CASE STUDY: HOW DOES A BIKE-SHARE NAVIGATE SPEEDY SUCCESS?
GOOGLE DATA ANALYTICS CASE STUDY

Table of Contents

Chapter I. Introduction

1.1. Background

This case study is one of four capstone assignments for the Google Data Analytics Professional Certificate program. The participants will perform data analysis for a fictional bike-share company to help them attract more riders by using all the knowledge they have learned through the course which focuses on analytical skills (data cleaning, analysis, and visualization) and tools (Excel, SQL, R Programming, Tableau).

This project uses 12 months of Cyclistic's historical trip public data sets between 2021-12 and 2022-11.

1.2. Scenario

Cyclistic is a successful company in the bike-share offering. Founded in 2016 and has grown to a fleet of 5,824 bicycles that are tracked 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 at any time.

There are 3 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.

Working as a junior data analyst in the marketing analyst team at Cyclists, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. The assignment is to design a new marketing strategy to convert casual riders into annual members and back it up with compelling data insights and professional data visualizations.

Chapter II. Ask & Prepare Phase

2.1. Ask

How do annual members and casual riders use Cyclistic bikes differently?

We must learn the key differences between casual riders and annual members to understand the customer, who is the key to our success in the marketing campaign.

2.2. Prepare

The data for the project is provided by Motivate International Inc. and is updated monthly. Since this project was started in December of 2022, the data from 2021 December to 2022 November has been selected for analysis.

Exploring the data set, there are 13 columns which contain multiple data about the customer trips:

1. **ride_id (string)**: Unique number assigned to a riding trip.
2. **rideable_type (string)**: Type of bike used during a trip; standard two-wheel bike, reclining bike, hand tricycle, or cargo bike.
3. **started_at (datetime)**: Start date and time for the trip
4. **ended_at (datetime)**: End date and time for the trip
5. **start_station_name (string)**: Name of the station where the trip started
6. **start_station_id (string)**: Unique identification code assigned to the start station.
7. **end_station_name (string)**: Name of the station where the trip ended.
8. **end_station_id (string)**: Unique identification code assigned to the end station.
9. **start_lat (numeric)**: Latitude coordinate of where the trip started.
10. **start_lng (numeric)**: Longitude coordinate of where the trip started.
11. **end_lat (numeric)**: Latitude coordinate of where the trip ended.
12. **end_lng (numeric)**: Longitude coordinate of where the trip ended.
13. **member_casual (categorical)**: Customer type; “member” = annual member, “casual” = casual rider.

Chapter III. Process Phase

3.1. Process

I used Rstudio Desktop to process, clean, analyse and visualize the data in this project. Because the data of each file is quite large, import and merge cannot be done on RStudio Cloud.

This step aims to import the data into RStudio and merge it all into the same data frame.

- Load the necessary libraries:

```
install.packages("dplyr")
install.packages("ggplot2")
install.packages("tidyverse")
install.packages("readxl")
install.packages("skimr")
library(dplyr)
library(ggplot2)
library(tidyverse)
library(xlsx)
library(data.table)
library(lubridate)
library(skimr)
```

- Import data into Rstudio:

```
file.path <- "C:/Users/minhh/Documents/RStudio File/Data/12.2021_to_11.2022"
file.list <- list.files(path=file.path, pattern = '*.xlsx')
df.list <- lapply(file.list,read_excel)
```

- Combine all the files in one single data frame:

```
tripdata.df <- rbindlist(df.list, fill = TRUE )

glimpse(tripdata.df) (Use glimpse to have an overview of the data frame)
```

```
Columns: 13
$ ride_id            <chr> "46F8167220E4431F", "73A77762838B32FD", "4CF42452054F59C5", "3278BA87BF698339"…
$ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", "classic_bike", "electric_b…
$ started_at         <dttm> 2021-12-07 15:06:07, 2021-12-11 03:43:29, 2021-12-15 23:10:28, 2021-12-26 16:…
$ ended_at           <dttm> 2021-12-07 15:13:42, 2021-12-11 04:10:23, 2021-12-15 23:23:14, 2021-12-26 16:…
$ start_station_name <chr> "Laflin St & Cullerton St", "LaSalle Dr & Huron St", "Halsted St & North Branc…
$ start_station_id   <chr> "13307", "KP1705001026", "KA1504000117", "KA1504000117", "18058", "SL-012", "1…
$ end_station_name   <chr> "Morgan St & Polk St", "Clarendon Ave & Leland Ave", "Broadway & Barry Ave", "…
$ end_station_id     <chr> "TA1307000130", "TA1307000119", "13137", "KP1705001026", "TA1307000142", "SL-0…
$ start_lat          <chr> "41854833", "4.18944051666666E+16", "4189935716666660", "4189939028549690", "4…
$ start_lng          <chr> "-8766366033333330", "-87632331", "-8764852183333330", "-8764854490756980", "-…
$ end_lat            <chr> "418719685", "41967968", "4193758231600620", "41894877", "41931248", "41872596…
$ end_lng            <chr> "-8765096533333330", "-87650001", "-876440978050232", "-87632326", "-87644336"…
$ member_casual      <chr> "member", "casual", "member", "member", "member", "member", "member", "casual"…
```

3.2. Clean

- Transforming data type:

We want start_lat, start_lng, end_lat, and end_lng to be numeric data types.

```
trips_data_type <- tripdata.df %>%
  mutate(
    start_lat = as.numeric(start_lat),
    start_lng = as.numeric(start_lng),
    end_lat = as.numeric(end_lat),
    end_lng = as.numeric(end_lng)
  )
glimpse(trips_data_type)
```

```
Columns: 13
$ ride_id            <chr> "46F8167220E4431F", "73A77762838B32FD", "4CF42452054F59C5", "3278BA87BF698339"…
$ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", "classic_bike", "electric_b…
$ started_at         <dttm> 2021-12-07 15:06:07, 2021-12-11 03:43:29, 2021-12-15 23:10:28, 2021-12-26 16:…
$ ended_at           <dttm> 2021-12-07 15:13:42, 2021-12-11 04:10:23, 2021-12-15 23:23:14, 2021-12-26 16:…
$ start_station_name <chr> "Laflin St & Cullerton St", "LaSalle Dr & Huron St", "Halsted St & North Branc…
$ start_station_id   <chr> "13307", "KP1705001026", "KA1504000117", "KA1504000117", "18058", "SL-012", "1…
$ end_station_name   <chr> "Morgan St & Polk St", "Clarendon Ave & Leland Ave", "Broadway & Barry Ave", "…
$ end_station_id     <chr> "TA1307000130", "TA1307000119", "13137", "KP1705001026", "TA1307000142", "SL-0…
$ start_lat          <dbl> 4.185483e+07, 4.189441e+16, 4.189936e+15, 4.189939e+15, 4.189558e+15, 4.186038…
$ start_lng          <dbl> -8.766366e+15, -8.763233e+07, -8.764852e+15, -8.764854e+15, -8.768202e+15, -8.…
$ end_lat            <dbl> 4.187197e+08, 4.196797e+07, 4.193758e+15, 4.189488e+07, 4.193125e+07, 4.187260…
$ end_lng            <dbl> -8.765097e+15, -8.765000e+07, -8.764410e+14, -8.763233e+07, -8.764434e+07, -8.…
$ member_casual      <chr> "member", "casual", "member", "member", "member", "member", "member", "casual"…
```

- Using summary for statistical summary:

```
summary(trips_data_type)
```

```
 ride_id           rideable_type        started_at                        ended_at                     
 Length:5733451     Length:5733451     Min.   :2021-12-01 00:00:01.00   Min.   :2021-12-01 00:02:40.00  
 Class :character   Class :character   1st Qu.:2022-05-17 12:04:44.50   1st Qu.:2022-05-17 12:27:04.00  
 Mode  :character   Mode  :character   Median :2022-07-13 22:04:44.00   Median :2022-07-13 22:22:06.00  
                                       Mean   :2022-07-06 05:55:33.92   Mean   :2022-07-06 06:14:59.07  
                                       3rd Qu.:2022-09-07 17:55:40.00   3rd Qu.:2022-09-07 18:11:41.00  
                                       Max.   :2022-11-30 23:56:11.00   Max.   :2022-12-01 11:45:53.00  

 start_station_name start_station_id   end_station_name   end_station_id       start_lat            
 Length:5733451     Length:5733451     Length:5733451     Length:5733451     Min.   :               42  
 Class :character   Class :character   Class :character   Class :character   1st Qu.:         41860384  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :         41940775  
                                                                             Mean   : 4727354018490000  
                                                                             3rd Qu.:  419182955629000  
                                                                             Max.   :42064825166700000  
                                                                             NA's   :3                  
   start_lng                   end_lat                     end_lng                 member_casual     
 Min.   :-8783325416670000   Min.   :                 0   Min.   :-8777453933330000   Length:5733451    
 1st Qu.: -876440978050000   1st Qu.:           4196909   1st Qu.:      -8763443007   Class :character  
 Median :         -87667252  Median :          41902973   Median :        -87640795   Mode  :character  
 Mean   :-2157997579650000   Mean   : 2158914514820000   Mean   :-1031417649110000                     
 3rd Qu.:         -87611894  3rd Qu.:        4190096039   3rd Qu.:          -8765103                     
 Max.   :               -88  Max.   :42064755166700000   Max.   :                 0                     
 NA's   :3                   NA's   :5877                NA's   :5877                                  
```

- Fixing incorrect format:

The start_lat, start_lng, end_lat, end_lng should be in decimal degrees (DD) such as 41.40338, -2.17403 for a better analysis. However, in our data frame, latitude and longitude are written as degrees, minutes, and seconds (DMS) in some cases, which is why removing the (dot) and taking only 4 values from the left is necessary. In start_lng and end_lng we must take 5 values including "-" because it is negative. Besides, we need to add the (dot) after the first 2 numeric characters.

I create 2 functions for switching the initial values to the value we need. First, we use **gsub()** to eliminate any character which is not a number(0-9) and the "-". Then use **substr()** to take only 4 or 5 characters we needed for each category. Then use **sub()** to add the (dot) after a specific character as noted before. Finally, use **as.numeric()** to change the data type to numeric.

```
lat_fix <- function(x){
  as.numeric(sub("(.{2})(.*)","\\1.\\2",substr(gsub("[^0-9-]","",x),1,4)))
}
lat_fix <- function(x){
  as.numeric(sub("(.{3})(.*)","\\1.\\2",substr(gsub("[^0-9-]","",x),1,5)))
}
trips_fix1 <- tripdata.df %>%
  mutate(
    start_lat = lat_fix(start_lat),
    start_lng = long_fix(start_lng),
```

```
  end_lat = lat_fix(end_lat),
  end_lng = long_fix(end_lng)
 )
summary(trips_fix1)
```

```
ride_id           rideable_type        started_at                          ended_at
 Length:5733451     Length:5733451     Min.   :2021-12-01 00:00:01.00   Min.   :2021-12-01 00:02:40.00
 Class :character   Class :character   1st Qu.:2022-05-17 12:04:44.50   1st Qu.:2022-05-17 12:27:04.00
 Mode  :character   Mode  :character   Median :2022-07-13 22:04:44.00   Median :2022-07-13 22:22:06.00
                                       Mean   :2022-07-06 05:55:33.92   Mean   :2022-07-06 06:14:59.07
                                       3rd Qu.:2022-09-07 17:55:40.00   3rd Qu.:2022-09-07 18:11:41.00
                                       Max.   :2022-11-30 23:56:11.00   Max.   :2022-12-01 11:45:53.00

 start_station_name start_station_id   end_station_name   end_station_id       start_lat
 Length:5733451     Length:5733451     Length:5733451     Length:5733451     Min.   :41.64
 Class :character   Class :character   Class :character   Class :character   1st Qu.:41.88
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :41.90
                                                                             Mean   :41.90
                                                                             3rd Qu.:41.93
                                                                             Max.   :45.63
                                                                             NA's   :3
   start_lng         end_lat          end_lng        member_casual
 Min.   :-87.84   Min.   : 0.00   Min.   :-88.14   Length:5733451
 1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66   Class :character
 Median :-87.64   Median :41.90   Median :-87.64   Mode  :character
 Mean   :-87.64   Mean   :41.90   Mean   :-87.64
 3rd Qu.:-87.62   3rd Qu.:41.93   3rd Qu.:-87.62
 Max.   :-73.79   Max.   :42.37   Max.   :  0.00
 NA's   :3        NA's   :5877    NA's   :5877
```

Overall, the data frame is good for exploring and analysing. There are still some NAs but we will explore them later.

3.3. Explore

- Count the number of NAs values in the data frame:

```
colSums(is.na(tripdata.df))
```

```
           ride_id      rideable_type         started_at           ended_at start_station_name
                 0                  0                  0                  0             854844
  start_station_id   end_station_name     end_station_id          start_lat          start_lng
            854847             915084            1234694                  3                  3
           end_lat            end_lng      member_casual
              5877               5877                  3
```

We can see that there are many NAs values, which are primarily from the column about station_id and station_name.

start_station_name has **854844** missing values, which is **14.91%**.
start_station_id has **854847** missing values, which is **14.91%**.
end_station_name has **915084** missing values, which is **15.96%**.
end_station_id has **1234694** missing values, which is **21.54%**.
The missing values on these columns are quite large and should be considered for addition.
Although starting and ending locations play an important role, relying solely on that and ignoring the rest such as time and member_type will make the analysis inaccurate. So I decided to keep the data.

- Check if the value in the column is corrected:

Because rideable_type is divided into 3 categories such as electric_bike, classic_bike, and docked_bike. So we have to check if there are any other type in the data frame.

```
unique(trips_fix1$rideable_type)
```

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

The same goes with member_casual, there are NA values, but as discussed above, it is not removed from the data frame.

```
unique(trips_fix1$member_casual)
```

```
[1] "member" "casual" NA
```

- Add new columns for analysis and visualization:

I add 4 columns, which are hour_start, week_day, month, and time_travel. Then create a new data frame **trip_fix2** including all the columns.

```
trip_fix2 <- trips_fix1 %>%
  mutate(
    hour_start = format(as.POSIXct(started_at),"%H"),
    week_day = wday(started_at, week_start = 1, label = TRUE),
    month = month(started_at, label = TRUE),
    time_travel = round(difftime(ended_at,started_at, units = "mins"),0)
  )
summary(trip_fix2)
```

```
   ride_id          rideable_type        started_at                          ended_at
 Length:5733451     Length:5733451     Min.   :2021-12-01 00:00:01.00   Min.   :2021-12-01 00:02:40.00
 Class :character   Class :character   1st Qu.:2022-05-17 12:04:44.50   1st Qu.:2022-05-17 12:27:04.00
 Mode  :character   Mode  :character   Median :2022-07-13 22:04:44.00   Median :2022-07-13 22:22:06.00
                                       Mean   :2022-07-06 05:55:33.92   Mean   :2022-07-06 06:14:59.07
                                       3rd Qu.:2022-09-07 17:55:40.00   3rd Qu.:2022-09-07 18:11:41.00
                                       Max.   :2022-11-30 23:56:11.00   Max.   :2022-12-01 11:45:53.00


 start_station_name start_station_id   end_station_name   end_station_id       start_lat
 Length:5733451     Length:5733451     Length:5733451     Length:5733451     Min.   :41.64
 Class :character   Class :character   Class :character   Class :character   1st Qu.:41.88
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :41.90
                                                                             Mean   :41.90
                                                                             3rd Qu.:41.93
                                                                             Max.   :45.63
                                                                             NA's   :3
   start_lng         end_lat          end_lng        member_casual       hour_start         week_day
 Min.   :-87.84   Min.   : 0.00   Min.   :-88.14   Length:5733451     Length:5733451     Mon:757407
 1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66   Class :character   Class :character   Tue:782733
 Median :-87.64   Median :41.90   Median :-87.64   Mode  :character   Mode  :character   Wed:817125
 Mean   :-87.64   Mean   :41.90   Mean   :-87.64                                         Thu:854086
 3rd Qu.:-87.62   3rd Qu.:41.93   3rd Qu.:-87.62                                         Fri:817407
 Max.   :-73.79   Max.   :42.37   Max.   :  0.00                                         Sat:922061
 NA's   :3        NA's   :5877    NA's   :5877                                           Sun:782632
     month         time_travel
 Jul    : 823488   Length:5733451
 Aug    : 785932   Class :difftime
 Jun    : 769204   Mode  :numeric
 Sep    : 701339
 May    : 634858
 Oct    : 558685
 (Other):1459945
```

- Divide into 2 sub-data frames:

The data frame was divided into 2 sub-data frames named trips_time (focusing on the day, and month) and trips_location (focusing on location), in these 2 sub-data frames we will remove the rows with NAs based on what we focus on, also rows with the NAs in member_casual column.

```
trips_time <- trip_fix2 %>%
  select(-c(start_station_name,start_station_id,end_station_name, end_station_id,
         start_lat,start_lng,end_lat,end_lng)) %>%
  drop_na()
summary(trips_time)
```

```
ride_id            rideable_type       started_at                       ended_at
 Length:5733448     Length:5733448     Min.   :2021-12-01 00:00:01.00   Min.   :2021-12-01 00:02:40.00
 Class :character   Class :character   1st Qu.:2022-05-17 12:04:42.75   1st Qu.:2022-05-17 12:27:03.00
 Mode  :character   Mode  :character   Median :2022-07-13 22:04:36.00   Median :2022-07-13 22:21:55.50
                                       Mean   :2022-07-06 05:55:31.27   Mean   :2022-07-06 06:14:56.44
                                       3rd Qu.:2022-09-07 17:55:41.25   3rd Qu.:2022-09-07 18:11:43.25
                                       Max.   :2022-11-30 23:56:11.00   Max.   :2022-12-01 11:45:53.00

 member_casual       hour_start         week_day         month          time_travel
 Length:5733448     Length:5733448     Mon:757406   Jul    : 823488   Length:5733448
 Class :character   Class :character   Tue:782732   Aug    : 785931   Class :difftime
 Mode  :character   Mode  :character   Wed:817125   Jun    : 769204   Mode  :numeric
                                       Thu:854086   Sep    : 701337
                                       Fri:817406   May    : 634858
                                       Sat:922061   Oct    : 558685
                                       Sun:782632   (Other):1459945
```

```
trips_location <- trip_fix2 %>%
  select(-c(started_at,ended_at,time_travel)) %>%
  drop_na()
summary(trips_location)
```

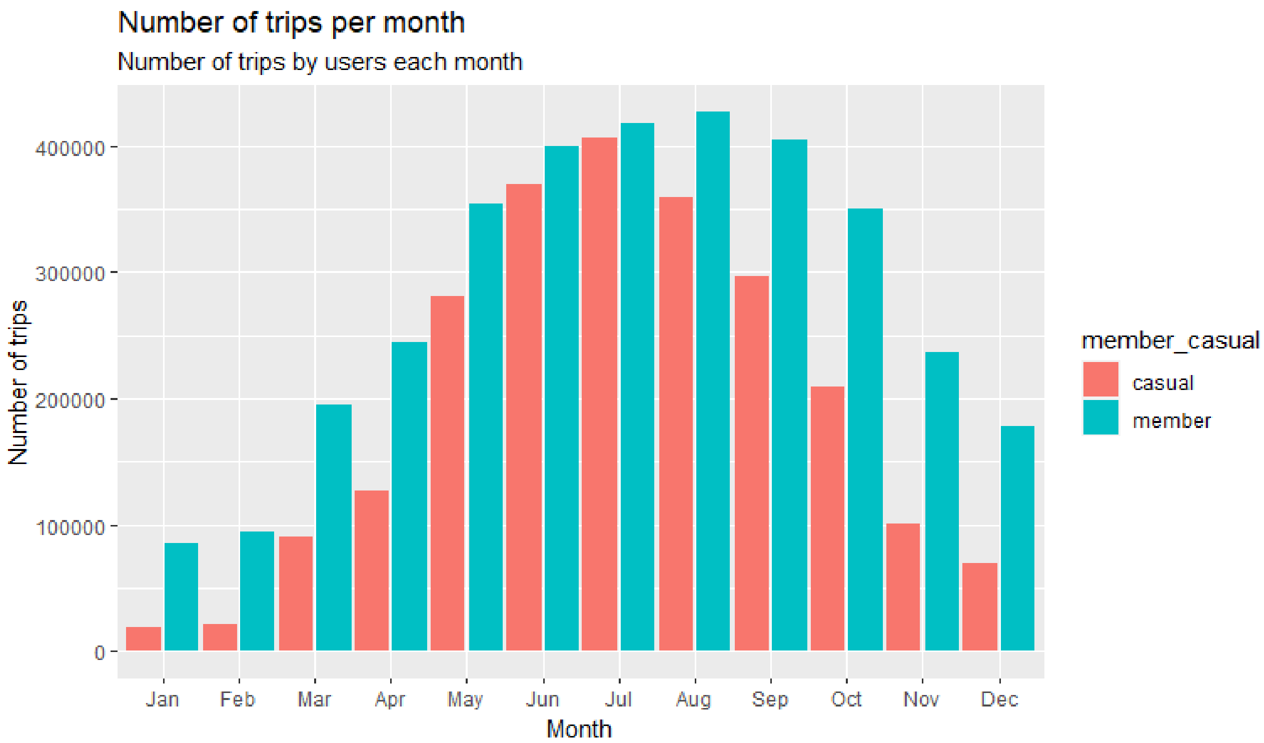
```
ride_id            rideable_type      start_station_name start_station_id   end_station_name
 Length:4118420     Length:4118420     Length:4118420     Length:4118420     Length:4118420
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character



 end_station_id       start_lat       start_lng         end_lat         end_lng        member_casual
 Length:4118420     Min.   :41.64   Min.   :-87.83   Min.   : 0.00   Min.   :-87.83   Length:4118420
 Class :character   1st Qu.:41.88   1st Qu.:-87.65   1st Qu.:41.88   1st Qu.:-87.65   Class :character
 Mode  :character   Median :41.89   Median :-87.64   Median :41.89   Median :-87.64   Mode  :character
                    Mean   :41.90   Mean   :-87.64   Mean   :41.90   Mean   :-87.64
                    3rd Qu.:41.92   3rd Qu.:-87.62   3rd Qu.:41.92   3rd Qu.:-87.62
                    Max.   :45.63   Max.   :-73.79   Max.   :42.06   Max.   :  0.00

  hour_start         week_day         month
 Length:4118420     Mon:556582   Jul    : 642680
 Class :character   Tue:569809   Jun    : 620350
 Mode  :character   Wed:588885   Aug    : 605325
                    Thu:601959   May    : 502545
                    Fri:566180   Oct    : 414269
                    Sat:661158   Apr    : 272560
                    Sun:573847   (Other):1060691
```

Chapter IV. Analyse & Share Phase

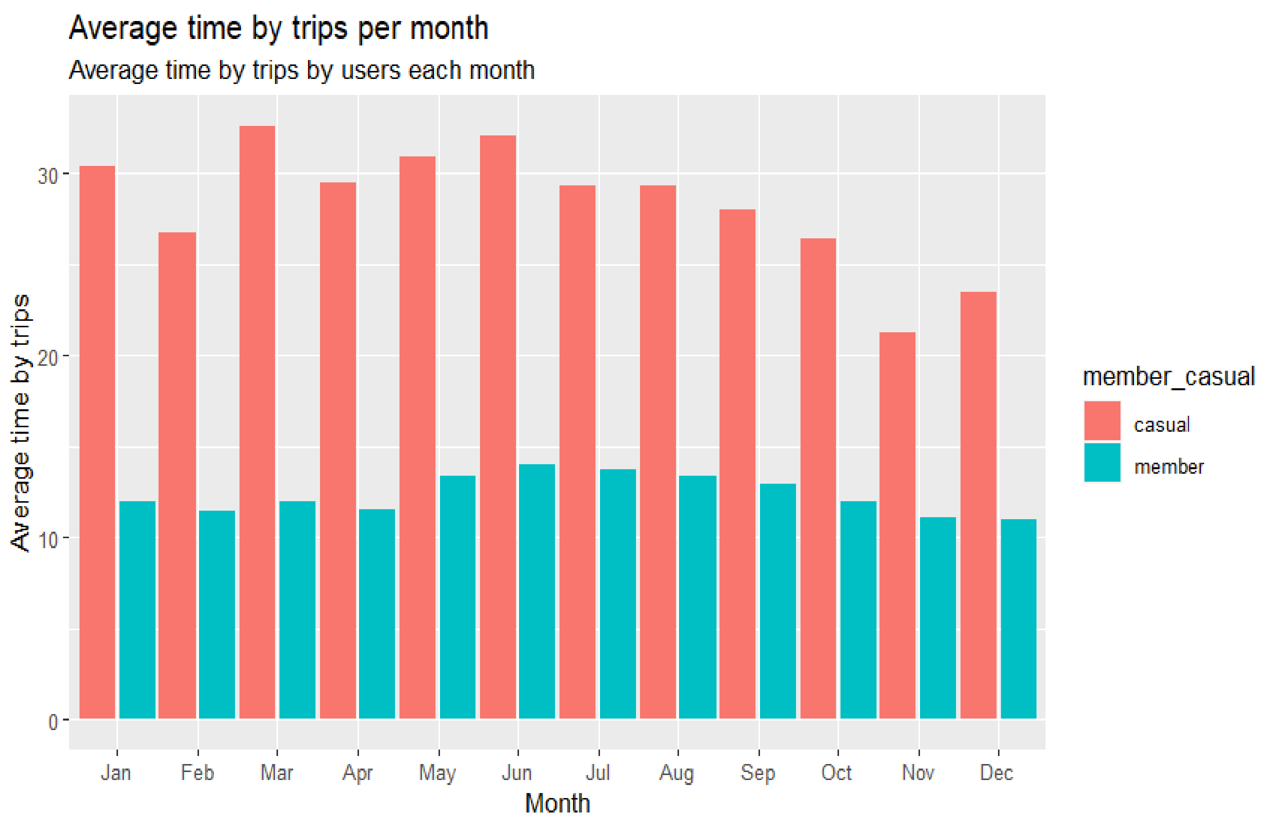
4.1. Analyse and visualise

- Analysis by month:

```
trips_month <- trips_time %>%
  group_by(month,member_casual) %>%
  summarise(
    total_trip = n()
    )
ggplot(trips_month,aes(x=month,y=total_trip,fill=member_casual)) +
  geom_col(position=position_dodge(1)) +
  labs(
    title = "Number of trips per month",
    subtitle = "Number of trips by users each month",
    x = "Month",
    y = "Number of trips"
  )
```

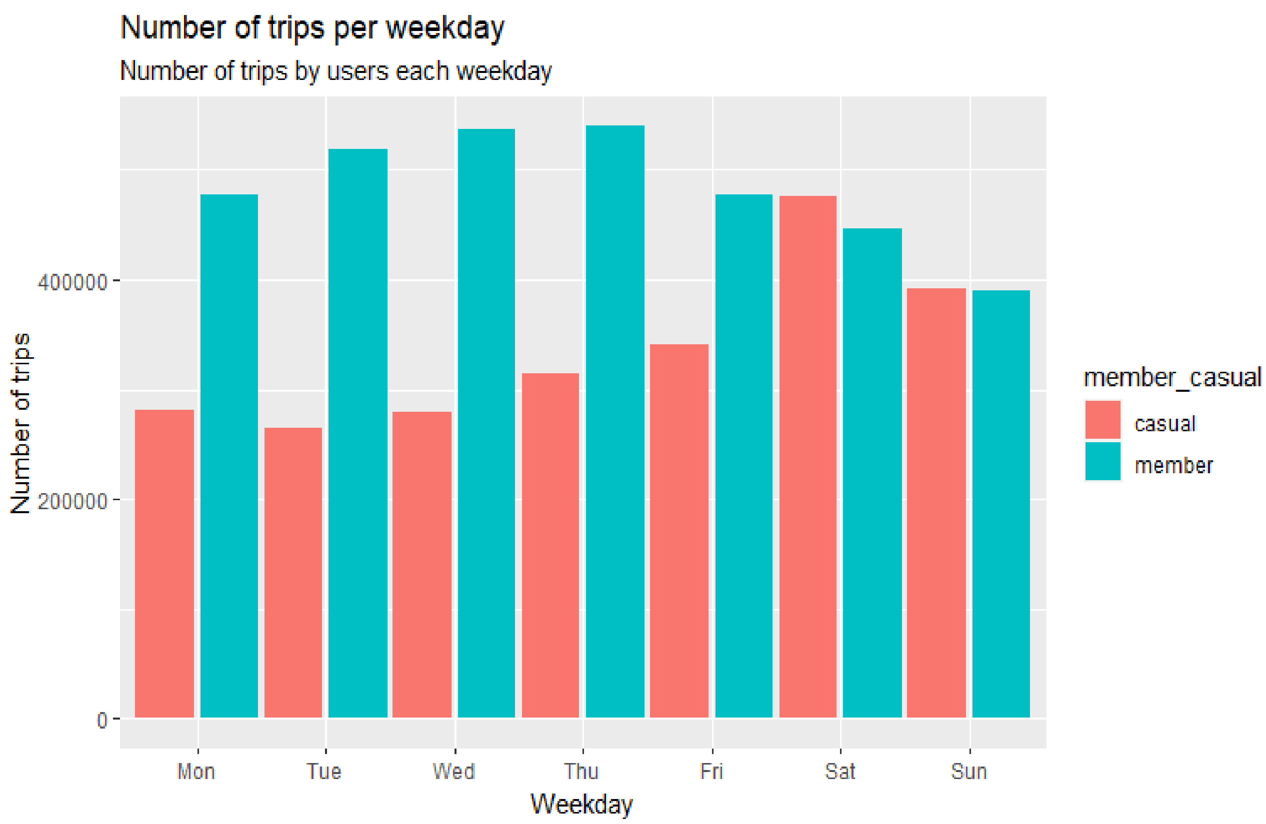


```
trips_month_avgtime <- trips_time %>%
  group_by(month,member_casual) %>%
  summarise(
    avg_time = mean(time_travel)
  )

ggplot(trips_month_avgtime,aes(x=month,y=avg_time,fill=member_casual)) +
  geom_col(position=position_dodge(1)) +
  labs(
    title = "Average time by trips per month",
    subtitle = "Average time by trips by users each month",
    x = "Month",
    y = "Average time by trips"
  )
```

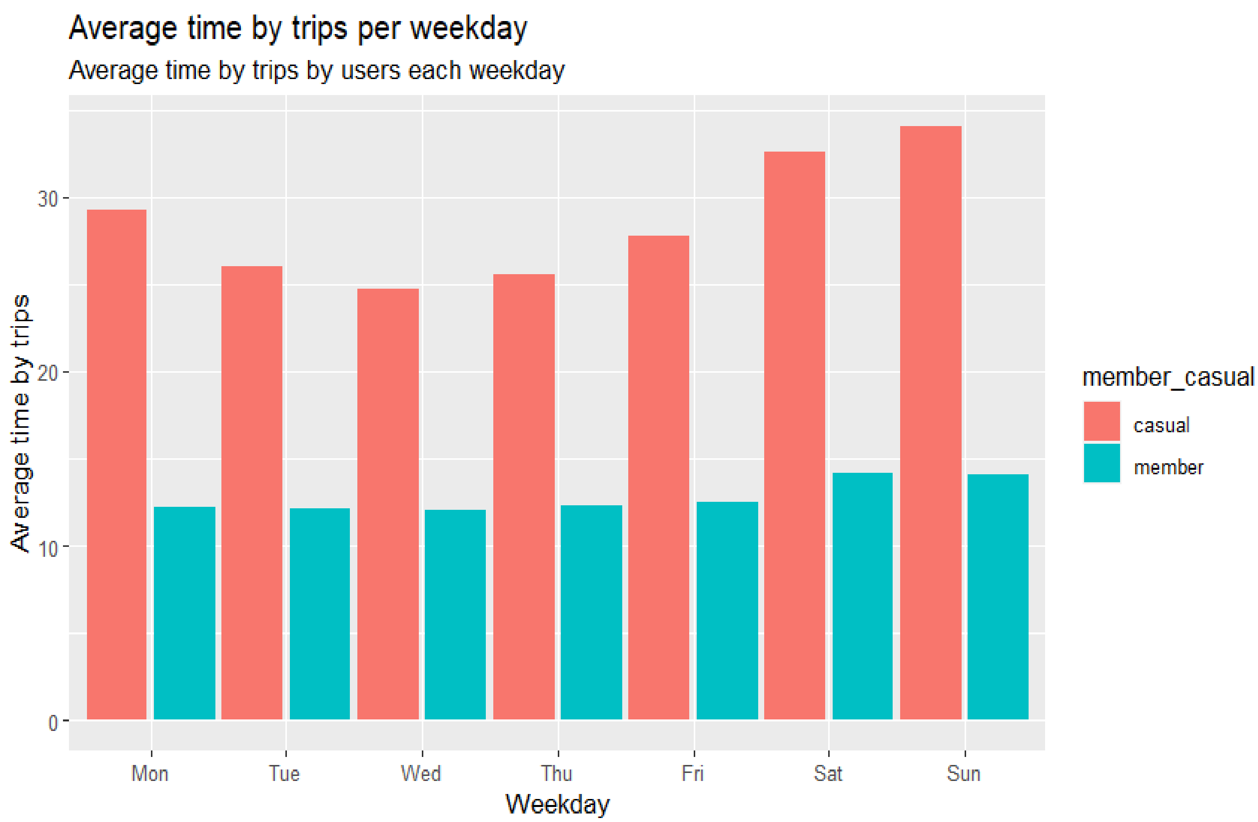


- Analysis by weekday:

```
trips_weekday <- trips_time %>%
  group_by(week_day,member_casual) %>%
  summarise(
    total_trip = n()
    )

ggplot(trips_weekday,aes(x=week_day,y=total_trip,fill=member_casual)) +
  geom_col(position=position_dodge(1)) +
  labs(
    title = "Number of trips per weekday",
    subtitle = "Number of trips by users each weekday",
    x = "Weekday",
    y = "Number of trips"
  )
```

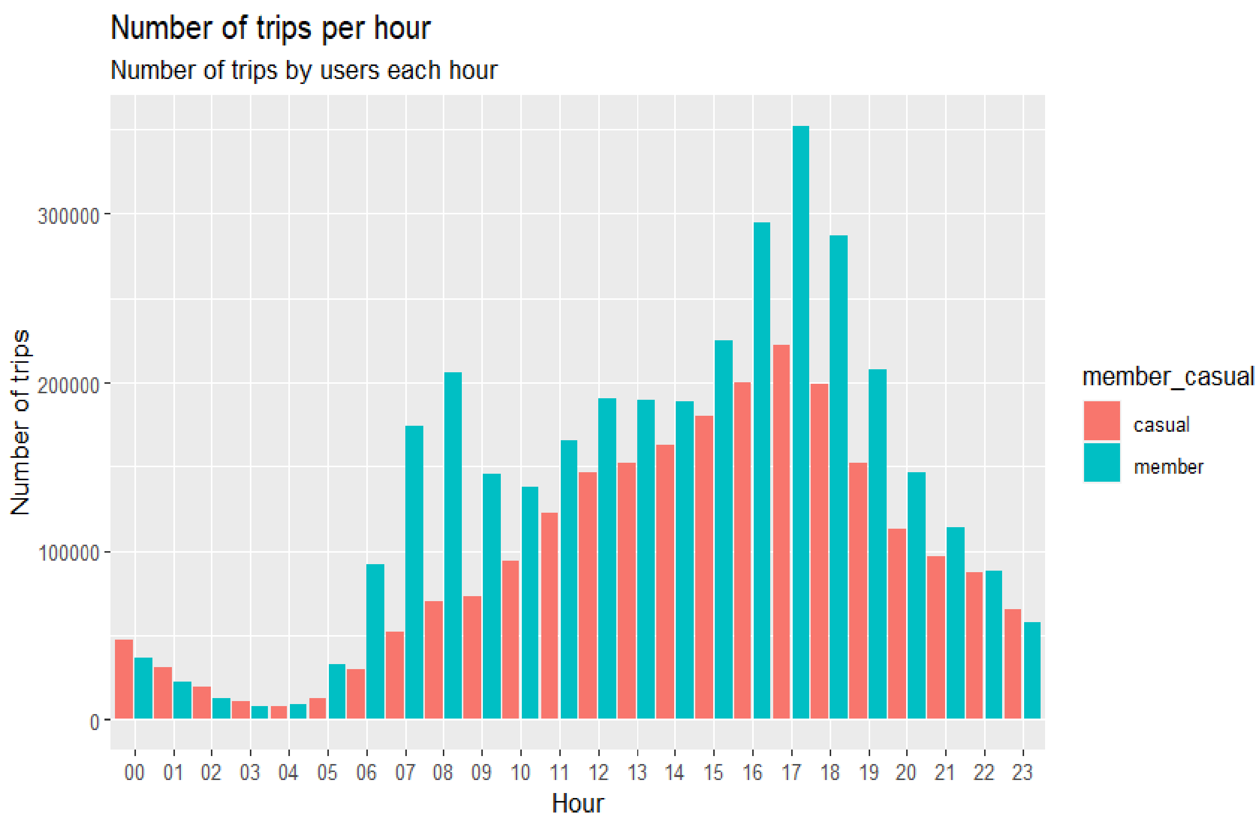


```
trips_weekday_avgtime <- trips_time %>%
  group_by(week_day,member_casual) %>%
  summarise(
    avg_time = mean(time_travel)
  )

ggplot(trips_weekday_avgtime,aes(x=week_day,y=avg_time,fill=member_casual)) +
  geom_col(position=position_dodge(1)) +
  labs(
    title = "Average time by trips per weekday",
    subtitle = "Average time by trips by users each weekday",
    x = "Weekday",
    y = "Average time by trips"
  )
```

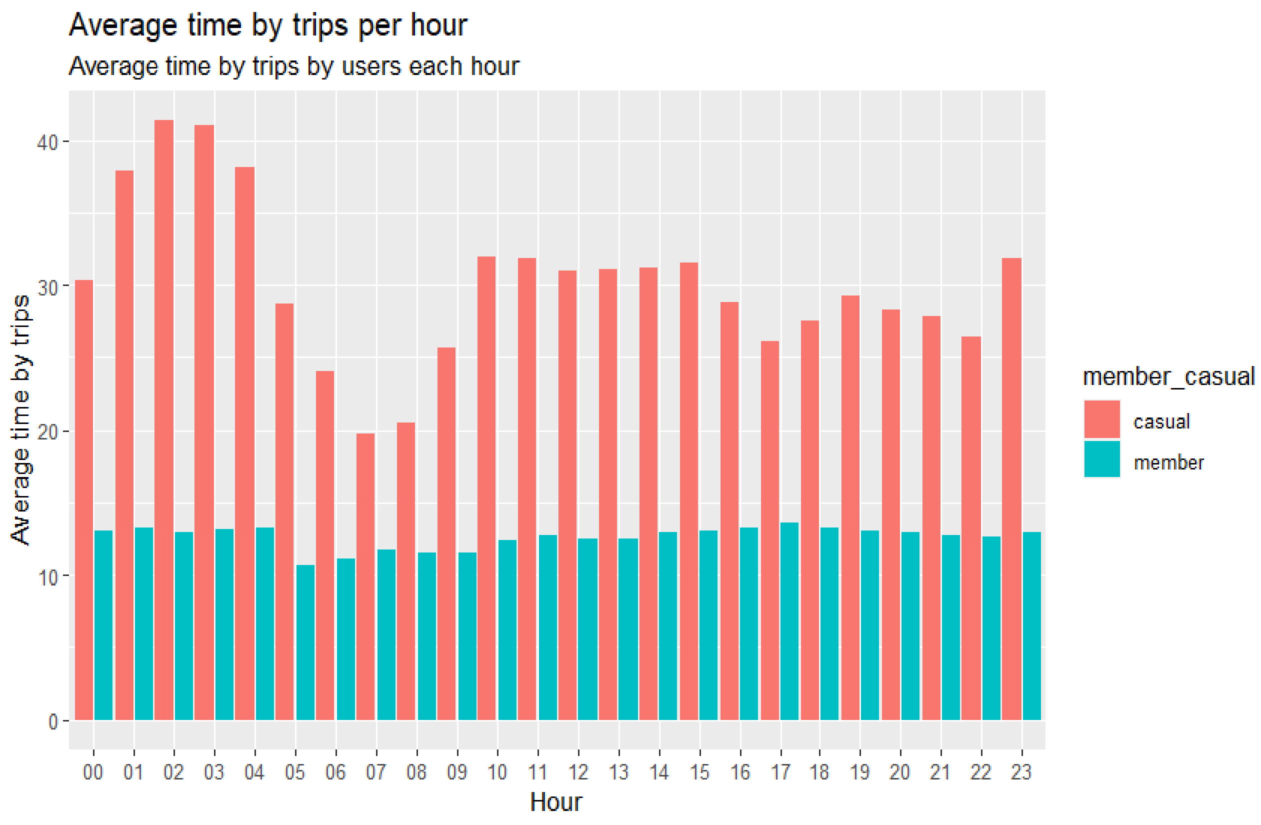


- Analysis by the hour:

```
trips_hour <- trips_time %>%
  group_by(hour_start,member_casual) %>%
  summarise(
    total_trip = n()
  )

ggplot(trips_hour,aes(x=hour_start,y=total_trip,fill=member_casual)) +
  geom_col(position=position_dodge(1)) +
  labs(
    title = "Number of trips per hour",
    subtitle = "Number of trips by users each hour",
    x = "Hour",
    y = "Number of trips"
  )
```

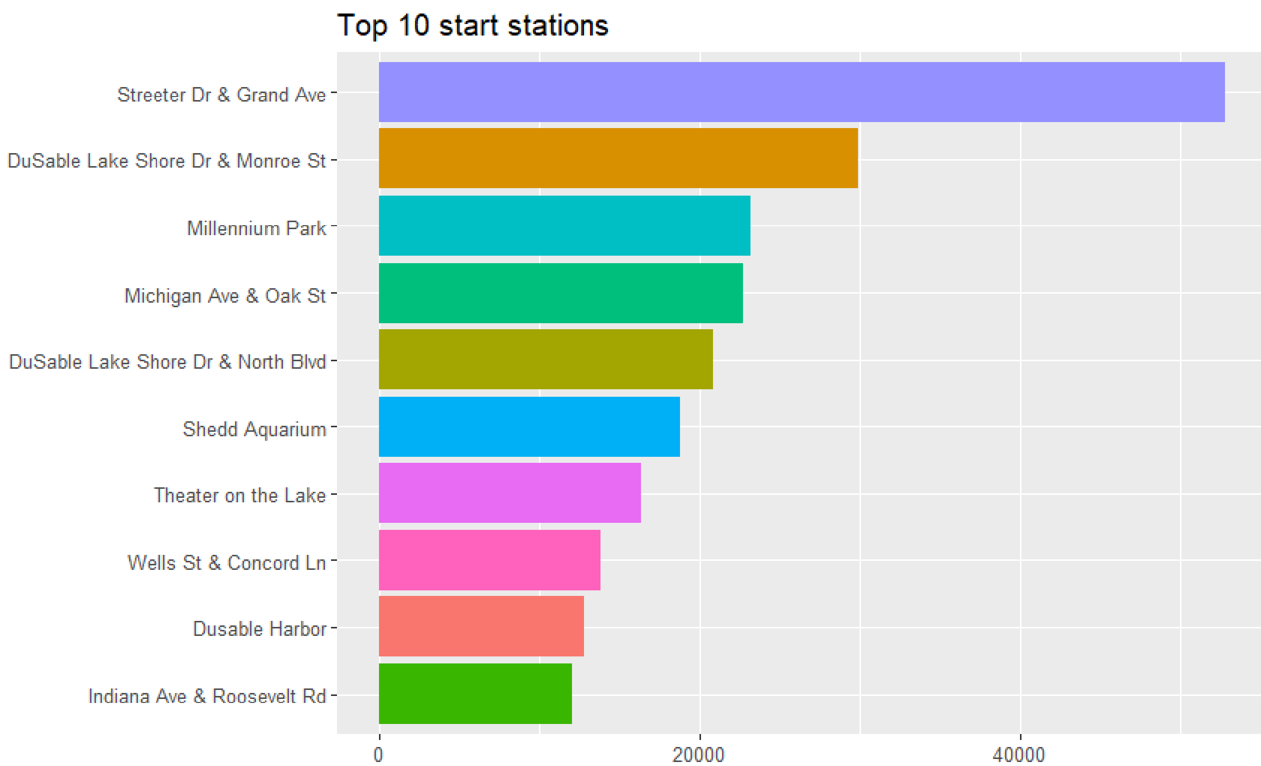


```
trips_hour_avgtime <- trips_time %>%
  group_by(hour_start,member_casual) %>%
  summarise(
    avg_time = mean(time_travel)
  )
ggplot(trips_hour_avgtime,aes(x=hour_start,y=avg_time,fill=member_casual)) +
  geom_col(position=position_dodge(1)) +
  labs(
    title = "Average time by trips per hour",
    subtitle = "Average time by trips by users each hour",
    x = "Hour",
    y = "Average time by trips"
  )
```



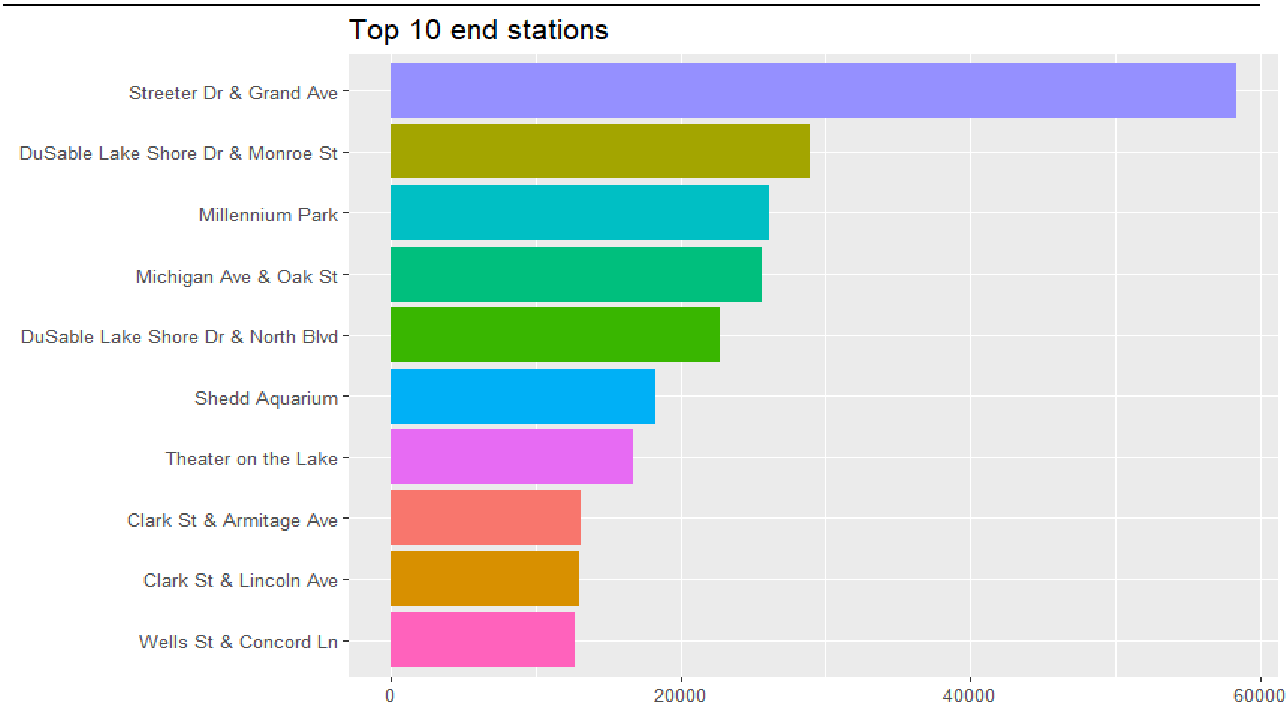
- Top 10 starting stations:

```
top10_start <- trips_location %>%
  group_by(member_casual,start_station_name) %>%
  summarise(
    numride_start = n()
  ) %>%
  filter(member_casual == "casual") %>%
  arrange(-numride_start)
ggplot(top10_start,aes(x=reorder(start_station_name,numride_start), y=numride_start, fill =
start_station_name)) +
  geom_col() +
  coord_flip() +
  labs(title = "Top 10 start stations" ) +
  theme(legend.position="none",
        axis.title.x = element_blank(),
        axis.title.y = element_blank()
  )
```



- Top 10 ending stations:

```
top10_end <- trips_location %>%
  group_by(member_casual,end_station_name) %>%
  summarise(
    numride_end= n()
  ) %>%
  filter(member_casual == "casual") %>%
  arrange(-numride_end) %>%
  slice(1:10)
ggplot(top10_end,aes(x=reorder(end_station_name,numride_end), y=numride_end, fill =
end_station_name)) +
  geom_col() +
  coord_flip() +
  labs(title = "Top 10 end stations" ) +
  theme(legend.position="none",
        axis.title.x = element_blank(),
        axis.title.y = element_blank()
  )
```



4.2. Conclusion

Generally, the number of trips of both types increases gradually in the middle of the year and then decreases towards the end of the year. In which the number of trips of casual members doubled in May and peaked in July. This is understandable given that that period is when students are on summer break. Besides, while casual users have an average time of each trip ranging from 22 to 32 minutes, the average time of member users has not changed too much with only around 14 minutes per trip.

There is a contrast in the number of trips by day of the week of 2 users, while member users tend to be high on weekdays (Monday to Friday), casual users increase gradually on weekends (Saturday and Sunday) and reach a peak on Saturday. This is likely because member users tend to use bicycles to go to work during the week and rest on weekends, while casual users use bicycles for sports activities on weekends.

In terms of average time by day of the week, there is not too much difference with member users, although there is an increase but not significantly at the weekend. Meanwhile, casual users increased from 4 to 5 minutes compared to Monday (weekday has the highest average time) and 8 to 9 minutes compared to Wednesday (weekday has the lowest average time).

During the day, both members and casual users focus mainly in the evening time (from 16h to 19h). In addition, at 7 am and 8 am, casual members are also quite active. It can be explained by the fact that many people use bicycles to go to work.

The average time of member users also did not change much, at about 12 minutes per trip. Meanwhile, casual members have a lot of time difference, the highest is 42 minutes (at 2 am) and the lowest is 20 minutes (at 7 am). We need to learn more about the travel time from 1 am to 4 am because it is quite high compared to other time frames.

There are 7/10 stations in the top 10 start stations that appear in the top 10 end stations.

4.3. Share

The data and my Rstudio file can be found here:

https://drive.google.com/drive/folders/1YPct-0kH3vjUKxRxOVTd1GTs7wXrxMnr?usp=sharing

Chapter V. Act Phase

- Marketing campaigns should take place at the time when casual members are most engaged, in June and July. Besides, weekend afternoons are also a good time because that is when the number of casual members participating is highest.
- Campaign locations should be considered as both popular as starting and starting stations.
- Casual user trips at 1:00 and 2:00 a.m. should be investigated because trip times at that time are the highest.