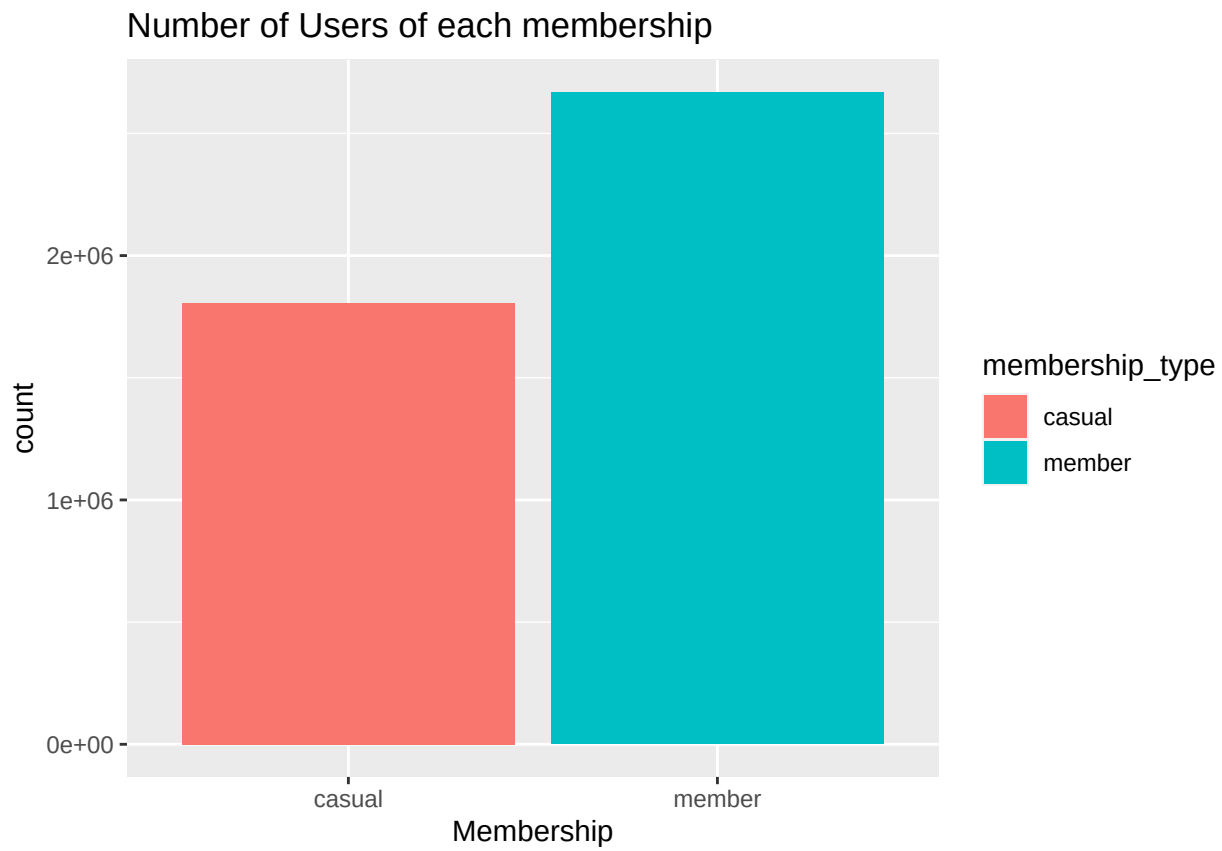
1st dataframe

```
str(dataframe)
```

```
## 'data.frame':    4474141 obs. of  4 variables:
##  $ rideable_type  : chr  "docked_bike" "classic_bike" "docked_bike" "docked_bike" ...
##  $ membership_type: chr  "casual" "member" "casual" "casual" ...
##  $ day            : Ord.factor w/ 7 levels "Sunday"<"Monday"<..: 3 4 7 1 7 2 6 5 6 1 ...
##  $ month          : Ord.factor w/ 12 levels "January"<"February"<..: 10 10 10 10 10 10 10 10 10 10 .
```

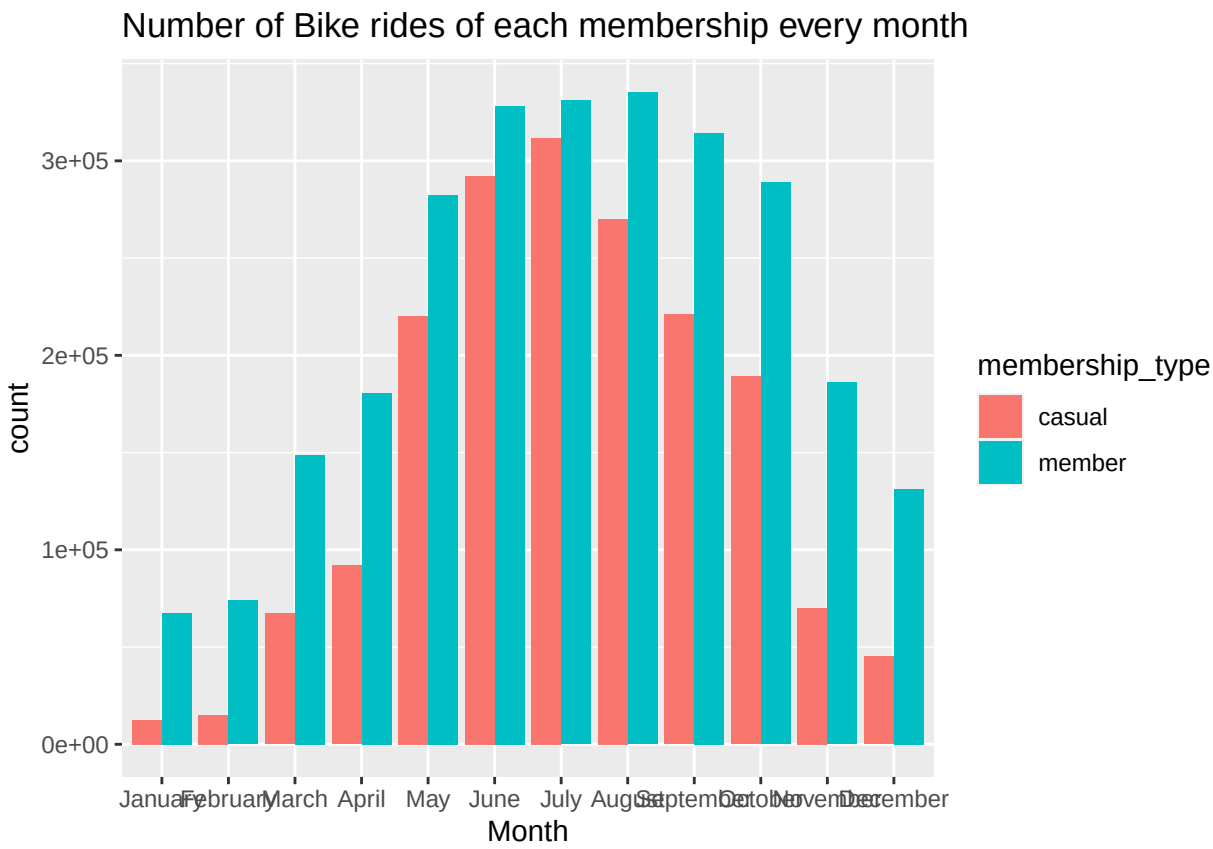
Visualization of Number of Users for each membership

```
plot1<-ggplot(dataframe,aes(x = membership_type,fill=membership_type))+
  geom_bar()+
  labs(
    title = "Number of Users of each membership",
    x = "Membership")
plot1
```



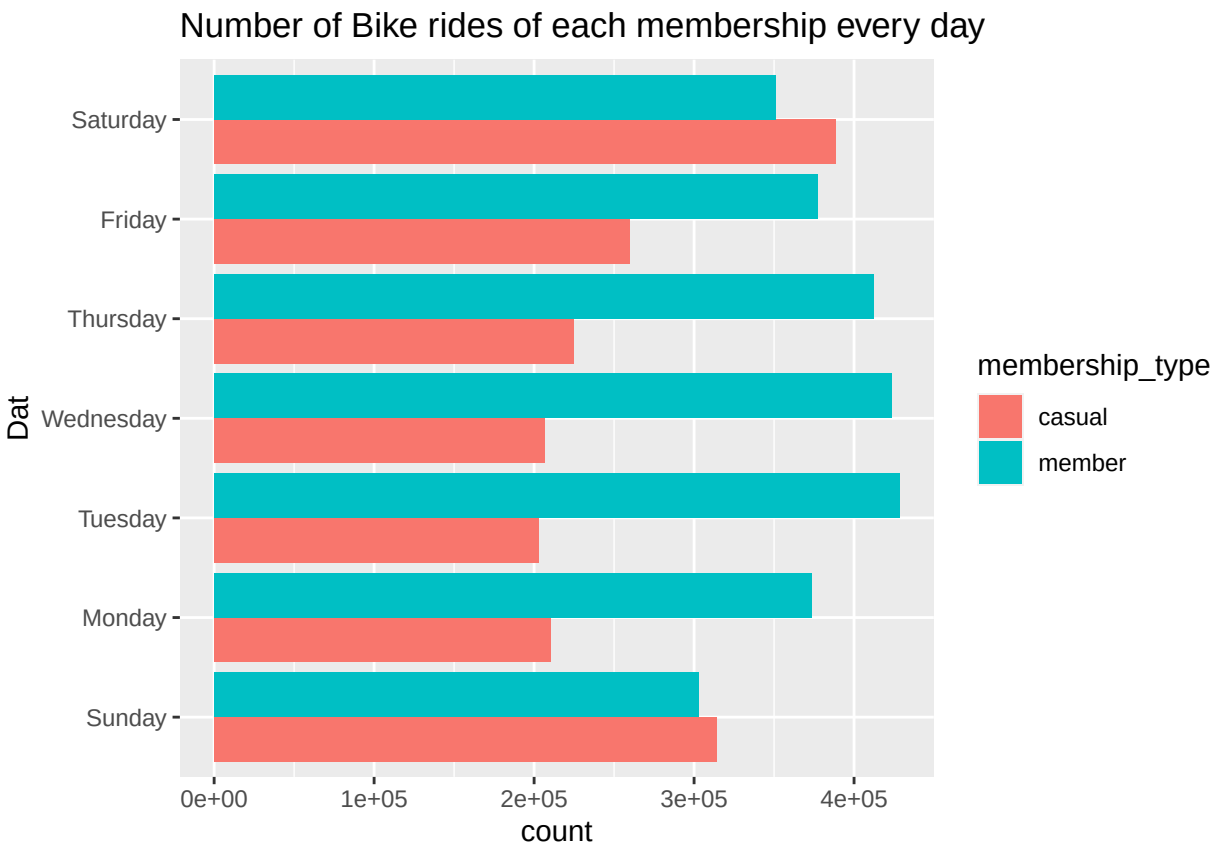
Visualization of Number of Bike rides of each membership by month

```
plot2 <- ggplot(dataframe, aes(x = month, fill = membership_type)) +
  geom_bar(position = "dodge")+
  labs(
    title = "Number of Bike rides of each membership every month",
    x = "Month")
plot2
```



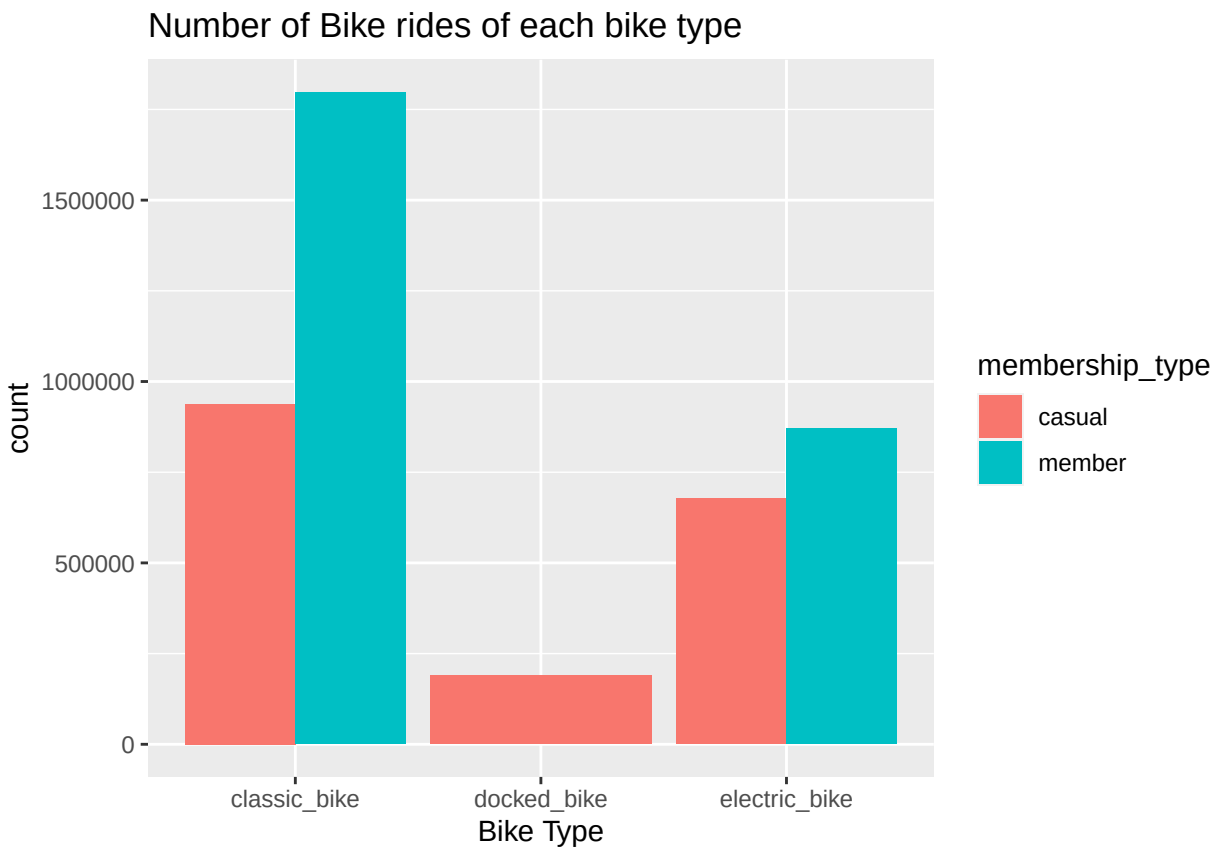
Visualization of Number of Bike rides of each membership by day

```
plot3 <- ggplot(dataframe, aes(x = day, fill = membership_type)) +
  geom_bar(position = "dodge")+coord_flip()+
  labs(
    title = "Number of Bike rides of each membership every day",
    x = "Dat")
plot3
```



Visualization of Number of Bike rides of each bike type

```
plot4 <- ggplot(dataframe, aes(x = rideable_type, fill = membership_type)) +
    geom_bar(position = "dodge")+
  labs(
    title = "Number of Bike rides of each bike type",
    x = "Bike Type")
plot4
```



Counting each unique starting station and display the 10 most crowded station

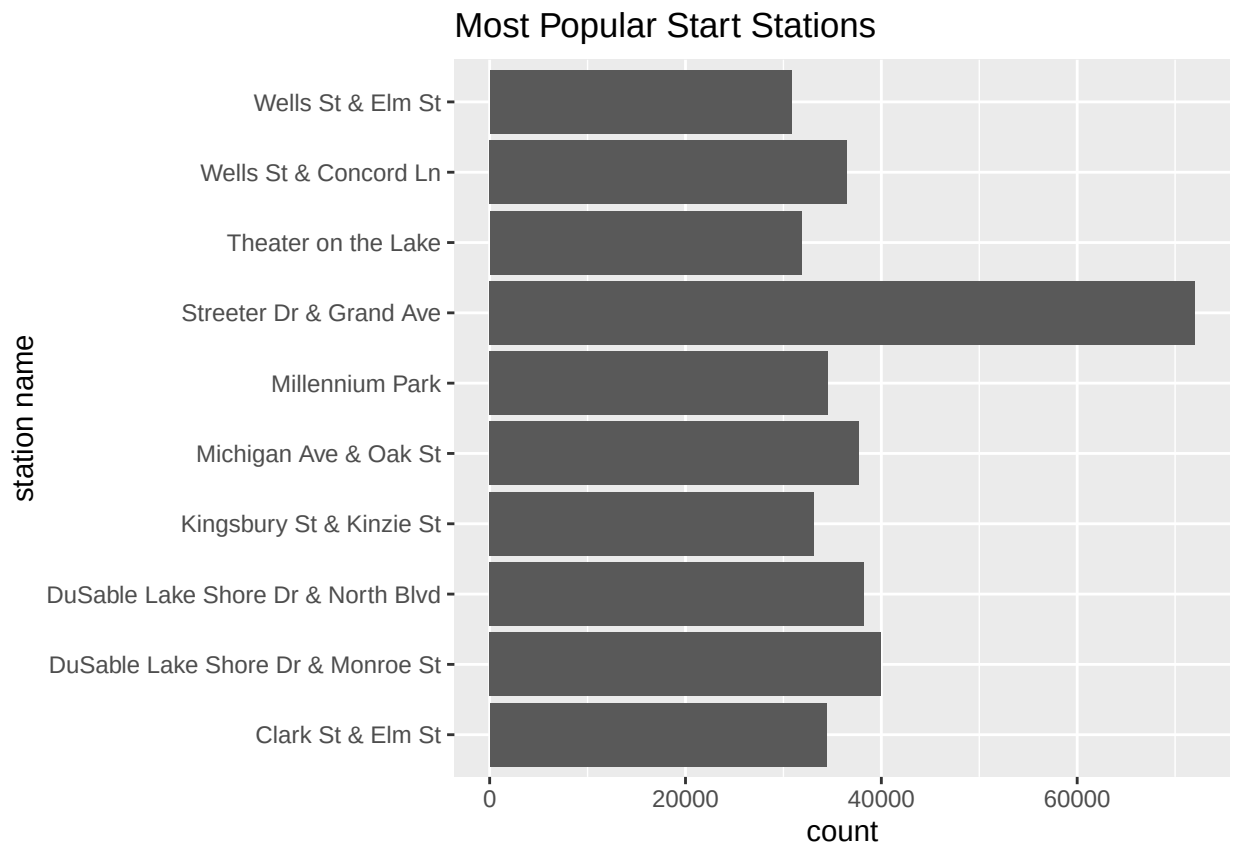
```
startcount<-stationdata %>%
  group_by(start_station_name)%>% summarize(count = n_distinct(ride_id))

startcount<- startcount[order(startcount$count,decreasing=TRUE),]
startcount<-head(startcount,10)
head(startcount)
```

```
## # A tibble: 6 x 2
##   start_station_name                count
##   <chr>                             <int>
## 1 Streeter Dr & Grand Ave           72044
## 2 DuSable Lake Shore Dr & Monroe St 39951
## 3 DuSable Lake Shore Dr & North Blvd 38236
## 4 Michigan Ave & Oak St             37716
## 5 Wells St & Concord Ln             36489
## 6 Millennium Park                   34554
```

Visualization of the 10 most popular start stations

```
plot5 <- ggplot(startcount, aes(x=start_station_name, y=count)) +
  geom_bar(stat="identity")+ coord_flip()+
  labs(
    title = "Most Popular Start Stations",
    x = "station name"
  )
plot5
```



Counting each unique end station and display the 10 most crowded station

```
endcount<-stationdata %>%
  group_by(end_station_name)%>% summarize(count = n_distinct(ride_id))

endcount<- endcount[order(endcount$count,decreasing=TRUE),]
endcount<-head(endcount,10)
head(endcount)
```

```
## # A tibble: 6 x 2
##   end_station_name                  count
##   <chr>                             <int>
## 1 Streeter Dr & Grand Ave           73731
## 2 DuSable Lake Shore Dr & North Blvd 41076
```

```
## 3 DuSable Lake Shore Dr & Monroe St  39017
## 4 Michigan Ave & Oak St              38801
## 5 Wells St & Concord Ln              36519
## 6 Millennium Park                    35549
```

Visualization of the 10 most popular end stations

```
plot6 <- ggplot(endcount, aes(x=end_station_name, y=count)) +
  geom_bar(stat="identity")+ coord_flip()+
  labs(
    title = "Most Popular end Stations",
    x = "station name"
  )
plot6
```