CyclistBikeShare

Prince Bhalawat

26/09/2021

Problem Statement statement

The bike sharing company wants to analyze their user data to find the main differences in behaviour between their two types of users, the "casual" who pays for each ride and the "annual member" who pays a yearly subscription to the service.

PHASE 1 : ASK

Key objectives:

1.Identify the business task:

- The company wants to improve their earnings reaching out to their "casual" riders, and for that they have to analyze in what aspects the "casual" and the annual customers differ, to be able to create a focused and successful marketing message to the "casual" customers that makes them change to the annual subscription.

2.Consider key stakeholders:

- The main stakeholders here are the director of marketing and my manager Lily Moreno, the rest of the marketing analytics team, and the Cyclistic executive team.

3.The business task:

Given these facts, the business task is defined as searching for differences in the two identified kinds of users in order to make a focused marketing campaign to the "casual" users in order for them to change to the annual subscription, or resumed in a question:

What could motivate the "casual" users to change to an annual subscription based on their behavior?

PHASE 2 : Prepare

Key objectives:

1.Determine the credibility of the data:

- The data is public data from a bike sharing company. It starts from the year 2013 until 2021 (three months), there isn't much of a naming convention as the files are sometimes organized by quarter, or month, or the whole year and their names vary a lot. The naming of the columns also changes and there are some columns added and deleted over the years. Nevertheless the data seems to be in good condition and its first hand data collected by the company itself with lots of entries and with lots of useful data.

2. Data Source:

- Past 12 month of original bike share dataset from 01/06/2020 to 30/05/2021 were extracted as 12 zipped .csv files (https://divvy-tripdata.s3.amazonaws.com/index.html). The data is made available and licensed by Motivate International Inc.

3. Data Organization & Description:

- File naming convention: TripData_YYYY_MM
- File Type: Not converted kept as it is i.e csv to perform R operations.

- File Content: Each excel files contains 13 columns containing information related to ride id, ridership type, ride time and location and location etc. Number of rows varies between 49k to 531k from different excel files.

4. Data Security:

- Riders' personal identifiable information is hidden through tokenization.
- Original files are backed up in a separate folder.

5. Data Limitations:

- As riders' personal identifiable information is hidden, thus will not be able to connect pass purchases to credit cards numbers to determine if casual riders live in the Cyclistic service area or if they have purchased multiple single passes.

6. Sort and filter the data:

- For this analysis I'm going to focus on the 2020-2021 period as it's the more relevant period to the business task and it has the more complete data with geo-location coordinates, and types of bike used.

```
#First step I did is to add all the necessary libraries for the analysis

library("tidyverse")
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     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
```

```
library("geosphere")
library("gridExtra")
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library("ggmap")
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```r
library("readr")
library("dplyr")
```

```r
#I loaded all the data to their respective variables
tripdata_202006 <- read_csv("202006-divvy-tripdata.csv")
```

```
## Warning in unlink(c(requestFile, responseFile)): cannot get info on 'C:/Users/
## admin/AppData/Local/Temp/Rtmp0EEahi/rstudio-ipc-requests-1ea0783030ba.rds',
## reason 'The system cannot find the file specified'
```

```
## Rows: 343005 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```r
tripdata_202007 <- read_csv("202007-divvy-tripdata.csv")
```

```
## Rows: 551480 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```r
tripdata_202008 <- read_csv("202008-divvy-tripdata.csv")
```

```
## Rows: 622361 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202009 <- read_csv("202009-divvy-tripdata.csv")
```

```
## Rows: 532958 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202010 <- read_csv("202010-divvy-tripdata.csv")
```

```
## Rows: 388653 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202011 <- read_csv("202011-divvy-tripdata.csv")
```

```
## Rows: 259716 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202012 <- read_csv("202012-divvy-tripdata.csv")
```

```
## Rows: 131573 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202101 <- read_csv("202101-divvy-tripdata.csv")
```

```
## Rows: 96834 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202102 <- read_csv("202102-divvy-tripdata.csv")
```

```
## Rows: 49622 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202103 <- read_csv("202103-divvy-tripdata.csv")
```

```
## Rows: 228496 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202104 <- read_csv("202104-divvy-tripdata.csv")
```

```
## Rows: 337230 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
tripdata_202105 <- read_csv("202105-divvy-tripdata.csv")
```

```
## Rows: 531633 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
#Then I joined all the data from 2020 and change the type of two columns (start_station_id,end_station_
id) to match the 2021 data:


tripdata_withDoubles <- rbind(tripdata_202006,tripdata_202007,tripdata_202008,tripdata_202009,tripdata_
202010,tripdata_202011,tripdata_202012)

tripdata_withDoubles <-  mutate(tripdata_withDoubles, start_station_id = as.character(start_station_id)
                   ,end_station_id = as.character(end_station_id))


tripdata_withChar <- rbind(tripdata_202101,tripdata_202102,tripdata_202103,tripdata_202104,tripdata_202
105)

#Then I join all the data:


all_tripdata <- rbind(tripdata_withChar,tripdata_withDoubles)
```

PHASE 3 : Process

Key objectives:

1.Clean the data, and prepare the data for analysis:

- Now that we have all the data in one place we can start to clean the data of possible errors like NA. Also we will make some changes to the data adding useful new columns based on calculations of already existing columns in order to facilitate our analysis and arrive at more insightful conclusions.

```
#Lets see that data:
print("####### GLIMPSE #######")
```

```
## [1] "####### GLIMPSE #######"
```

```
glimpse(all_tripdata)
```

```
## Rows: 4,073,561
## Columns: 13
## $ ride_id            <chr> "E19E6F1B8D4C42ED", "DC88F20C2C55F27F", "EC45C94683~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2021-01-23 16:14:19, 2021-01-27 18:43:08, 2021-01-~
## $ ended_at           <dttm> 2021-01-23 16:24:44, 2021-01-27 18:47:12, 2021-01-~
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor~
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660~
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "Wood St & Augu~
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "657", "13258",~
## $ start_lat          <dbl> 41.90034, 41.90033, 41.90031, 41.90040, 41.90033, 4~
## $ start_lng          <dbl> -87.69674, -87.69671, -87.69664, -87.69666, -87.696~
## $ end_lat            <dbl> 41.89000, 41.90000, 41.90000, 41.92000, 41.90000, 4~
## $ end_lng            <dbl> -87.72000, -87.69000, -87.70000, -87.69000, -87.700~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
```

```
print("####### SUMMARY #######")
```

```
## [1] "####### SUMMARY #######"
```

```
summary(all_tripdata)
```

```
##    ride_id          rideable_type        started_at                 
##  Length:4073561     Length:4073561     Min.   :2020-06-03 05:59:59  
##  Class :character   Class :character   1st Qu.:2020-08-07 19:09:29  
##  Mode  :character   Mode  :character   Median :2020-09-30 07:36:28  
##                                        Mean   :2020-11-08 09:07:38  
##                                        3rd Qu.:2021-03-13 10:03:09  
##                                        Max.   :2021-05-31 23:59:16  
##                                                                     
##     ended_at                   start_station_name start_station_id  
##  Min.   :2020-06-03 06:03:37   Length:4073561     Length:4073561    
##  1st Qu.:2020-08-07 19:39:10   Class :character   Class :character  
##  Median :2020-09-30 07:51:42   Mode  :character   Mode  :character  
##  Mean   :2020-11-08 09:31:51                                        
##  3rd Qu.:2021-03-13 10:22:00                                        
##  Max.   :2021-06-10 22:17:11                                        
##                                                                     
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:4073561     Length:4073561     Min.   :41.64   Min.   :-87.87  
##  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.   :42.08   Max.   :-87.52  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   :41.54   Min.   :-88.07   Length:4073561    
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character  
##  Median :41.90   Median :-87.64   Mode  :character  
##  Mean   :41.90   Mean   :-87.64                     
##  3rd Qu.:41.93   3rd Qu.:-87.63                     
##  Max.   :42.16   Max.   :-87.44                     
##  NA's   :5037    NA's   :5037
```

```
#Now lets clean the data to be able to properly work with it:

#Fist we drop all NA:

all_tripdata_clean <- drop_na(all_tripdata)

#Then lets create some new columns.

#First lets separate the dates into month, day, year and day of the week:

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

#Then lets make some useful new columns with the duration of the ride, distance traveled, and speed:

#First the ride length in seconds:
all_tripdata_clean$ride_length <- difftime(all_tripdata_clean$ended_at,all_tripdata_clean$started_at)

#Then the ride distance traveled in km
all_tripdata_clean$ride_distance <- distGeo(matrix(c(all_tripdata_clean$start_lng, all_tripdata_clean$s
tart_lat), ncol = 2), matrix(c(all_tripdata_clean$end_lng, all_tripdata_clean$end_lat), ncol = 2))
all_tripdata_clean$ride_distance <- all_tripdata_clean$ride_distance/1000

#At last the speed in Km/h
all_tripdata_clean$ride_speed = c(all_tripdata_clean$ride_distance)/as.numeric(c(all_tripdata_clean$rid
e_length), units="hours")
```

```
# The dataframe includes a few hundred entries when bikes were taken out of docks and checked for quali
ty by Divvy or ride_length was negative:

all_tripdata_clean <- all_tripdata_clean[!(all_tripdata_clean$start_station_name == "HQ QR" | all_tripd
ata_clean$ride_length<0),]
```

PHASE 4 : Analyze

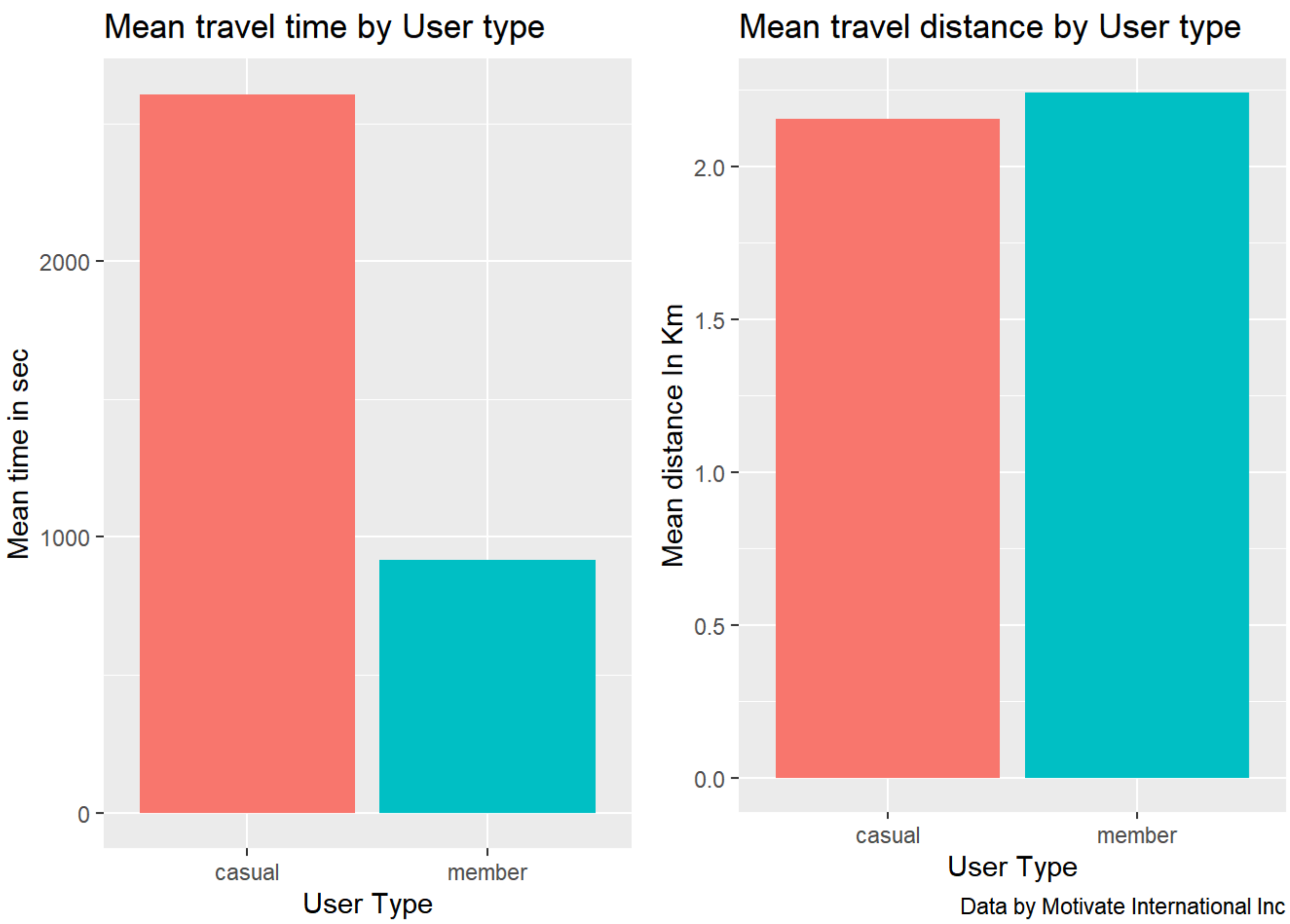
Key objectives:

1.Identify trends and relationships.:

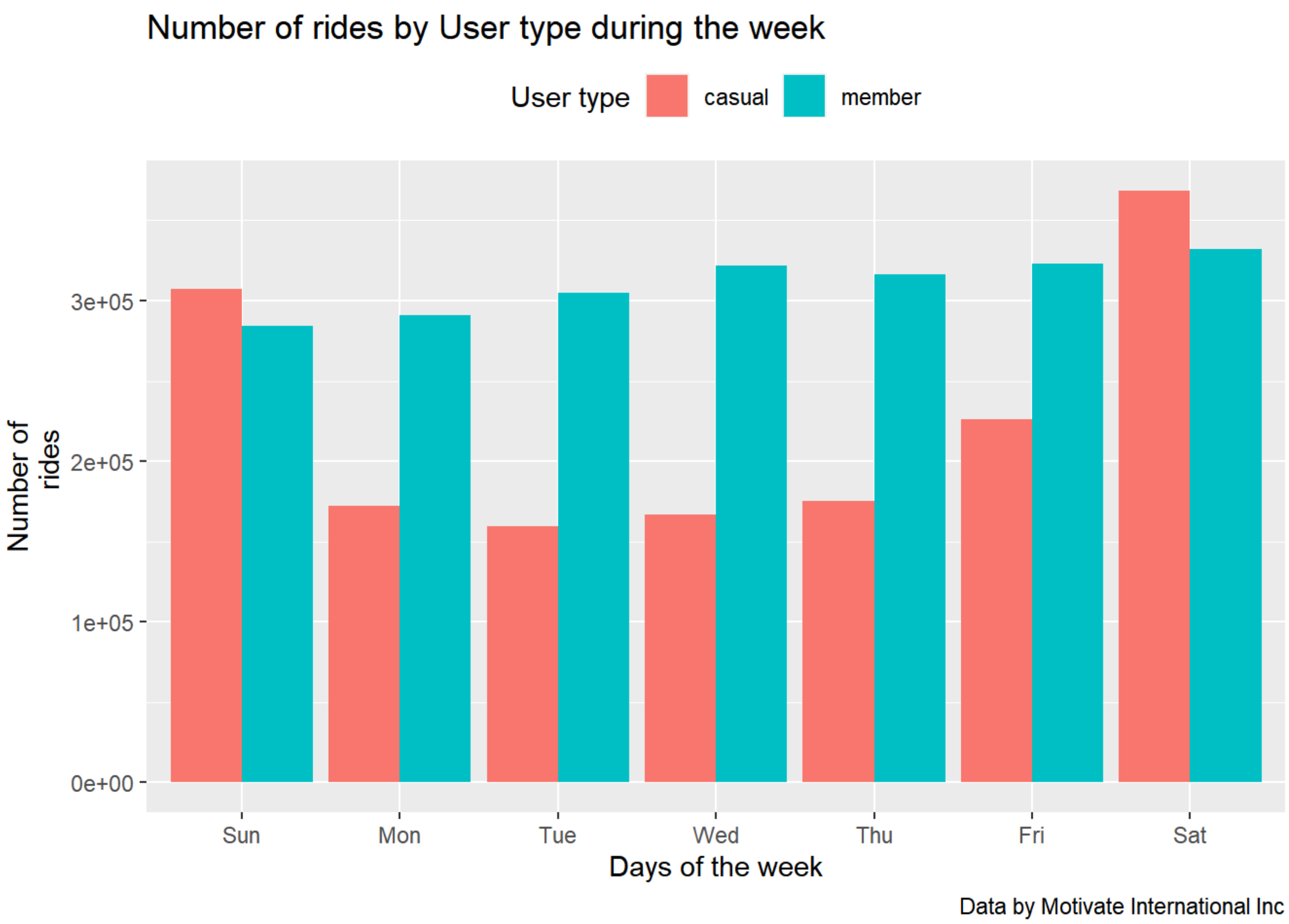
- We have now a complete data frame with all the info we need to identify the differences in behaviour between the casual and the member users.

```
#Fist we calculate the average distance, distance for both the casual and member type users:

userType_means <- all_tripdata_clean %>%
  group_by(member_casual) %>%
  summarise(mean_time = mean(ride_length),mean_distance = mean(ride_distance))

membervstime <- ggplot(userType_means) +
                geom_col(mapping=aes(x=member_casual,y=mean_time,fill=member_casual), show.legend = FAL
SE)+
                labs(title = "Mean travel time by User type",x="User Type",y="Mean time in sec")

membervsdistance <- ggplot(userType_means) +
                    geom_col(mapping=aes(x=member_casual,y=mean_distance,fill=member_casual), show.lege
nd = FALSE)+
                    labs(title = "Mean travel distance by User type",x="User Type",y="Mean distance In
 Km",caption = "Data by Motivate International Inc")

grid.arrange(membervstime, membervsdistance, ncol = 2)
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



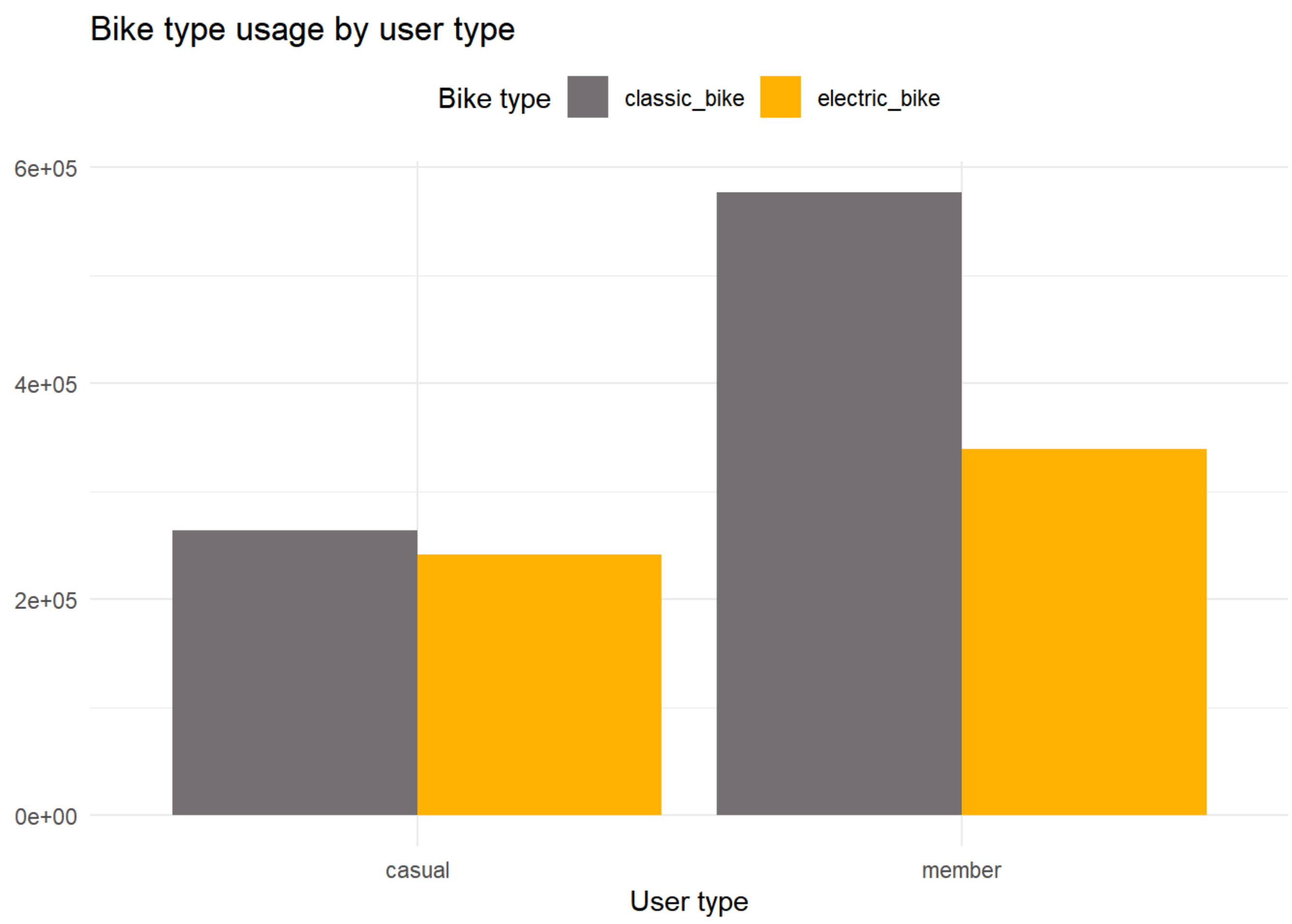
```
#The we check  the number of rides diferences by weekday:
all_tripdata_clean %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),average_duration = mean(ride_length),.groups = 'drop') %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Number of rides by User type during the week",x="Days of the week",y="Number of
      rides",caption = "Data by Motivate International Inc", fill="User type") +
  theme(legend.position="top")
```



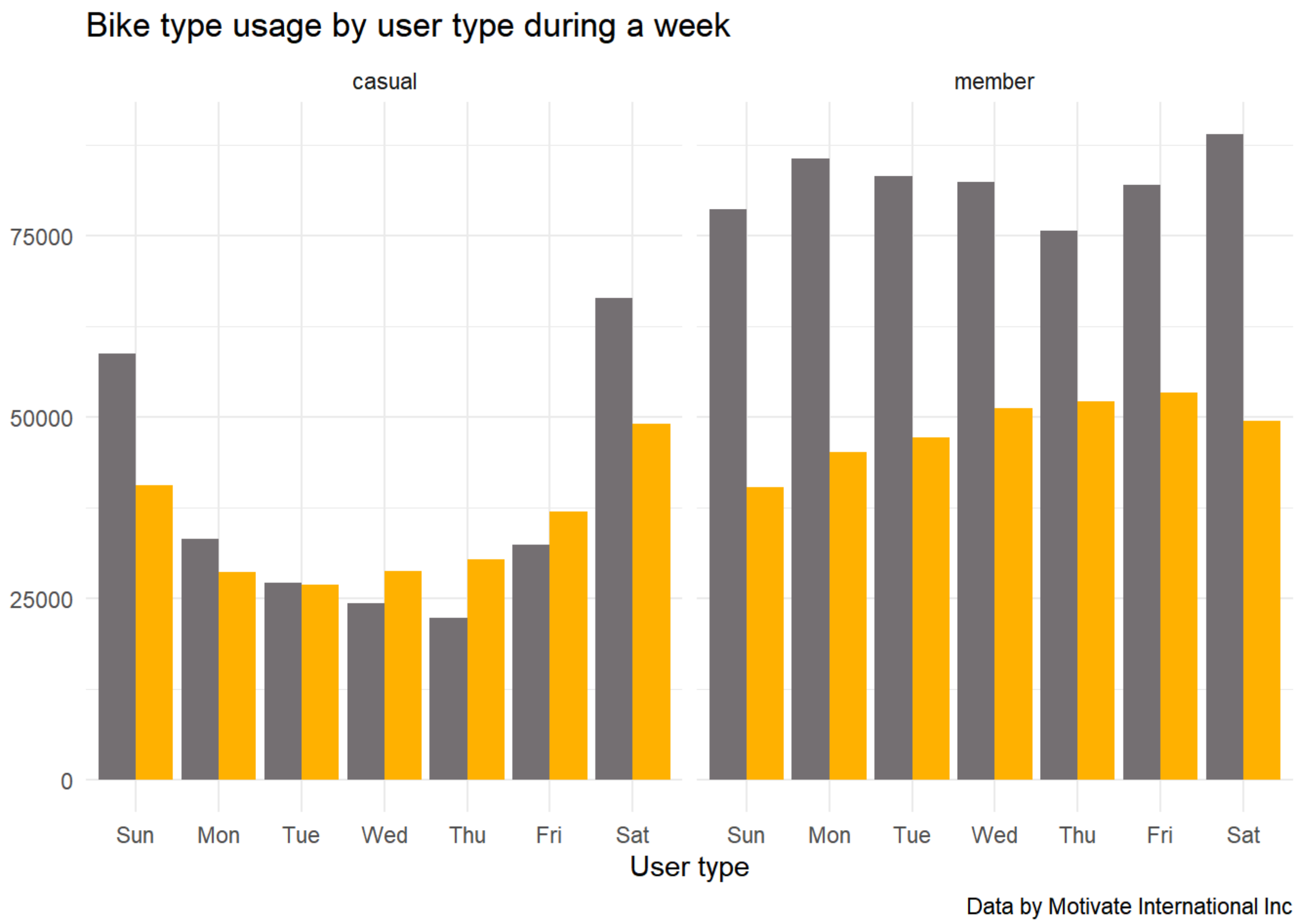
Analysis:

- We can see from the bar graph that the member riders take more number of rides on weekdays whereas the number of rides taken by casual riders are more on Saturdays and Sundays. Saturdays have the highest number of riders for both casual and member.
- From the above graph it is clear that the average ride duration (time) on a daily basis for casual riders is very much higher than the member riders.
- It seems that the casual users travel the same average distance than the member users, but they have much longer rides, that would indicate a more leisure oriented usage vs a more “public transport” or pragmatic use of the bikes by the annual members.
- This idea is reinforced by the fact that annual users have a very stable use of the service during the week, but the casual users are more of a weekend user.

```
#Create a new data frame with only the rows with info in the "bike type" column:

with_bike_type <- all_tripdata_clean %>% filter(rideable_type=="classic_bike" | rideable_type=="electri
c_bike")

#Then lets check the bike type usage by user type:

with_bike_type %>%
    group_by(member_casual,rideable_type) %>%
    summarise(totals=n(), .groups="drop")  %>%

ggplot()+
    geom_col(aes(x=member_casual,y=totals,fill=rideable_type), position = "dodge") +
    labs(title = "Bike type usage by user type",x="User type",y=NULL, fill="Bike type") +
    scale_fill_manual(values = c("classic_bike" = "#746F72","electric_bike" = "#FFB100")) +
    theme_minimal() +
    theme(legend.position="top")
```



```
#And their usage by both user types during a week:

with_bike_type %>%
    mutate(weekday = wday(started_at, label = TRUE)) %>%
    group_by(member_casual,rideable_type,weekday) %>%
    summarise(totals=n(), .groups="drop") %>%

ggplot(aes(x=weekday,y=totals, fill=rideable_type)) +
  geom_col(, position = "dodge") +
  facet_wrap(~member_casual) +
  labs(title = "Bike type usage by user type during a week",x="User type",y=NULL,caption = "Data by Mot
ivate International Inc") +
  scale_fill_manual(values = c("classic_bike" = "#746F72","electric_bike" = "#FFB100")) +
  theme_minimal() +
  theme(legend.position="none")
```



Analysis:

- Here we can see that the annual members use both types of bikes for their rides, but the casual users show a clear preference for the electric bikes, which makes sense given the long duration of their rides.
- On a weekly basis we can see that for the annual members there is a small difference of usage between the start of the week, where they prefer the classic bike and the end of the week, where they use more electric bikes.
- For the casual users we see in general the same pattern of usage from the previous weekly charts, preferring the electric vs the classic bikes and having a weekend usage of the service.

```r
#Lets check now the coordinates data of the rides, to see if is there any interesting pattern:

#First we create a table only for the most popular routes (>250 times)
coordinates_table <- all_tripdata_clean %>%
filter(start_lng != end_lng & start_lat != end_lat) %>%
group_by(start_lng, start_lat, end_lng, end_lat, member_casual, rideable_type) %>%
summarise(total = n(),.groups="drop") %>%
filter(total > 250)

#Then we create two sub tables for each user type
casual <- coordinates_table %>% filter(member_casual == "casual")
member <- coordinates_table %>% filter(member_casual == "member")

#Lets store bounding box coordinates for ggmap:
chi_bb <- c(
  left = -87.700424,
  bottom = 41.790769,
  right = -87.554855,
  top = 41.990119
)

#Here we store the stamen map of Chicago
chicago_stamen <- get_stamenmap(
  bbox = chi_bb,
  zoom = 12,
  maptype = "toner"
)
```

```
## Source : http://tile.stamen.com/toner/12/1050/1520.png
```

```
## Source : http://tile.stamen.com/toner/12/1051/1520.png
```

```
## Source : http://tile.stamen.com/toner/12/1050/1521.png
```

```
## Source : http://tile.stamen.com/toner/12/1051/1521.png
```

```
## Source : http://tile.stamen.com/toner/12/1050/1522.png
```

```
## Source : http://tile.stamen.com/toner/12/1051/1522.png
```

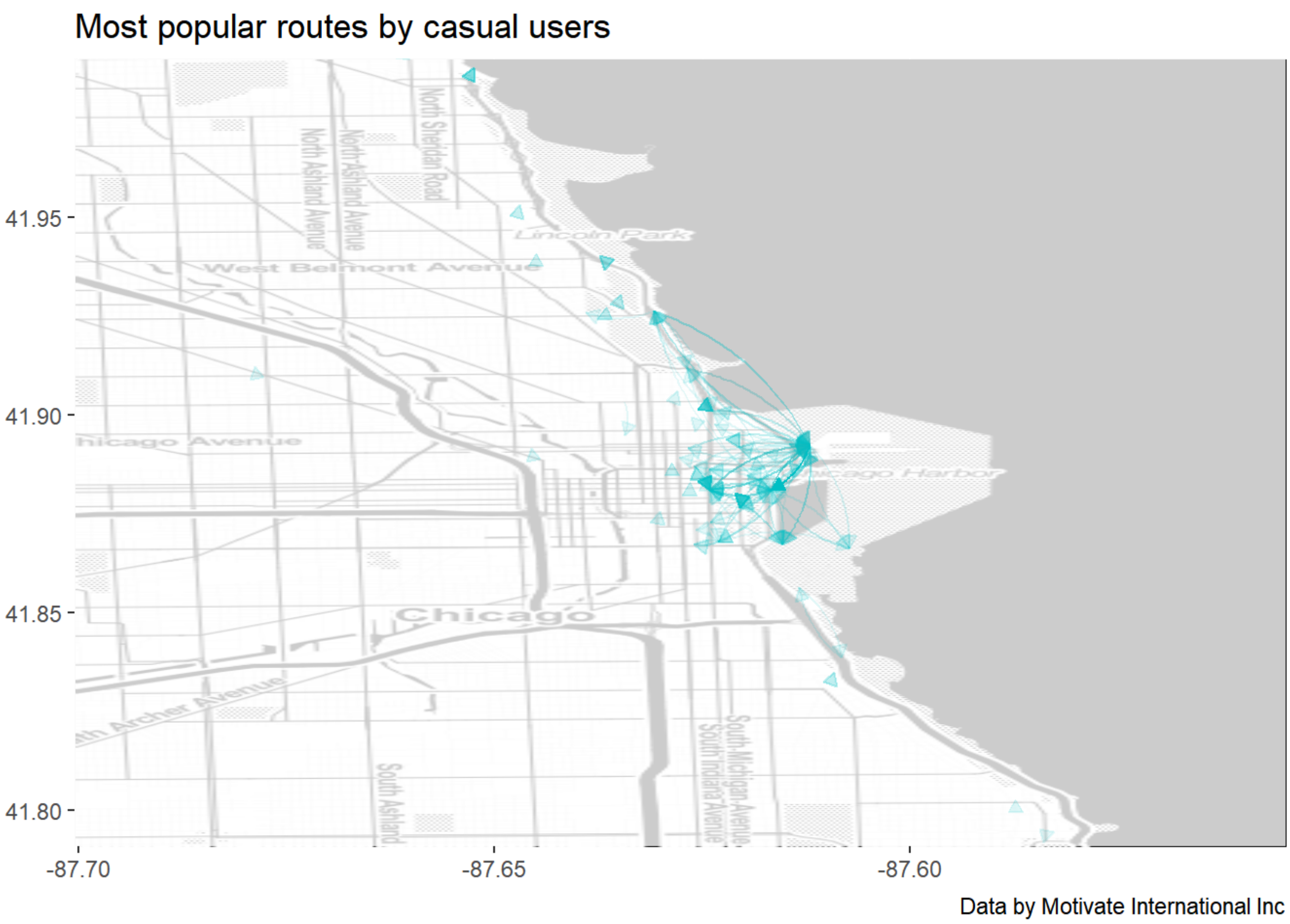
```
## Source : http://tile.stamen.com/toner/12/1050/1523.png
```

```
## Source : http://tile.stamen.com/toner/12/1051/1523.png
```

```
#Then we plot the data on the map
ggmap(chicago_stamen,darken = c(0.8, "white")) +
   geom_curve(casual, mapping = aes(x = start_lng, y = start_lat, xend = end_lng, yend = end_lat, alpha = total, color=rideable_type), size = 0.5, curvature = .2,arrow = arrow(length=unit(0.2,"cm"), ends="first", type = "closed")) +
    coord_cartesian() +
    labs(title = "Most popular routes by casual users",x=NULL,y=NULL, color="User type", caption = "Data by Motivate International Inc") +
    theme(legend.position="none")
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one.
```

```
## Warning: Removed 4 rows containing missing values (geom_curve).
```

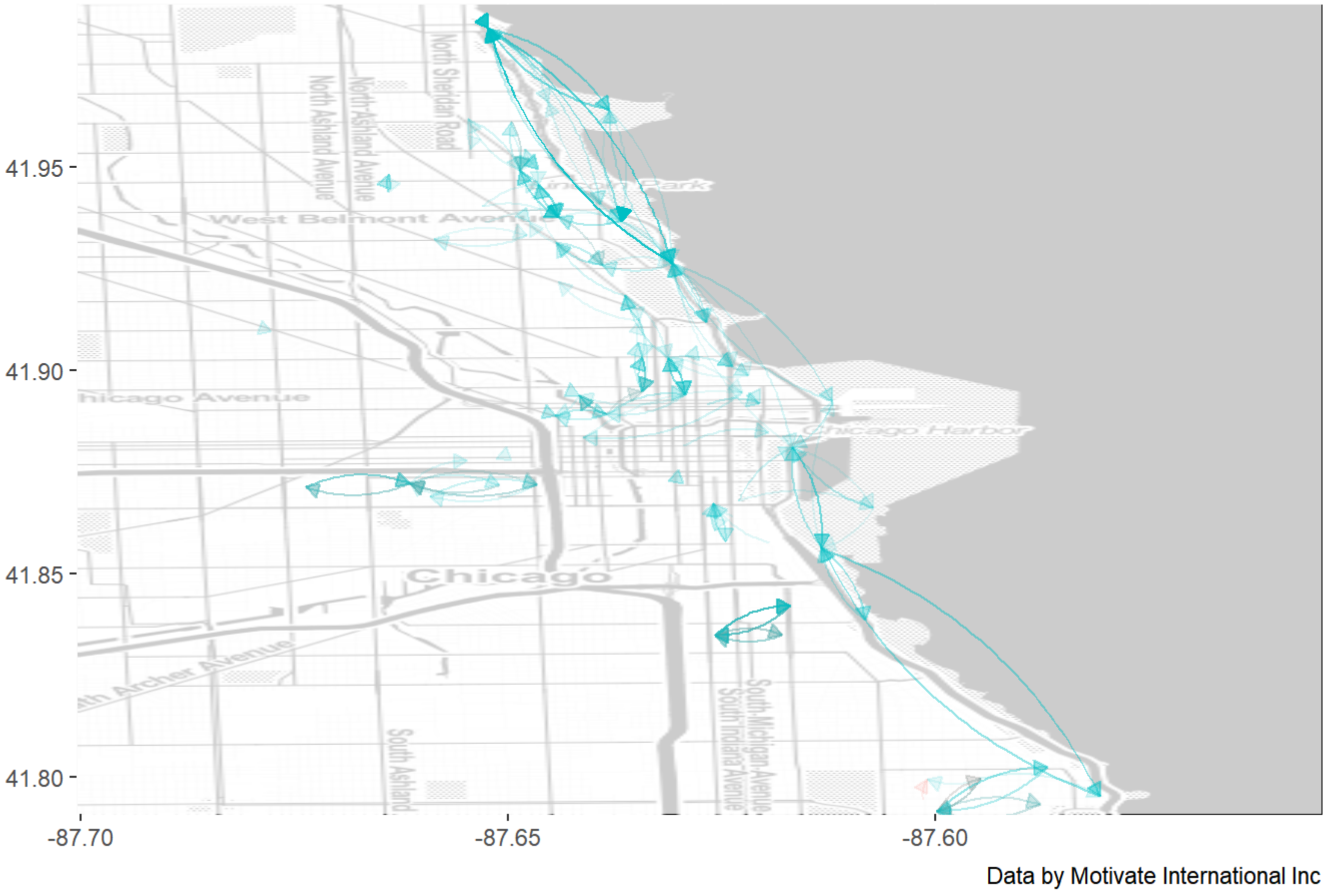



```
ggmap(chicago_stamen,darken = c(0.8, "white")) +
    geom_curve(member, mapping = aes(x = start_lng, y = start_lat, xend = end_lng, yend = end_lat, alpha= total, color=rideable_type), size = 0.5, curvature = .2,arrow = arrow(length=unit(0.2,"cm"), ends="first", type = "closed")) +
    coord_cartesian() +
    labs(title = "Most popular routes by annual members",x=NULL,y=NULL, caption = "Data by Motivate International Inc") +
    theme(legend.position="none")
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one.
```

```
## Warning: Removed 9 rows containing missing values (geom_curve).
```

Most popular routes by annual members



Analysis:

- The coordinates data resulted to be very interesting, as we can clearly see the casual is usually located around the center of the town, with all their trips located around that area which makes sense given that they have a more relaxed leisure rides, on weekends probably also tourist or sightseeing related rides, that naturally focus more on the downtown area where most of the interest points are.
- This contrasts heavily with the longer range of the annual users that connect the downtown with the outskirts of the city, that would suggest they are mostly people that live outside the downtown and use the service to commute everyday to their works in the city.

PHASE 5 : Share

Key objectives:

1.Sharing my conclusions.:

- Taking in consideration both the business task: **What could motivate the "casual" users to change to an annual subscription based on their behavior?** and the insights we've learned from the available data we can make some conclusions.

  1)**The Casual users** have **leisure**, and **tourism** rides mostly on **weekends** and using **electric bikes**.

  2)**The Annual users** have **commute** or **pragmatic** rides, during **all week** using both **electric & classic bikes**

- I would share this info, the data and my analysis to the marketing team, and I would suggest that in order to **convert the casual to the annual** users it would be interesting to focus the messages on the **leisure** aspect of the service, and maybe offer some kind of **promotion related to weekends and/or electric bikes**.

Phase 6 : Act

Summary of the insights gained from Visualization

Key Findings & Insights

-Over a period of 1 year the maximum number of rides are taken between 3p.m and 6p.m and the lowest number of rides between 12a.m and 5a.m. -The casual riders throughout the week have substantially higher average ride duration (almost double) as compared to the member riders that means the casual member take rides for larger distance wheres the member riders take rides for shorter distance mostly. -The number of rides for casual riders increases even more on weekends whereas for member riders the amount of rides remains somewhat constant throughout the week. -The number of rides taken by both memeber and casual riders on a monthly basis increases drastically between april and august (summers) and it is peak in august whereas the number of rides taken by both the rider types decreases considerably after the month of august and it is the lowest in january and february (winters). -The number of rides taken by the member rider is much higher than the rides taken by the casual rider on a daily basis except for weekends where it is somewhat similar.

In additional to sharing the insights gathered to Lily Monero and the executive stakeholder. *I would like to propose a few recommendations based on data evidence:*

- Based on the trips made, the marketing campaign should be launched between February to August as the number of trips made by cyclists starts to build up.
- As casual rider usage often peaks on the weekend, the marketing campaign can include weekend only membership membership subscription at lower price to attract casual riders to convert to members
- Modification to membership subscription, such as ride length based charges which charges lesser as ride length increases. This provides more incentive for the member rides to cycle longer distances. With such modification, it could also encourage casual riders to convert to members to enjoy the ride length discounts.

Thank you for reading my case study project until the end, hope you have enjoyed it! (: