Market Analysis : Cyclistic Bike share program

R Amarthya Sreechand

2025-06-29

Contents

Personal Objective :

- **This project was undertook to impart a systematic mindset for solving complex problems, and thus is the beginning of my learning journey in building and refining a system on top of the Google data analysis process for solving complex problems.**
- The following is a system for analyzing data which anyone can apply to real world analyses. To put this system out into the world, and for myself to refer in my upcoming projects, I used the data of a bike sharing company called 'Cyclistic' (Divvy).

MARKET ANALYSIS : Cyclistic Bike Share Program

- You can skip to the final slideshow report section for the Summary of the Analysis. Or, you can view it here : Click Here to view the Final Slideshow Report
- Check the Changelog here : Click here to view the Changelog

CONTEXT :

Cyclistic is a bike-share program that features more than 5,800 bicycles and 600 docking stations. Cyclistic users are more likely to ride for leisure, but about 30% use the bikes to commute to work each day. 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, maximizing the number of annual members will be key to future growth.

Therefore, Cyclistic marketing team has set a clear goal :

- **Design marketing strategies aimed at converting casual riders into annual members.**

Factorization Method:

For achieving this goal, I need to understand how (Categories) Annual members & Casual riders differ in their behavioral patterns or trends.

I will be using the Factorization method for analyzing the data. Where, Quantitative data is classified into Environmental and Engagement factors. Then, the **Data patterns or trends are discovered by combining Environmental factor/s with Engagement factors.**

Qualitative data -

Environmental/Contextual factors (Categories):

- Time (Eg; Weekday, Month, …)
- Location/Geography (Eg; Bike Stations, Routes, City, …)
- User Types (Eg; Member/Casual, User/Customer/Lead, …)
- Traffic
- Weather/Climate
- Human Events (Seasonal/ Recurring fests, …)
- …

Quantitative data -

Engagement factors (Measurable):

- Values (Eg; Trip duration, Trip frequency, …)

Actionable insights developed from the data insights (trends/patterns) are implemented on the the individual or on the combination of Contextual factors (Categories). So I would have to keep them in the back of my mind for asking the right questions.

1. ASK PHASE (Deliverable = Business Task)

This is the MOST IMPORTANT Phase of the Data analysis process. If you got this wrong, then the error would get multiplied in the final result ie; the actionable insights you recommend would be heavily misleading.

So I will take my time to think through this, and use a systematic process to ASK the right questions to define the right Business task to solve the problem.

1. Problem statement : What are the differences in behavioral patterns between Casual riders and Annual members from which we can produce actionable marketing strategies to convert Casual riders to Annual members.

2. Questions (Used Factorization method to extract the Right Questions) :

Right questions are SMART questions (Specific, Measurable, Actionable, Relevant, and Time bound). Right questions are asked to prepare a Business task and the Scope of Work.

As my investigation is into the difference between Casual riders and Annual members, all of my questions would include Member/Casual User Types (Contextual Factor) as the CORE FACTOR.

a) **Time :**

- What is the average trip duration/ ride count/ total trip duration of Casual riders & Annual members for each **month** in the time period 2019-2020 Q1 ?
- What is the average trip duration/ ride count/ total trip duration of Casual riders & Annual members for each **day of the week** in the time period 2019-2020 Q1 ?

b) **Location/Geography :**

- What are the **stations** with highest Ride counts for Casual riders & Annual members respectively from the year 2019 to 2020 Q1 ?
- Which **route/s** (station pairs) have the highest ride counts/ highest Total trip duration/ highest average trip duration with respect to Casual riders and Annual members from the year 2019 to 2020 Q1 ?

c) **User characteristics :**

- Which of the **genders** have the highest Ride counts/ highest Total trip duration/ highest average trip duration with respect to Casual riders and Annual members in the time period 2019-2020 Q1 ?
- Which of the **age groups** have the highest Ride counts/ highest Total trip duration/ highest average trip duration with respect to Casual riders and Annual members in the time period 2019-2020 Q1 ?

Using these questions, let's create the Scope of Work -

Scope of Work :

Data Analyst : R Amarthya Sreechand

Client : Cyclistic

Business task : Identify daily and monthly patterns in trip duration and frequency among casual riders and annual members, segmented by gender, age, station activity, and route preferences, to develop targeted marketing strategies that convert casual riders into annual members.

Major Activities :

- Collect Data which is relevant, original, comprehensive, current & cited
- Organized and Clean the Data
- Analyze the Data & identify customer behavior systematically
- Share the top 3 high leverage recommendations based on analysis
- Final report for easy comprehension of the analysis

Project Deliverables :

- A clear statement of the business task.
- A description of all data sources used.
- Documentation of any cleaning or manipulation of data (Changelog).
- A summary of analysis.
- Supporting visualizations and key findings (Final Report).
- Top 3 recommendations based on analysis.

Project does not include :

- Any scope outside the existing stations.

- Cost-Benefit analysis

Major Milestones :

- Data preparation and processing : 25/05/2025
- Data analysis : 29/05/2025
- Final report with recommendations : 31/05/2025

Estimated Completion Date : 31/05/2025

2. PREPARE PHASE (Deliverable = Description of all data sources used)

Now, I have to find the datasets which are Reliable, Original, Complete, Current and Cited, that are relevant for solving the Business task.

Collection of Data :

- Data sources : To collect reliable, original, complete, latest and cited data, I will be downloading the company's latest data from the Cyclistic data repository.
  - Divvy_Trips_2019_Q1.zip
  - Divvy_Trips_2019_Q2.zip
  - Divvy_Trips_2019_Q3.zip
  - Divvy_Trips_2019_Q4.zip
  - Divvy_Trips_2020_Q1.zip
- Collected & stored Cyclistic's data for the :
  - Year 2019
  - First Quarter of the year 2020

Selecting the appropriate tools :

An important Decision - For viewing the data's organization, and for efficient analysis.

The Tools are selected using these Criteria :

a) Size of the Datasets :

- '< 10 MB' - Use Excel, Google sheets
- '> 10 MB' - Use SQL, R

b) Storage :

- Database - Use SQL (Eg; BigQuery)
- Cloud / Local storage - Use Google sheets, Excel, R.

The Logical conclusion from this information is to choose R.

Exploring the Data :

Now, let's load up and explore the datasets in R to check their Schema and how it's organized so as to determine how to clean and process them in the next Phase.

- Step 1: Load packages

```
# Libraries for Cleaning & Analysis

library(tidyverse)
library(skimr)
library(janitor)
library(lubridate)
library(tidytext)

# Libraries for map generation

library("tidygeocoder")
library("leaflet")
library("mapview")
library("webshot")
```

- Step 2: Import data

```
tripsq1_2019_df <- read_csv("Divvy_Trips_2019_Q1.csv")

tripsq2_2019_df <- read_csv("Divvy_Trips_2019_Q2.csv")

tripsq3_2019_df <- read_csv("Divvy_Trips_2019_Q3.csv")

tripsq4_2019_df <- read_csv("Divvy_Trips_2019_Q4.csv")

tripsq1_2020_df <- read_csv("Divvy_Trips_2020_Q1.csv")
```

- Step 3: Getting to know the data

- **2019 Q1 Data :**

```
head(tripsq1_2019_df)
```

```
## # A tibble: 6 x 12
##    trip_id start_time          end_time            bikeid tripduration
##      <dbl> <dttm>              <dttm>               <dbl>        <dbl>
## 1 21742443 2019-01-01 00:04:37 2019-01-01 00:11:07   2167          390
## 2 21742444 2019-01-01 00:08:13 2019-01-01 00:15:34   4386          441
## 3 21742445 2019-01-01 00:13:23 2019-01-01 00:27:12   1524          829
## 4 21742446 2019-01-01 00:13:45 2019-01-01 00:43:28    252         1783
## 5 21742447 2019-01-01 00:14:52 2019-01-01 00:20:56   1170          364
## 6 21742448 2019-01-01 00:15:33 2019-01-01 00:19:09   2437          216
## # i 7 more variables: from_station_id <dbl>, from_station_name <chr>,
## #   to_station_id <dbl>, to_station_name <chr>, usertype <chr>, gender <chr>,
## #   birthyear <dbl>
```

- **2019 Q2 Data :**

```
glimpse(tripsq2_2019_df)
```

```
## Rows: 1,108,163
## Columns: 12
## $ `01 - Rental Details Rental ID`                     <dbl> 22178529, 22178530,~
## $ `01 - Rental Details Local Start Time`              <dttm> 2019-04-01 00:02:2~
## $ `01 - Rental Details Local End Time`                <dttm> 2019-04-01 00:09:4~
## $ `01 - Rental Details Bike ID`                       <dbl> 6251, 6226, 5649, 4~
## $ `01 - Rental Details Duration In Seconds Uncapped` <dbl> 446, 1048, 252, 357~
## $ `03 - Rental Start Station ID`                      <dbl> 81, 317, 283, 26, 2~
## $ `03 - Rental Start Station Name`                    <chr> "Daley Center Plaza~
## $ `02 - Rental End Station ID`                        <dbl> 56, 59, 174, 133, 1~
## $ `02 - Rental End Station Name`                      <chr> "Desplaines St & Ki~
## $ `User Type`                                         <chr> "Subscriber", "Subs~
## $ `Member Gender`                                     <chr> "Male", "Female", "~
## $ `05 - Member Details Member Birthday Year`          <dbl> 1975, 1984, 1990, 1~
```

- **2019 Q3 Data :**

```
head(tripsq3_2019_df)
```

```
## # A tibble: 6 x 12
##    trip_id start_time          end_time            bikeid tripduration
##      <dbl> <dttm>              <dttm>               <dbl>        <dbl>
## 1 23479388 2019-07-01 00:00:27 2019-07-01 00:20:41   3591         1214
## 2 23479389 2019-07-01 00:01:16 2019-07-01 00:18:44   5353         1048
## 3 23479390 2019-07-01 00:01:48 2019-07-01 00:27:42   6180         1554
## 4 23479391 2019-07-01 00:02:07 2019-07-01 00:27:10   5540         1503
## 5 23479392 2019-07-01 00:02:13 2019-07-01 00:22:26   6014         1213
## 6 23479393 2019-07-01 00:02:21 2019-07-01 00:07:31   4941          310
## # i 7 more variables: from_station_id <dbl>, from_station_name <chr>,
## #   to_station_id <dbl>, to_station_name <chr>, usertype <chr>, gender <chr>,
## #   birthyear <dbl>
```

- **2019 Q4 Data :**

```
glimpse(tripsq4_2019_df)
```

```
## Rows: 704,054
## Columns: 12
## $ trip_id           <dbl> 25223640, 25223641, 25223642, 25223643, 25223644, 25~
## $ start_time        <dttm> 2019-10-01 00:01:39, 2019-10-01 00:02:16, 2019-10-0~
## $ end_time          <dttm> 2019-10-01 00:17:20, 2019-10-01 00:06:34, 2019-10-0~
## $ bikeid            <dbl> 2215, 6328, 3003, 3275, 5294, 1891, 1061, 1274, 6011~
## $ tripduration      <dbl> 940, 258, 850, 2350, 1867, 373, 1072, 1458, 1437, 83~
## $ from_station_id   <dbl> 20, 19, 84, 313, 210, 156, 84, 156, 156, 336, 77, 19~
## $ from_station_name <chr> "Sheffield Ave & Kingsbury St", "Throop (Loomis) St ~
## $ to_station_id     <dbl> 309, 241, 199, 290, 382, 226, 142, 463, 463, 336, 50~
## $ to_station_name   <chr> "Leavitt St & Armitage Ave", "Morgan St & Polk St", ~
```

```
## $ usertype          <chr> "Subscriber", "Subscriber", "Subscriber", "Subscribe~
## $ gender            <chr> "Male", "Male", "Female", "Male", "Male", "Female", ~
## $ birthyear         <dbl> 1987, 1998, 1991, 1990, 1987, 1994, 1991, 1995, 1993~
```

- **2020 Q1 Data :**

```
glimpse(tripsq1_2020_df)
```

```
## Rows: 426,887
## Columns: 13
## $ ride_id            <chr> "EACB19130B0CDA4A", "8FED874C809DC021", "789F3C21E4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-01-21 20:06:59, 2020-01-30 14:22:39, 2020-01-~
## $ ended_at           <dttm> 2020-01-21 20:14:30, 2020-01-30 14:26:22, 2020-01-~
## $ start_station_name <chr> "Western Ave & Leland Ave", "Clark St & Montrose Av~
## $ start_station_id   <dbl> 239, 234, 296, 51, 66, 212, 96, 96, 212, 38, 117, 1~
## $ end_station_name   <chr> "Clark St & Leland Ave", "Southport Ave & Irving Pa~
## $ end_station_id     <dbl> 326, 318, 117, 24, 212, 96, 212, 212, 96, 100, 632,~
## $ start_lat          <dbl> 41.9665, 41.9616, 41.9401, 41.8846, 41.8856, 41.889~
## $ start_lng          <dbl> -87.6884, -87.6660, -87.6455, -87.6319, -87.6418, -~
## $ end_lat            <dbl> 41.9671, 41.9542, 41.9402, 41.8918, 41.8899, 41.884~
## $ end_lng            <dbl> -87.6674, -87.6644, -87.6530, -87.6206, -87.6343, -~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

Observations :

After exploring these 5 datasets, I got to know that -

- Dataset "tripsq2_2019_df" has -
  - Different variable headers than the rest of the Datasets of the year 2019 (Eg; '01 - Rental Details Rental ID' in "tripsq2_2019_df" and 'trip_id' in "tripsq3_2019_df")
- Dataset "tripsq1_2020_df" has -
  - Different variable headers than the Datasets of the year 2019 (Eg; 'ride_id' in "tripsq1_2020_df" and 'trip_id' in "tripsq3_2019_df")
  - User type observations are categorized differently from that of the year 2019. (Eg; 'member'/'casual' in "tripsq1_2020_df" and 'Subscriber'/'Customer' in "tripsq3_2019_df")
  - Unique Id ie; the ride_id variable has 'character' datatype whereas trip_id variable of the datasets in the year 2019 has 'double' as the data type.
- The most relevant variables in these datasets for our business task are -
  - "trip_id", "bike_id", "start_time", "end_time", "start_station_id", "start_station_name" "end_station_id", "end_station_name", "gender", "birth_year", "user_type"

Metadata of the relevant variables/column headers :

- trip_id – Unique identifier for each bike trip.
- bike_id – Unique identifier for the bike used during the trip.
- start_time – Timestamp indicating when the trip started.
- end_time – Timestamp indicating when the trip ended.
- start_station_id – ID of the station where the trip began.

- start_station_name – Name of the station where the trip began.
- end_station_id – ID of the station where the trip ended.
- end_station_name – Name of the station where the trip ended.
- gender – Gender of the rider (e.g., Male, Female, Other).
- birth_year – Year of birth of the rider (used to calculate age).
- user_type – Type of user: member or casual rider.

Now that we have stored and explored the data, let's go ahead Clean and Process the data for Analysis.

3. PROCESS PHASE (Deliverable = Documentation of any cleaning or manipulation of data)

I have to make the datatypes and variable names consistent across all datasets, and then merge the datasets together for cleaning the textual errors, missing values, and other data entry errors - with higher efficiency.

Pre - Cleaning :

1) First, select the necessary variables which are relevant to the Business task.

Here, I will be selecting these variables from our initial observations of the datasets :

- "trip_id", "bike_id", "start_time", "end_time", "start_station_id", "start_station_name" "end_station_id", "end_station_name", "gender", "birth_year", "user_type"

2) Second, using the Observations from the Prepare phase, make every selected variables/columns in each dataset consistent with each other :

- Rename the selected columns/variables in all the datasets to make them consistent to each other. (Eg; Rename "ride_id" or "01 - Rental Details Rental ID" to "trip_id")
- Make the data type of Unique Ids (Unique Id = "trip_id"/"ride_id"/"01 - Rental Details Rental ID") consistent by changing them into 'character' type.
- In the User type variable/column, replace "Subscriber" with "member" and "Customer" with "casual"

- Change the data type of the User type variable into 'factor' type for easy categorization.

Let's create a function to do all these manipulations for the datasets 'tripsq4_2019_df', 'tripsq3_2019_df', 'tripsq1_2019_df'. We can do the same manipulations for 'tripsq2_2019_df' and 'tripsq1_2020_df' separately.

```
# Function to transform dataframes 'tripsq1_2019_df', 'tripsq3_2019_df', 'tripsq4_2019_df'.

pre_process_2019_q1q3q4 <- function(df) {
  df %>%
    mutate(
      trip_id = as.character(trip_id),
      user_type = as.factor(recode(usertype,
                                   "Subscriber" = "member",
                                   "Customer" = "casual")),
      gender = as.factor(gender)
      ) %>%
    rename(
```

```
      start_station_id = from_station_id,
      start_station_name = from_station_name,
      end_station_id = to_station_id,
      end_station_name = to_station_name,
      birth_year = birthyear,
      bike_id = bikeid
    ) %>%
    select(
      trip_id, bike_id, start_time, end_time,
      start_station_id, start_station_name,
      end_station_id, end_station_name,
      gender, birth_year, user_type
      )
}
```

Use the function to transform 2019's q1,q3,q4 data :

- ***(Manipulation_1) Transform the dataframes 'tripsq1_2019_df', 'tripsq3_2019_df', 'tripsq4_2019_df' to have data consistency with each other and with all other dataframes.***

```
# Transform the dataframes 'tripsq1_2019_df', 'tripsq3_2019_df', 'tripsq4_2019_df' to have data consist

tripsq1_2019_df_1 <- pre_process_2019_q1q3q4(tripsq1_2019_df)
tripsq3_2019_df_1 <- pre_process_2019_q1q3q4(tripsq3_2019_df)
tripsq4_2019_df_1 <- pre_process_2019_q1q3q4(tripsq4_2019_df)
```

Now, transform 2019's q2 -

- ***(Manipulation_2) Transform the dataframe 'tripsq2_2019_df' to have data consistency with all other dataframes.***

```
# Transform the dataframe 'tripsq2_2019_df' to have data consistency with all other dataframes.

tripsq2_2019_df_1 <- tripsq2_2019_df %>%
  mutate(trip_id = as.character(`01 - Rental Details Rental ID`),
         user_type = as.factor(recode(`User Type`, "Subscriber" = "member", "Customer" = "casual")),
         gender = as.factor(`Member Gender`)) %>%
  rename(start_time = `01 - Rental Details Local Start Time`,
         end_time = `01 - Rental Details Local End Time`,
         start_station_id = `03 - Rental Start Station ID`,
         start_station_name = `03 - Rental Start Station Name`,
         end_station_id = `02 - Rental End Station ID`,
         end_station_name = `02 - Rental End Station Name`,
         birth_year = `05 - Member Details Member Birthday Year`,
         bike_id = `01 - Rental Details Bike ID`) %>%
  select(trip_id, bike_id, start_time, end_time,
         start_station_id, start_station_name,
         end_station_id, end_station_name,
         gender, birth_year, user_type)
```

and 2020 q1 data -

- ***(Manipulation_3) Transform the dataframe 'tripsq1_2020_df' to make the data consistent with all other dataframes.***

```
# Transform the dataframe 'tripsq1_2020_df' to make the data consistent with all other dataframes.

tripsq1_2020_df_1 <- tripsq1_2020_df %>%
  mutate(user_type = as.factor(member_casual)) %>%
  rename(trip_id = ride_id,
         start_time = started_at,
         end_time = ended_at) %>%
  select(trip_id, start_time, end_time,
         start_station_id, start_station_name,
         end_station_id, end_station_name, user_type)
```

Let's view the modified datasets -

```
glimpse(tripsq1_2019_df_1)
```

```
## Rows: 365,069
## Columns: 11
## $ trip_id            <chr> "21742443", "21742444", "21742445", "21742446", "21~
## $ bike_id            <dbl> 2167, 4386, 1524, 252, 1170, 2437, 2708, 2796, 6205~
## $ start_time         <dttm> 2019-01-01 00:04:37, 2019-01-01 00:08:13, 2019-01-~
## $ end_time           <dttm> 2019-01-01 00:11:07, 2019-01-01 00:15:34, 2019-01-~
## $ start_station_id   <dbl> 199, 44, 15, 123, 173, 98, 98, 211, 150, 268, 299, ~
## $ start_station_name <chr> "Wabash Ave & Grand Ave", "State St & Randolph St",~
## $ end_station_id     <dbl> 84, 624, 644, 176, 35, 49, 49, 142, 148, 141, 295, ~
## $ end_station_name   <chr> "Milwaukee Ave & Grand Ave", "Dearborn St & Van Bur~
## $ gender             <fct> Male, Female, Female, Male, Male, Female, Male, Mal~
## $ birth_year         <dbl> 1989, 1990, 1994, 1993, 1994, 1983, 1984, 1990, 199~
## $ user_type          <fct> member, member, member, member, member, member, mem~
```

```
glimpse(tripsq2_2019_df_1)
```

```
## Rows: 1,108,163
## Columns: 11
## $ trip_id            <chr> "22178529", "22178530", "22178531", "22178532", "22~
## $ bike_id            <dbl> 6251, 6226, 5649, 4151, 3270, 3123, 6418, 4513, 328~
## $ start_time         <dttm> 2019-04-01 00:02:22, 2019-04-01 00:03:02, 2019-04-~
## $ end_time           <dttm> 2019-04-01 00:09:48, 2019-04-01 00:20:30, 2019-04-~
## $ start_station_id   <dbl> 81, 317, 283, 26, 202, 420, 503, 260, 211, 211, 304~
## $ start_station_name <chr> "Daley Center Plaza", "Wood St & Taylor St", "LaSal~
## $ end_station_id     <dbl> 56, 59, 174, 133, 129, 426, 500, 499, 211, 211, 232~
## $ end_station_name   <chr> "Desplaines St & Kinzie St", "Wabash Ave & Roosevel~
## $ gender             <fct> Male, Female, Male, Male, Male, Male, Male, Male, N~
## $ birth_year         <dbl> 1975, 1984, 1990, 1993, 1992, 1999, 1969, 1991, NA,~
## $ user_type          <fct> member, member, member, member, member, member, mem~
```

```
glimpse(tripsq3_2019_df_1)
```

```
## Rows: 1,640,718
```

```
## Columns: 11
## $ trip_id            <chr> "23479388", "23479389", "23479390", "23479391", "23~
## $ bike_id            <dbl> 3591, 5353, 6180, 5540, 6014, 4941, 3770, 5442, 295~
## $ start_time         <dttm> 2019-07-01 00:00:27, 2019-07-01 00:01:16, 2019-07-~
## $ end_time           <dttm> 2019-07-01 00:20:41, 2019-07-01 00:18:44, 2019-07-~
## $ start_station_id   <dbl> 117, 381, 313, 313, 168, 300, 168, 313, 43, 43, 511~
## $ start_station_name <chr> "Wilton Ave & Belmont Ave", "Western Ave & Monroe S~
## $ end_station_id     <dbl> 497, 203, 144, 144, 62, 232, 62, 144, 195, 195, 84,~
## $ end_station_name   <chr> "Kimball Ave & Belmont Ave", "Western Ave & 21st St~
## $ gender             <fct> Male, NA, NA, NA, NA, Male, NA, NA, NA, NA, NA, NA,~
## $ birth_year         <dbl> 1992, NA, NA, NA, NA, 1990, NA, NA, NA, NA, NA, NA,~
## $ user_type          <fct> member, casual, casual, casual, casual, member, cas~
```

```
glimpse(tripsq4_2019_df_1)
```

```
## Rows: 704,054
## Columns: 11
## $ trip_id            <chr> "25223640", "25223641", "25223642", "25223643", "25~
## $ bike_id            <dbl> 2215, 6328, 3003, 3275, 5294, 1891, 1061, 1274, 601~
## $ start_time         <dttm> 2019-10-01 00:01:39, 2019-10-01 00:02:16, 2019-10-~
## $ end_time           <dttm> 2019-10-01 00:17:20, 2019-10-01 00:06:34, 2019-10-~
## $ start_station_id   <dbl> 20, 19, 84, 313, 210, 156, 84, 156, 156, 336, 77, 1~
## $ start_station_name <chr> "Sheffield Ave & Kingsbury St", "Throop (Loomis) St~
## $ end_station_id     <dbl> 309, 241, 199, 290, 382, 226, 142, 463, 463, 336, 5~
## $ end_station_name   <chr> "Leavitt St & Armitage Ave", "Morgan St & Polk St",~
## $ gender             <fct> Male, Male, Female, Male, Male, Female, Female, Mal~
## $ birth_year         <dbl> 1987, 1998, 1991, 1990, 1987, 1994, 1991, 1995, 199~
## $ user_type          <fct> member, member, member, member, member, member, mem~
```

```
glimpse(tripsq1_2020_df_1)
```

```
## Rows: 426,887
## Columns: 8
## $ trip_id            <chr> "EACB19130B0CDA4A", "8FED874C809DC021", "789F3C21E4~
## $ start_time         <dttm> 2020-01-21 20:06:59, 2020-01-30 14:22:39, 2020-01-~
## $ end_time           <dttm> 2020-01-21 20:14:30, 2020-01-30 14:26:22, 2020-01-~
## $ start_station_id   <dbl> 239, 234, 296, 51, 66, 212, 96, 96, 212, 38, 117, 1~
## $ start_station_name <chr> "Western Ave & Leland Ave", "Clark St & Montrose Av~
## $ end_station_id     <dbl> 326, 318, 117, 24, 212, 96, 212, 212, 96, 100, 632,~
## $ end_station_name   <chr> "Clark St & Leland Ave", "Southport Ave & Irving Pa~
## $ user_type          <fct> member, member, member, member, member, member, mem~
```

We now have consistent variables names and datatypes across all datasets, enabling us to merge them seamlessly.

3) Merge all the datasets into one.

- ***(Manipulation_4) Merge all the data frames 'tripsq1_2019_df', 'tripsq2_2019_df', 'tripsq3_2019_df', 'tripsq4_2019_df', 'tripsq1_2020_df', & store as 'all_trips_19_20'***

```r
# Merge all the data frames 'tripsq1_2019_df', 'tripsq2_2019_df', 'tripsq3_2019_df', 'tripsq4_2019_df',

all_trips_19_20 <- bind_rows(
  tripsq1_2019_df_1,
  tripsq2_2019_df_1,
  tripsq3_2019_df_1,
  tripsq4_2019_df_1,
  tripsq1_2020_df_1
)

# Structure of dataframe 'all_trips_19_20'

str(all_trips_19_20)
```

```
## tibble [4,244,891 x 11] (S3: tbl_df/tbl/data.frame)
##  $ trip_id           : chr [1:4244891] "21742443" "21742444" "21742445" "21742446" ...
##  $ bike_id           : num [1:4244891] 2167 4386 1524 252 1170 ...
##  $ start_time        : POSIXct[1:4244891], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ end_time          : POSIXct[1:4244891], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ start_station_id  : num [1:4244891] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:4244891] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave
##  $ end_station_id    : num [1:4244891] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:4244891] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)"
##  $ gender            : Factor w/ 2 levels "Female","Male": 2 1 1 2 2 1 2 2 2 2 ...
##  $ birth_year        : num [1:4244891] 1989 1990 1994 1993 1994 ...
##  $ user_type         : Factor w/ 2 levels "casual","member": 2 2 2 2 2 2 2 2 2 2 ...
```

There are 4.24 Million rows in this merged Table.

2. Cleaning & Processing the Data :

1) Checking for any duplicate entries :

```r
# Checking for duplicate rows with same trip_id

all_trips_19_20 %>%
  group_by(trip_id) %>%
  summarise(count = n(), .groups = "drop") %>%
  filter(count>1) %>%
  glimpse()
```

```
## Rows: 0
## Columns: 2
## $ trip_id <chr>
## $ count   <int>
```

When we checked the Unique_Id (trip_id), there are no duplicates. But, we should group all the other columns and check if there are any duplicates this way too :

```
# Checking for duplicate rows with same bike_id, start_time, end_time, start_station_id, start_station_
all_trips_19_20 %>%
  group_by(bike_id, start_time, end_time, start_station_id, start_station_name, end_station_id, end_sta
  summarise(count = n(), .groups = "drop") %>%
  filter(count>1) %>%
  str()
```

```
## tibble [164 x 11] (S3: tbl_df/tbl/data.frame)
##  $ bike_id          : num [1:164] 22 73 239 369 375 398 419 457 488 488 ...
##  $ start_time       : POSIXct[1:164], format: "2019-07-11 17:03:39" "2019-07-11 17:06:55" ...
##  $ end_time         : POSIXct[1:164], format: "2019-07-11 17:07:28" "2019-07-11 17:24:04" ...
##  $ start_station_id : num [1:164] 342 176 313 447 51 108 48 184 252 267 ...
##  $ start_station_name: chr [1:164] "Wolcott Ave & Polk St" "Clark St & Elm St" "Lakeview Ave & Fulle
##  $ end_station_id   : num [1:164] 122 313 268 449 643 135 130 202 267 267 ...
##  $ end_station_name : chr [1:164] "Ogden Ave & Congress Pkwy" "Lakeview Ave & Fullerton Pkwy" "Lake
##  $ user_type        : Factor w/ 2 levels "casual","member": 2 2 2 2 2 2 1 2 2 1 ...
##  $ gender           : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 1 1 2 ...
##  $ birth_year       : num [1:164] 1951 1967 1992 1995 1989 ...
##  $ count            : int [1:164] 2 2 2 2 2 2 2 2 2 2 ...
```

There are duplicates in our data as seen above.

- ***(Manipulation_5) Remove duplicate rows from 'all_trips_19_20', and store the new dataframe as 'all_trips_19_20_1' .***

```
# Selecting distinct rows and storing them into a new dataframe 'all_trips_19_20_1'

all_trips_19_20_1 <- all_trips_19_20 %>%
  distinct(
    bike_id, start_time, end_time, start_station_id, start_station_name,
    end_station_id, end_station_name, user_type, gender, birth_year,
    .keep_all = TRUE
  )

# No. of rows deleted.

nrow(all_trips_19_20) - nrow(all_trips_19_20_1)
```

```
## [1] 170
```

- Cleaned all the duplicate rows in the Table.

2) Checking for Data inconsistencies :

a. **Check if the end date is smaller than start date -**

```
# Check by filtering

all_trips_19_20_1 %>%
  filter(end_time < start_time) %>%
  glimpse()
```

```
## Rows: 130
## Columns: 11
## $ trip_id            <chr> "25625830", "25625836", "25625838", "25625839", "25~
## $ bike_id            <dbl> 4141, 6329, 964, 2214, 4179, 2920, 3338, 2142, 5877~
## $ start_time         <dttm> 2019-11-03 01:43:21, 2019-11-03 01:46:01, 2019-11-~
## $ end_time           <dttm> 2019-11-03 01:09:56, 2019-11-03 01:10:44, 2019-11-~
## $ start_station_id   <dbl> 632, 373, 229, 131, 298, 460, 177, 177, 177, 484, 1~
## $ start_station_name <chr> "Clark St & Newport St", "Kedzie Ave & Chicago Ave"~
## $ end_station_id     <dbl> 133, 498, 87, 131, 258, 238, 327, 327, 327, 484, 32~
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "California Ave & Fletc~
## $ gender             <fct> Male, NA, Female, Male, Male, Male, NA, NA, NA, NA,~
## $ birth_year         <dbl> 1995, NA, 1987, 1996, 1975, 1992, NA, NA, NA, NA, 1~
## $ user_type          <fct> casual, casual, member, member, member, member, cas~
```

There are 130 rows with start and end date-time inconsistencies. Let's swap the date-times.

- ***(Manipulation_6) Swap the start & end date-times in 'all_trips_19_20_1', and store the new dataframe as 'all_trips_19_20_2' .***

```
# Swapping the inconsistent start and end date-times, and saving the table to a new dataframe 'all_trip

all_trips_19_20_2 <- all_trips_19_20_1 %>%
  mutate(
    temp = if_else(end_time < start_time, start_time, end_time),
    start_time = if_else(end_time < start_time, end_time, start_time),
    end_time = temp
  ) %>%
  select(-temp)

# Verify the success of manipulation

all_trips_19_20_2 %>%
  filter(end_time < start_time) %>%
  glimpse()
```

```
## Rows: 0
## Columns: 11
## $ trip_id            <chr>
## $ bike_id            <dbl>
## $ start_time         <dttm>
## $ end_time           <dttm>
## $ start_station_id   <dbl>
## $ start_station_name <chr>
## $ end_station_id     <dbl>
## $ end_station_name   <chr>
## $ gender             <fct>
## $ birth_year         <dbl>
## $ user_type          <fct>
```

b. **Check if there are start date-times the same as end date-times; and Check if there are end_time, start_time combinations with duration (end_time - start_time) above 24 hours -**

```
# Check if there are start date-times the same as end date-times

all_trips_19_20_2 %>%
  filter(start_time == end_time) %>%
  glimpse()
```

```
## Rows: 88
## Columns: 11
## $ trip_id            <chr> "23EF1DCC9FCA40BA", "86163D9676BBBE62", "836931C569~
## $ bike_id            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_time         <dttm> 2020-02-28 11:34:40, 2020-02-26 14:41:16, 2020-02-~
## $ end_time           <dttm> 2020-02-28 11:34:40, 2020-02-26 14:41:16, 2020-02-~
## $ start_station_id   <dbl> 675, 675, 675, 675, 675, 675, 675, 675, 675, 675, 6~
## $ start_station_name <chr> "HQ QR", "HQ QR", "HQ QR", "HQ QR", "HQ QR", "HQ QR~
## $ end_station_id     <dbl> 675, 675, 675, 675, 675, 675, 675, 675, 675, 675, 6~
## $ end_station_name   <chr> "HQ QR", "HQ QR", "HQ QR", "HQ QR", "HQ QR", "HQ QR~
## $ gender             <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ birth_year         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ user_type          <fct> casual, casual, casual, casual, casual, casual, cas~
```

```
# Check if there are end_time, start_time combinations with duration (end_time - start_time) above 24 h

all_trips_19_20_2 %>%
  filter(as.numeric(end_time - start_time, units ="secs") > 24*60*60) %>%
  arrange(end_time - start_time) %>%
  glimpse()
```

```
## Rows: 2,139
## Columns: 11
## $ trip_id            <chr> "24864528", "22317802", "439BCB941291A940", "246815~
## $ bike_id            <dbl> 499, 1205, NA, 1612, 5889, NA, 2064, 2260, 2206, 64~
## $ start_time         <dttm> 2019-09-09 18:15:38, 2019-04-15 13:37:53, 2020-01-~
## $ end_time           <dttm> 2019-09-10 18:15:39, 2019-04-16 13:38:32, 2020-01-~
## $ start_station_id   <dbl> 115, 177, 622, 534, 35, 145, 195, 229, 451, 76, 447~
## $ start_station_name <chr> "Sheffield Ave & Wellington Ave", "Theater on the L~
## $ end_station_id     <dbl> 664, 329, 622, 201, 251, 673, 264, 304, 625, 7, 242~
## $ end_station_name   <chr> "Leavitt St & Belmont Ave (*)", "Lake Shore Dr & Di~
## $ gender             <fct> Female, NA, NA, Male, NA, NA, NA, NA, NA, NA, NA, N~
## $ birth_year         <dbl> 1994, NA, NA, 1985, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ user_type          <fct> casual, casual, member, casual, casual, member, cas~
```

Let's remove all these outliers

- ***(Manipulation_7) Remove the rows with Trip duration (end_time - start_time) as 0 and Trip duration above 24 hrs in 'all_trips_19_20_2', and store the new dataframe as 'all_trips_19_20_3'.***

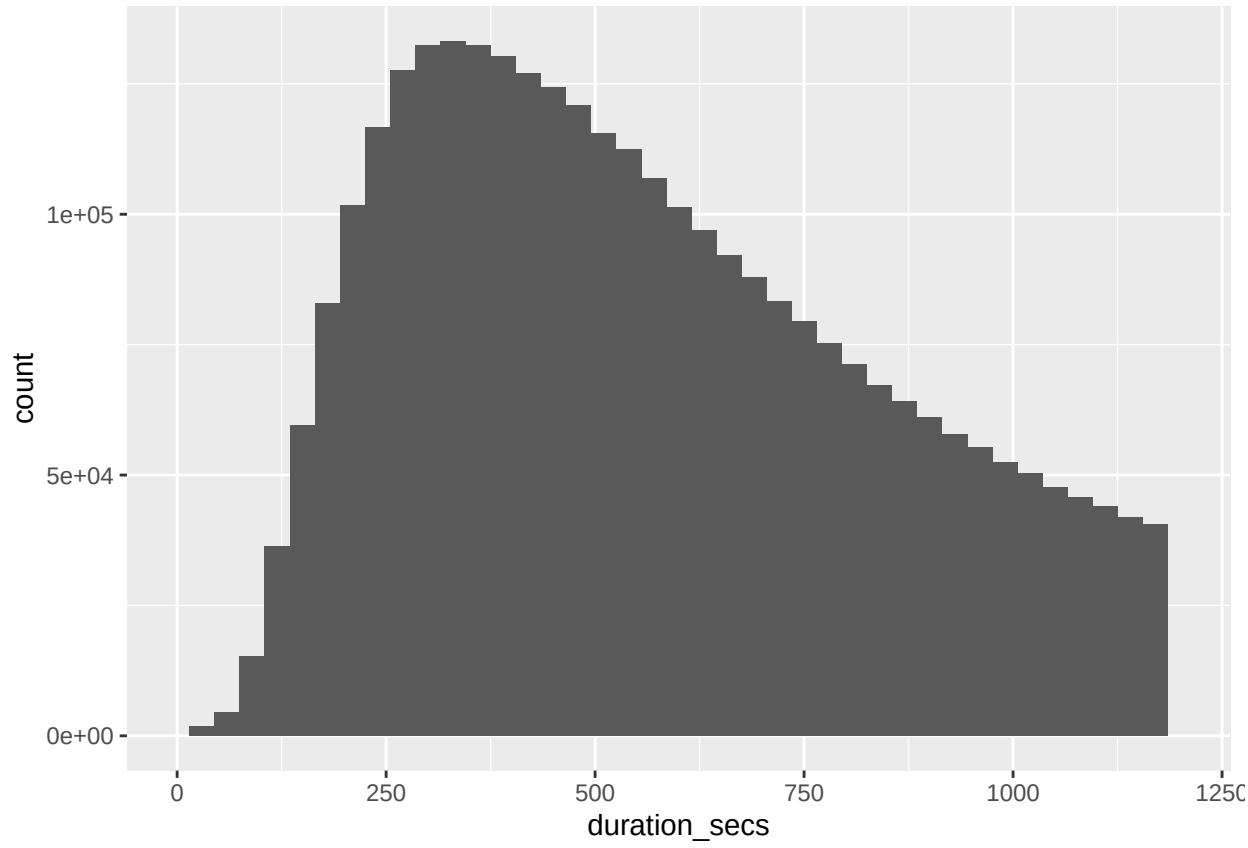
```
# Remove the rows with Trip duration (end_time - start_time) as 0 and Trip duration above 24 hrs and st

all_trips_19_20_3 <- all_trips_19_20_2  %>%
  filter(start_time != end_time &
```

```
            as.numeric(end_time - start_time, "secs") <= 24*60*60)


# Verify the removal of rows (No. of rows removed)

nrow(all_trips_19_20_2) - nrow(all_trips_19_20_3)
```

```
## [1] 2227
```

c) **Look at the duration distribution to check the range of noise in the duration (Eg; 1 second of ride duration - This data is Noise (As this data point doesn't make any sense)) -**

```
# Histogram of Number of Trips vs Trip Duration (sec)

all_trips_19_20_3 %>%
  mutate(duration_secs = as.numeric(end_time - start_time, "secs")) %>%
  ggplot(aes(x = duration_secs)) +
  geom_histogram(binwidth = 30) +
  xlim(0, 1200)
```



We can see from the Histogram that in the first 60 seconds ie; 2 steps in the graph (1 step = 30 sec), there is no significant Trip count - which means that the first 60 seconds is Noise in the data.

So I will be removing the data in which the duration is below 60 sec.

- ***(Manipulation_8) Remove the rows with Trip duration (end_time - start_time) below 60 seconds and store the new dataframe as 'all_trips_19_20_4'.***

```
# Remove the rows with Trip duration (end_time - start_time) below 60 seconds and store the new datafra

all_trips_19_20_4 <- all_trips_19_20_3 %>%
  filter(as.numeric(end_time - start_time, "secs") >= 60)

# Number of rows removed :

nrow(all_trips_19_20_3) - nrow(all_trips_19_20_4)
```

```
## [1] 7543
```

```
# Check if there is any rows with Trip duration below 60 seconds

all_trips_19_20_4 %>%
  mutate(duration_sec = as.numeric(end_time - start_time, "secs")) %>%
  filter(duration_sec < 60) %>%
  select(trip_id, start_time, end_time, duration_sec) %>%
  arrange(duration_sec)
```

```
## # A tibble: 0 x 4
## # i 4 variables: trip_id <chr>, start_time <dttm>, end_time <dttm>,
## #   duration_sec <dbl>
```

- Cleaned all Data inconsistencies in the table.

3) Checking for any data range errors :

a) **Check for data range errors in the 'birth_year' column**

```
# Check if there are any riders with age below 16 or above 90 years

all_trips_19_20_4 %>%
  filter(as.numeric(year(start_time) - birth_year) > 90 | as.numeric(year(start_time) - birth_year) < 1
  arrange(desc(birth_year)) %>%
  glimpse()
```

```
## Rows: 1,037
## Columns: 11
## $ trip_id            <chr> "22463474", "22483110", "22634065", "23679951", "23~
## $ bike_id            <dbl> 6225, 6391, 2076, 3458, 6211, 1797, 1081, 5286, 438~
## $ start_time         <dttm> 2019-04-26 17:20:58, 2019-04-29 18:44:12, 2019-05-~
## $ end_time           <dttm> 2019-04-26 19:21:07, 2019-04-29 19:59:27, 2019-05-~
## $ start_station_id   <dbl> 204, 421, 204, 236, 341, 97, 309, 309, 464, 464, 46~
## $ start_station_name <chr> "Prairie Ave & Garfield Blvd", "MLK Jr Dr & 56th St~
## $ end_station_id     <dbl> 421, 421, 421, 341, 291, 100, 260, 260, 464, 464, 2~
## $ end_station_name   <chr> "MLK Jr Dr & 56th St (*)", "MLK Jr Dr & 56th St (*)~
## $ gender             <fct> Female, Female, Female, NA, NA, NA, Male, Male, Mal~
## $ birth_year         <dbl> 2014, 2014, 2014, 1928, 1928, 1928, 1925, 1925, 192~
## $ user_type          <fct> member, member, member, casual, casual, casual, cas~
```

We can see that there are 1037 riders with age below 16, and riders with age above 90 - which is an Anomaly. Let's remove those anomalies.

- ***(Manipulation_9) Remove all the rows with birth_years corresponding to ages below 16 and above 90 from 'all_trips_19_20_4', and store the new dataframe as 'all_trips_19_20_5' .***

```
# Remove all the rows with birth_years corresponding to ages below 16 and above 90 from 'all_trips_19_2

all_trips_19_20_5 <- all_trips_19_20_4 %>%
  filter(is.na(birth_year) | !((year(start_time) - birth_year) > 90 | as.numeric(year(start_time) - bir

# Verify the Operation

all_trips_19_20_5 %>%
  filter(as.numeric(year(start_time) - birth_year) > 90 | as.numeric(year(start_time) - birth_year) < 1
  arrange(desc(birth_year)) %>%
  glimpse()
```

```
## Rows: 0
## Columns: 11
## $ trip_id            <chr>
## $ bike_id            <dbl>
## $ start_time         <dttm>
## $ end_time           <dttm>
## $ start_station_id   <dbl>
## $ start_station_name <chr>
## $ end_station_id     <dbl>
## $ end_station_name   <chr>
## $ gender             <fct>
## $ birth_year         <dbl>
## $ user_type          <fct>
```

```
# Find how many rows were removed

nrow(all_trips_19_20_4) - nrow(all_trips_19_20_5)
```

```
## [1] 1037
```

- Cleaned all Data range errors in the table.

4) Checking for missing values / textual errors :

a) **Check for missing values the columns of 'factor' datatype (gender, user_type) -**

```
all_trips_19_20_5 %>%
  distinct(gender)
```

```
## # A tibble: 3 x 1
##   gender
```

```
##   <fct>
## 1 Male
## 2 Female
## 3 <NA>
```

```
all_trips_19_20_5 %>%
  distinct(user_type)
```

```
## # A tibble: 2 x 1
##   user_type
##   <fct>
## 1 member
## 2 casual
```

- 'user_type' has it's necessary distinct categories "member", "casual" - which means no textual errors or missing values in that column.
- 'gender' field has 3 distinct categories. One category is that of missing value ie; 'NA'.

Investigating into the missing values in 'gender' column :

```
all_trips_19_20_5 %>%
  filter(is.na(gender)) %>%
  glimpse()
```

```
## Rows: 977,102
## Columns: 11
## $ trip_id            <chr> "21742463", "21742465", "21742494", "21742498", "21~
## $ bike_id            <dbl> 3914, 3355, 2517, 374, 1776, 341, 4507, 628, 4333, ~
## $ start_time         <dttm> 2019-01-01 00:29:19, 2019-01-01 00:29:28, 2019-01-~
## $ end_time           <dttm> 2019-01-01 01:08:12, 2019-01-01 01:07:49, 2019-01-~
## $ start_station_id   <dbl> 35, 35, 290, 367, 367, 316, 316, 260, 35, 35, 318, ~
## $ start_station_name <chr> "Streeter Dr & Grand Ave", "Streeter Dr & Grand Ave~
## $ end_station_id     <dbl> 39, 39, 476, 9, 9, 457, 457, 240, 37, 37, 229, 282,~
## $ end_station_name   <chr> "Wabash Ave & Adams St", "Wabash Ave & Adams St", "~
## $ gender             <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ birth_year         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ user_type          <fct> casual, casual, casual, casual, casual, casual, cas~
```

There are approx. 1 Million rows of missing values out of approx. 4 million total rows.

b) **And, 'birth_year' also seems to have missing values. Let's check that too.**

```
all_trips_19_20_5 %>%
  filter(is.na(birth_year)) %>%
  glimpse()
```

```
## Rows: 956,712
## Columns: 11
## $ trip_id            <chr> "21742463", "21742465", "21742494", "21742498", "21~
## $ bike_id            <dbl> 3914, 3355, 2517, 374, 1776, 341, 4507, 628, 4333, ~
```

```
## $ start_time         <dttm> 2019-01-01 00:29:19, 2019-01-01 00:29:28, 2019-01-~
## $ end_time           <dttm> 2019-01-01 01:08:12, 2019-01-01 01:07:49, 2019-01-~
## $ start_station_id   <dbl> 35, 35, 290, 367, 367, 316, 316, 260, 35, 35, 318, ~
## $ start_station_name <chr> "Streeter Dr & Grand Ave", "Streeter Dr & Grand Ave~
## $ end_station_id     <dbl> 39, 39, 476, 9, 9, 457, 457, 240, 37, 37, 229, 282,~
## $ end_station_name   <chr> "Wabash Ave & Adams St", "Wabash Ave & Adams St", "~
## $ gender             <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ birth_year         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ user_type          <fct> casual, casual, casual, casual, casual, casual, cas~
```

In 'birth_year' column too, there are approx. 1 Million rows of missing values.

I couldn't find any accurate data to fill the missing values in both the columns. So, if I removed all the rows having missing values, I could remove more than 25% of invaluable data. So, let's keep it as is. But I should have this information in the back of my mind while doing analysis.

c) **Check the bike_id column for missing values :**

```
all_trips_19_20_5 %>%
  filter(is.na(bike_id)) %>%
  glimpse()
```

```
## Rows: 418,911
## Columns: 11
## $ trip_id            <chr> "EACB19130B0CDA4A", "8FED874C809DC021", "789F3C21E4~
## $ bike_id            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_time         <dttm> 2020-01-21 20:06:59, 2020-01-30 14:22:39, 2020-01-~
## $ end_time           <dttm> 2020-01-21 20:14:30, 2020-01-30 14:26:22, 2020-01-~
## $ start_station_id   <dbl> 239, 234, 296, 51, 66, 212, 96, 96, 212, 38, 117, 1~
## $ start_station_name <chr> "Western Ave & Leland Ave", "Clark St & Montrose Av~
## $ end_station_id     <dbl> 326, 318, 117, 24, 212, 96, 212, 212, 96, 100, 632,~
## $ end_station_name   <chr> "Clark St & Leland Ave", "Southport Ave & Irving Pa~
## $ gender             <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ birth_year         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ user_type          <fct> member, member, member, member, member, member, mem~
```

There are approx. 0.4 Million rows with missing values in bike_id. But there are approx. 3.5 million or more data with accurate data. As I couldn't find consistent data to fill these missing fields, I will be keeping it as is. I will be keeping this information in the back of my mind while doing analysis.

d) **Check if any other columns have missing values :**

```
all_trips_19_20_5 %>%
  filter(is.na(trip_id) |
           is.na(start_time) |
           is.na(end_time) |
           is.na(start_station_id) |
           is.na(start_station_name) |
           is.na(end_station_id) |
           is.na(end_station_name) |
           is.na(user_type)) %>%
  glimpse()
```

```
## Rows: 0
## Columns: 11
## $ trip_id            <chr>
## $ bike_id            <dbl>
## $ start_time         <dttm>
## $ end_time           <dttm>
## $ start_station_id   <dbl>
## $ start_station_name <chr>
## $ end_station_id     <dbl>
## $ end_station_name   <chr>
## $ gender             <fct>
## $ birth_year         <dbl>
## $ user_type          <fct>
```

There is no missing values in the other columns * Cleaned all the missing values in the Table.

That's it for Organizing and Cleaning the data.

Changelog :

Click here to view the Changelog

Summary of the Process phase -

- Pre-cleaned the data by :
  - Standardizing the data
  - Organizing the data by Merging
- Cleaned all the :
  - Duplicate entries
  - Data inconsistencies
  - Data range errors
  - Missing values and textual errors if any

4. ANALYZE PHASE (Deliverable = Summary of Analysis)

Calculation :

Now, let's calculate all the necessary data that we need :

a) trip_duration

- ***(Manipulation_10) Calculate Trip duration as the variable trip_duration. Save the new dataframe as all_trips_19_20_6.***

```
# Calculate Trip duration as the variable trip_duration. Save the new dataframe as all_trips_19_20_6

all_trips_19_20_6 <- all_trips_19_20_5 %>%
  mutate(trip_duration = as.numeric(end_time - start_time, "secs")) %>%
  arrange(trip_duration)

# View 'all_trips_19_20_6'

str(all_trips_19_20_6)
```

```
## tibble [4,233,914 x 12] (S3: tbl_df/tbl/data.frame)
##  $ trip_id           : chr [1:4233914] "B9ED9D8CE75F542B" "7DC888EB586ED128" "681409CD394F390A" "C06
##  $ bike_id           : num [1:4233914] NA NA NA NA NA NA NA NA NA NA ...
##  $ start_time        : POSIXct[1:4233914], format: "2020-01-26 01:51:36" "2020-01-07 15:56:04" ...
##  $ end_time          : POSIXct[1:4233914], format: "2020-01-26 01:52:36" "2020-01-07 15:57:04" ...
##  $ start_station_id  : num [1:4233914] 41 174 174 291 91 14 198 164 197 238 ...
##  $ start_station_name: chr [1:4233914] "Federal St & Polk St" "Canal St & Madison St" "Canal St & Ma
##  $ end_station_id    : num [1:4233914] 394 77 77 291 77 14 198 164 197 238 ...
##  $ end_station_name  : chr [1:4233914] "Clark St & 9th St (AMLI)" "Clinton St & Madison St" "Clinton
##  $ gender            : Factor w/ 2 levels "Female","Male": NA NA NA NA NA NA NA NA NA NA ...
##  $ birth_year        : num [1:4233914] NA NA NA NA NA NA NA NA NA NA ...
##  $ user_type         : Factor w/ 2 levels "casual","member": 2 2 2 2 2 2 2 2 2 2 ...
##  $ trip_duration     : num [1:4233914] 60 60 60 60 60 60 60 60 60 60 ...
```

b) week_day

- ***(Manipulation_11) Calculate Day of the week as the variable week_day. Save the new dataframe as all_trips_19_20_7.***

```
# Calculate Day of the week as the variable week_day. Save the new dataframe as all_trips_19_20_7

all_trips_19_20_7 <- all_trips_19_20_6 %>%
  mutate(week_day = wday(start_time, label = TRUE, abbr = TRUE))


# View 'all_trips_19_20_7'

skim_without_charts(all_trips_19_20_7)
```

Table 1: Data summary

Name	all_trips_19_20_7
Number of rows	4233914
Number of columns	13
Column type frequency:	
character	3
factor	3
numeric	5
POSIXct	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
trip_id	0	1	7	16	0	4233914	0
start_station_name	0	1	5	43	0	643	0
end_station_name	0	1	5	43	0	644	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	977102	0.77	FALSE	2	Mal: 2399200, Fem: 857612
user_type	0	1.00	FALSE	2	mem: 3310385, cas: 923529
week_day	0	1.00	TRUE	7	Tue: 658968, Thu: 652542, Wed: 652142, Fri: 637177

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
bike_id	418911	0.90	3380.25	1902.49	1	1727	3451	5046	6946
start_station_id	0	1.00	202.02	156.25	1	77	174	289	675
end_station_id	0	1.00	202.82	156.35	1	77	174	291	675
birth_year	956712	0.77	1984.10	10.78	1929	1979	1987	1992	2003
trip_duration	0	1.00	1116.11	2082.95	60	402	692	1250	86385

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
start_time	0	1	2019-01-01 00:04:37	2020-03-31 23:51:34	2019-08-05 07:52:58	3695904
end_time	0	1	2019-01-01 00:11:07	2020-04-01 07:38:49	2019-08-05 08:06:14	3628412

c) month_name

- ***(Manipulation_12) Calculate the Months of the year as the variable month_name. Save the new dataframe as all_trips_19_20_8.***

```
# Calculate the Months of the year as the variable month_name. Save the new dataframe as all_trips_19_2

all_trips_19_20_8 <- all_trips_19_20_7 %>%
  mutate(month_name = month(start_time, label = TRUE, abbr = TRUE))

# View 'all_trips_19_20_8'

glimpse(all_trips_19_20_8)
```

```
## Rows: 4,233,914
## Columns: 14
## $ trip_id            <chr> "B9ED9D8CE75F542B", "7DC888EB586ED128", "681409CD39~
## $ bike_id            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_time         <dttm> 2020-01-26 01:51:36, 2020-01-07 15:56:04, 2020-01-~
## $ end_time           <dttm> 2020-01-26 01:52:36, 2020-01-07 15:57:04, 2020-01-~
## $ start_station_id   <dbl> 41, 174, 174, 291, 91, 14, 198, 164, 197, 238, 199,~
## $ start_station_name <chr> "Federal St & Polk St", "Canal St & Madison St", "C~
## $ end_station_id     <dbl> 394, 77, 77, 291, 77, 14, 198, 164, 197, 238, 199, ~
## $ end_station_name   <chr> "Clark St & 9th St (AMLI)", "Clinton St & Madison S~
## $ gender             <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

```
## $ birth_year          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ user_type           <fct> member, member, member, member, member, member, mem~
## $ trip_duration       <dbl> 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,~
## $ week_day            <ord> Sun, Tue, Wed, Wed, Wed, Tue, Mon, Wed, Mon, Fri, T~
## $ month_name          <ord> Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, F~
```

d) bike_route

- ***(Manipulation_13) Calculate the route (station pair) as the variable bike_route. Save the new dataframe as all_trips_19_20_9.***

```
# Calculate the route (station pair) as the variable bike_route. Save the new dataframe as all_trips_19_

all_trips_19_20_9 <- all_trips_19_20_8 %>%
  mutate(bike_route = paste0(start_station_name, " - ", end_station_name))

# View 'all_trips_19_20_9'

glimpse(all_trips_19_20_9)
```

```
## Rows: 4,233,914
## Columns: 15
## $ trip_id            <chr> "B9ED9D8CE75F542B", "7DC888EB586ED128", "681409CD39~
## $ bike_id            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_time         <dttm> 2020-01-26 01:51:36, 2020-01-07 15:56:04, 2020-01-~
## $ end_time           <dttm> 2020-01-26 01:52:36, 2020-01-07 15:57:04, 2020-01-~
## $ start_station_id   <dbl> 41, 174, 174, 291, 91, 14, 198, 164, 197, 238, 199,~
## $ start_station_name <chr> "Federal St & Polk St", "Canal St & Madison St", "C~
## $ end_station_id     <dbl> 394, 77, 77, 291, 77, 14, 198, 164, 197, 238, 199, ~
## $ end_station_name   <chr> "Clark St & 9th St (AMLI)", "Clinton St & Madison S~
## $ gender             <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ birth_year         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ user_type          <fct> member, member, member, member, member, member, mem~
## $ trip_duration      <dbl> 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,~
## $ week_day           <ord> Sun, Tue, Wed, Wed, Wed, Tue, Mon, Wed, Mon, Fri, T~
## $ month_name         <ord> Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, F~
## $ bike_route         <chr> "Federal St & Polk St - Clark St & 9th St (AMLI)", ~
```

e) rider_age

- ***(Manipulation_14) Calculate the rider age as the variable rider_age. Save the new dataframe as all_trips_19_20_10.***

```
# Calculate the rider age as the variable rider_age. Save the new dataframe as all_trips_19_20_10.

all_trips_19_20_10 <- all_trips_19_20_9 %>%
  mutate(rider_age = year(start_time) - birth_year)

# View 'all_trips_19_20_10'

glimpse(all_trips_19_20_10)
```

```
## Rows: 4,233,914
## Columns: 16
## $ trip_id            <chr> "B9ED9D8CE75F542B", "7DC888EB586ED128", "681409CD39~
## $ bike_id            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_time         <dttm> 2020-01-26 01:51:36, 2020-01-07 15:56:04, 2020-01-~
## $ end_time           <dttm> 2020-01-26 01:52:36, 2020-01-07 15:57:04, 2020-01-~
## $ start_station_id   <dbl> 41, 174, 174, 291, 91, 14, 198, 164, 197, 238, 199,~
## $ start_station_name <chr> "Federal St & Polk St", "Canal St & Madison St", "C~
## $ end_station_id     <dbl> 394, 77, 77, 291, 77, 14, 198, 164, 197, 238, 199, ~
## $ end_station_name   <chr> "Clark St & 9th St (AMLI)", "Clinton St & Madison S~
## $ gender             <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ birth_year         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ user_type          <fct> member, member, member, member, member, member, mem~
## $ trip_duration      <dbl> 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,~
## $ week_day           <ord> Sun, Tue, Wed, Wed, Wed, Tue, Mon, Wed, Mon, Fri, T~
## $ month_name         <ord> Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, Jan, F~
## $ bike_route         <chr> "Federal St & Polk St - Clark St & 9th St (AMLI)", ~
## $ rider_age          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

Metadata of the calculated variables / column headers :

- trip_duration – Total duration of the trip, measured in seconds.
- week_day – Day of the week on which the trip started (e.g; Mon, Tue).
- month_name – Month in which the trip took place (e.g; Jan, Feb).
- bike_route – Route of the rider's trip, defined by combining the start and end station names (e.g; "Station A → Station B").
- rider_age – Age of the rider, calculated up to the exact day the trip started.

Descriptive analysis :

Note :

Most busy => Highest total trip count

Least busy => Least total trip count

1. Trip duration by Usertype :

a) **Maximum, Minimum, Average -**

```
# Trip duration statistics

trip_duration_stat <- all_trips_19_20_10 %>%
  group_by(user_type) %>%
  summarise('maximum_duration (minutes)' = max(trip_duration)/60, 'minimum_duration (minutes)' = min(tri

# View

head(trip_duration_stat)
```

```
## # A tibble: 2 x 4
##   user_type maximum_duration (mi~1 minimum_duration (mi~2 average_duration (mi~3
##   <fct>                     <dbl>                  <dbl>                  <dbl>
```

```
## 1 casual                    1439.                  1.02                  39.5
## 2 member                    1440.                  1                     12.8
## # i abbreviated names: 1: `maximum_duration (minutes)`,
## #   2: `minimum_duration (minutes)`, 3: `average_duration (minutes)`
```

(1A)

1. Annual members ride an average of 12.8 minutes while Casuals ride for 39 minutes on an average.

2. Most busy day (Mode of 'week_day') & Least busy day of the week (2019 - 2020 Q1):

```r
# Summarise counts per user_type & week_day once

day_counts <- all_trips_19_20_10 %>%
  group_by(user_type, week_day) %>%
  summarise(trip_count = n(), .groups = 'drop')

# Find busiest day per user_type

busiest <- day_counts %>%
  group_by(user_type) %>%
  slice_max(trip_count, n = 1, with_ties = FALSE) %>%
  select(user_type, busiest_day = week_day, busiest_trip_count = trip_count)

# Find least busy day per user_type

least_busy <- day_counts %>%
  group_by(user_type) %>%
  slice_min(trip_count, n = 1, with_ties = FALSE) %>%
  select(user_type, least_busy_day = week_day, least_busy_trip_count = trip_count)

# Join busiest and least busy by user_type

most_least_busy_day <- busiest %>%
  inner_join(least_busy, by = "user_type")

# View the summary

head(most_least_busy_day)
```

```
## # A tibble: 2 x 5
## # Groups:   user_type [2]
##   user_type busiest_day busiest_trip_count least_busy_day least_busy_trip_count
##   <fct>     <ord>                   <int> <ord>                           <int>
## 1 casual    Sat                    215174 Tue                             93035
## 2 member    Tue                    565933 Sun                            291600
```

(2A)

1. Casual riders : Saturday is the most busy day, while Tuesday is the least busy day
2. Annual members : Tuesday is the most busy day, while Sunday is the least busy day

Analysis Summaries :

- Blue Ocean Action plan = Raise the resources on high leverage activities, Create high leverage activities, Reduce resources on low leverage activities, Remove the low leverage activities
  - Always summarize every analysis keeping this Action plan in mind.

The 'Whole' method of analysis : 'Whole' method equation = (V1, V2, V3,..) x CoreC0.(Combination(C1, C2, C3,…)) *where, V = Value, C = Category, CoreC0 = Core category*

Prioritizing categories :

1. Time (Wide, then narrow. Eg; Month, then Weekday)
2. Location (Wide, then narrow. Eg; Station, then Route.)
3. User_characteristics

Eg;

(V1) x CoreC0.(Combination(C1, C2, C3)) = V1.C1, V1.C1.C2, V1.C1.C3, V1.C1.C2.C3; V1.C2, V1.C2.C3; V1.C3 (Every combination has C0 in it.)

Here, (Trip count, Trip duration) x *user_type.*(month_name, week_day, start_station_name, bike_route, gender, rider_age)

1. V1 = Total Trip count

a1) What is the day with the highest trip count for Casual riders and Annual members respectively.

b) month, station, route, gender, age (5C1 = 5)

c) month-station, month-route, month-gender, month-age, station-route, station-gender, station-age, route-gender, route-age, gender-age (5C2 = 10)

d) month-station-route, month-station-gender, month-station-age, month-route-gender, month-route-age, month-gender-age, station-route-gender, station-route-age, station-gender-age, route-gender-age (5C3 = 10)

e) month-station-route-gender, month-station-route-age, month-station-gender-age, month-route-gender-age, station-route-gender-age (5C4 = 5)

f) month-station-route-gender-age (5C5 = 1)

a2) month

b) station, route, gender, age (4C1 = 4)

c) station-route, station-gender, station-age, route-gender, router-age, gender-age (4C2 = 6)

d) station-route-gender, station-router-age, station-gender-age, route-gender-age (4C3 = 4)

e) station-route-gender-age (4C4 = 1)

a3) station

b) route, gender, age

c) route-gender, route-age, gender-age

d) route-gender-age

a4) route

b) gender, age

c) gender-age

a5) gender

b) age

a6) age

2. V2 = Total trip duration

- same pattern as above

3. V3 = Average trip duration

- same pattern as above

'Wide to Narrow' method of analysis : The above method is complicated, but keep it's logic in your mind. And use this method instead - for faster & efficient analysis.

Imagine going from Wide to narrow in category for each Value (V).

V1 : Trip count

C0 : Core category

Eg;

1. First, look at Wider time ie; Monthly patterns. **V1 x C0 x C1**
2. Then, Narrower ie; Weekday patterns (Analyse the weekdays only after knowing which months have high count, low count, what the monthly patterns are…) **V1 x C0 x C2**
3. Then, Wider location ie; Stations' patterns (Analyse the stations only after analyzing the patterns and summaries of it's parent ie; Weekdays, Months) **V1 x C0 x C3**
4. Then, Narrower, ie; Routes' patterns (Analyse the Routes only after knowing the patterns of it's parent ie; Stations) **V1 x C0 x C4**
5. Then, User_characteristic patterns (Analyse this only after building a connection with all the above categories in a Hierarchical manner on the Value you are evaluating.) **V1 x C0 x C5**

Don't complicate the analysis by combining the Categories. We only need a single category combined with the Core category (Here, Core category = user_type). And then, just do this *Hierarchical analysis* and find the Relation between categories by connecting the dots hierarchically.

Similar process for other Values.

Eg; All the analyses done below is using this methodology. For better understanding, Keep attention ahead !

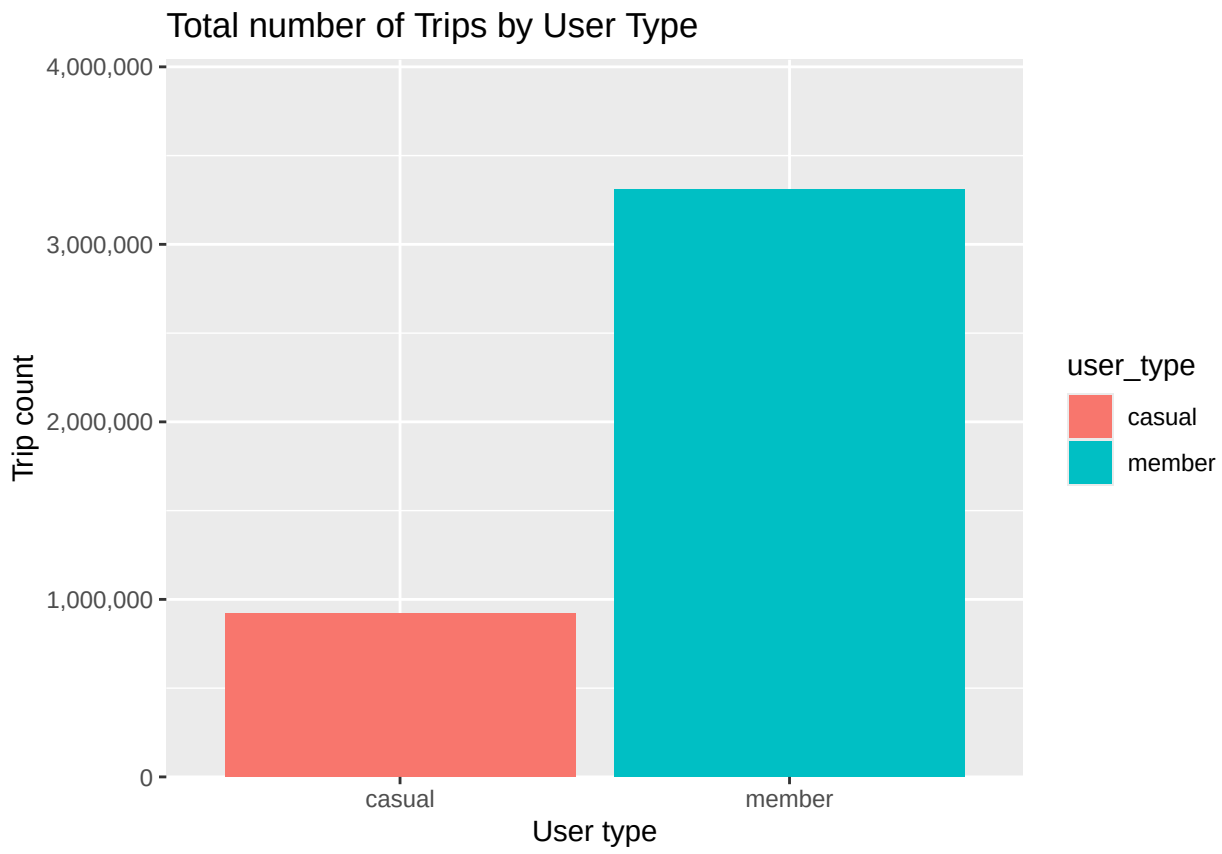
Now that we know the system to do the analysis, let's go ahead and do it.

1. Trip count - ***0) Total number of trips by Usertype (2019 - 2020 Q1) :***

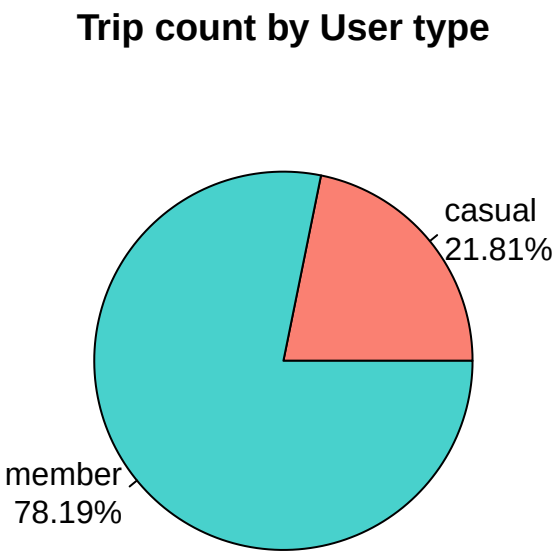
```
# Total number of trips by Usertype

summary0_df <- all_trips_19_20_10 %>%
  group_by(user_type) %>%
  summarise(total_trip_count = n())

# View the summary

head(summary0_df)
```

```
## # A tibble: 2 x 2
##   user_type total_trip_count
##   <fct>                <int>
## 1 casual              923529
## 2 member             3310385
```

```
# Total number of trips by Usertype - Column graph

ggplot(data = summary0_df,
       mapping = aes(x = user_type,
                     y = total_trip_count,
                     fill = user_type)) +
  geom_col() +
  labs(title = "Total number of Trips by User Type",
       x = "User type",
       y = "Trip count") +
  scale_y_continuous(limits = c(0, 4000000),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01)))
```




```
# Percentage number of trips by Usertype - Pie graph

summary0_df <- summary0_df %>%
  mutate(percentage_trip_count = total_trip_count/sum(total_trip_count)*100)

annotation0 <- paste0(summary0_df$user_type, "\n",
                      round(summary0_df$percentage_trip_count,2), "%")

pie(summary0_df$percentage_trip_count,
    labels = annotation0,
    col = c("salmon", "mediumturquoise"),
    main = "Trip count by User type")
```



(3A)

1. Casual riders : Trip count is approx. 0.92 Million, which is 22% of the Total trip count.
2. Annual members : Trip count is approx. 3.3 Million, which is 78% of the Total trip count.

1) Total number of trips in a Month by Usertype (2019 - 2020 Q1) :

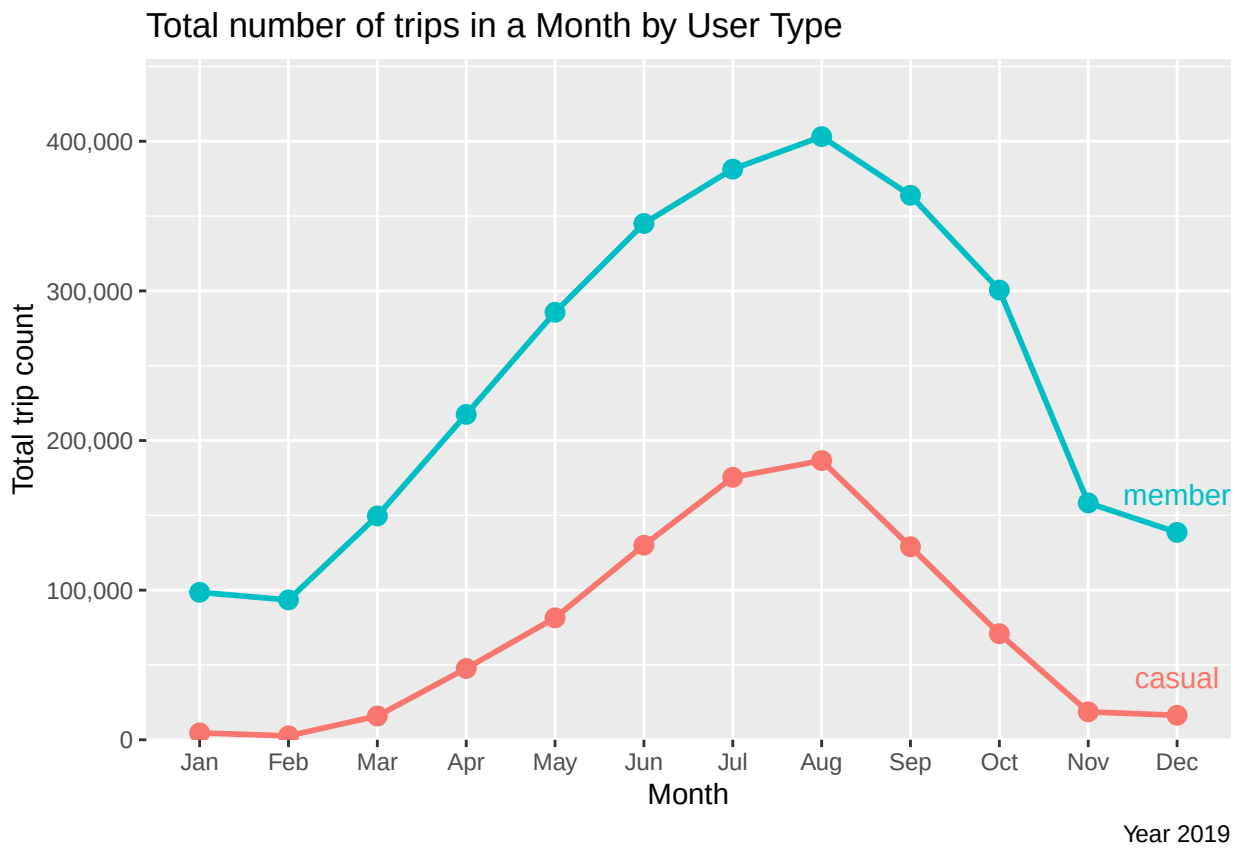
```
# Total number of trips in a Month by Usertype

summary1_df <- all_trips_19_20_10 %>%
  filter(year(start_time) == 2019) %>%
  group_by(month_name, user_type) %>%
  summarize(
    total_trip_count = n()
    )
```

```
## `summarise()` has grouped output by 'month_name'. You can override using the
## `.groups` argument.
```

```
# Summary wide table view

glimpse(summary1_df %>% pivot_wider(
    names_from = month_name,
    values_from = c(total_trip_count)))
```

```
## Rows: 2
## Columns: 13
## $ user_type <fct> casual, member
## $ Jan       <int> 4589, 98554
## $ Feb       <int> 2627, 93463
## $ Mar       <int> 15877, 149596
## $ Apr       <int> 47665, 217454
## $ May       <int> 81505, 285730
## $ Jun       <int> 130061, 345068
## $ Jul       <int> 175408, 381414
## $ Aug       <int> 186613, 403117
## $ Sep       <int> 128985, 363879
## $ Oct       <int> 70887, 300586
## $ Nov       <int> 18653, 158314
## $ Dec       <int> 16365, 138593
```

```r
# Total number of trips in a Month by Usertype - Line plot

label_data_1 <- summary1_df %>%
  filter(month_name == "Dec")

ggplot(data = summary1_df,
       mapping = aes(x = month_name,
                     y = total_trip_count,
                     colour = user_type,
                     group = user_type)) +
  geom_line(size =  1) +
  geom_point(size = 3) +
  labs(title = "Total number of trips in a Month by User Type",
       x = "Month",
       y = "Total trip count",
       caption = "Year 2019") +
  geom_text(data = label_data_1,
            aes(label = user_type),
            vjust = -1.3,
            size = 4,
            show.legend = FALSE) +
  scale_y_continuous(limits = c(0, 450000),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01))) +
  theme(legend.position = "none")
```



(4A)

1. Casual riders : January and February are the least busy months with February as the minimum (Below the count of 5,000). Then after February, a linear increase upto the Most busiest month August - a 5900% increase from February to a count of approx. 185,000.Then a linear decrease until November to a count of approx. 18,000. Then a slight linear decrease until December. Trip counts exceeded 50,000 in May, June, July, August, September, and October

2. Annual members : January and February are the least busy months with February as the minimum (Below the count of 10,000). Then after February, a linear increase upto the Most busiest month August - a 3900% increase from February to a count of approx. 400,000. Then a linear decrease until October, and a sudden drop in count by 50% in November. Then a linear decrease until December. Trip counts exceeded 250,000 in May, June, July, August, September, and October.

2) Total number of trips in a Weekday by Usertype (2019 - 2020 Q1) :

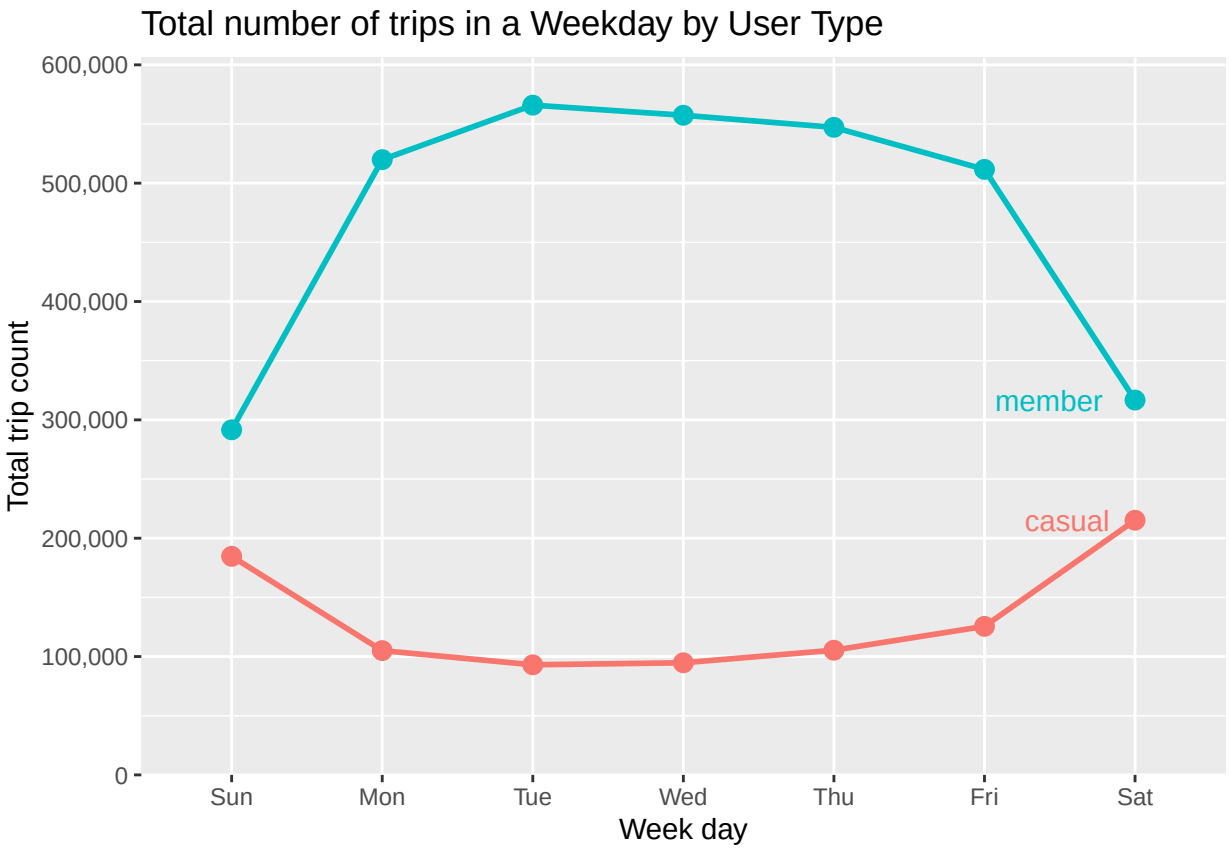
```
# Total number of trips in a Weekday by Usertype

summary2_df <- all_trips_19_20_10 %>%
  group_by(week_day, user_type) %>%
  summarize(
    total_trip_count = n()
    )
```

```
## `summarise()` has grouped output by 'week_day'. You can override using the
## `.groups` argument.
```

```r
# Summary wide table view

head(summary2_df %>% pivot_wider(
    names_from = week_day,
    values_from = c(total_trip_count)))
```

```
## # A tibble: 2 x 8
##   user_type    Sun    Mon    Tue    Wed    Thu    Fri    Sat
##   <fct>      <int>  <int>  <int>  <int>  <int>  <int>  <int>
## 1 casual    184684 105015  93035  94740 105361 125520 215174
## 2 member    291600 519888 565933 557402 547181 511657 316724
```

```r
# Total number of trips in a Weekday by Usertype  - Line plot

label_data_2 <- summary2_df %>%
  filter(week_day == "Sat")

ggplot(data = summary2_df,
       mapping = aes(x = week_day,
                     y = total_trip_count,
                     colour = user_type,
                     group = user_type)) +
  geom_line(size =  1) +
  geom_point(size = 3) +
  labs(title = "Total number of trips in a Weekday by User Type",
       x = "Week day",
       y = "Total trip count") +
  geom_text(data = label_data_2,
            aes(label = user_type),
            hjust = 1.3,
            size = 4,
            show.legend = FALSE) +
  scale_y_continuous(limits = c(0, 600000),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01))) +
  theme(legend.position = "none")
```



(5A)

1. Casual riders : Saturday and Sunday has a count of approx. 200,000. Then, the count reduces sharply by 35% through Monday ie; below 130,000 count and reaches a Minimum at Tuesday. Then it slowly and steadily increase until Friday, and then a sharp increase on Saturday.
2. Annual members : Sunday is the least busy day (approx. 300,000), then the count sharply increase by 70% on Monday ie; above 500,000 count, and then reaches a Maximum on Tuesday. Then a slow & steady decrease until Friday, but the count stays above 500,000. Then a sharp decrease by Saturday to an approx. count of 300,000.

3) Total number of trips in a Station by Usertype (2019 - 2020 Q1) :

- **Let's look at the Top 20 Stations by Trip count & User type.**

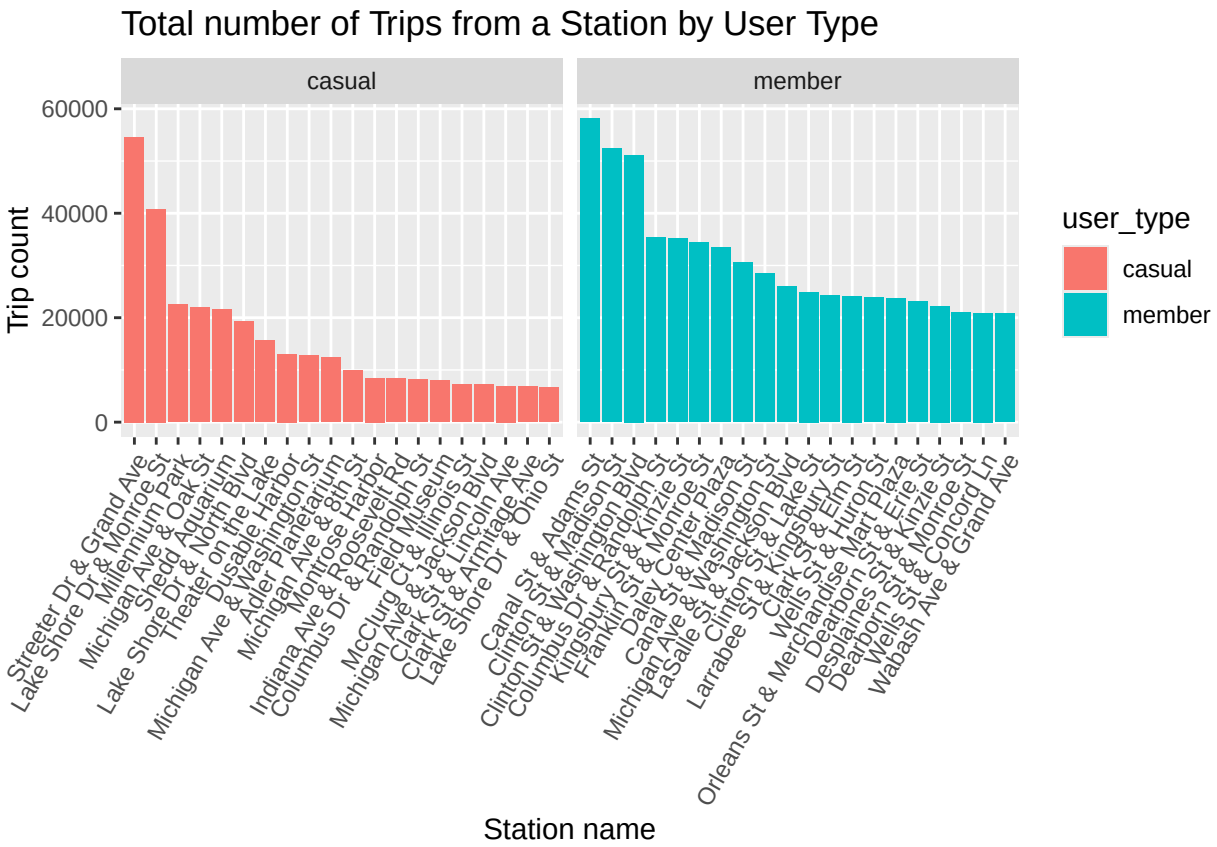
```
# Total number of trips from a Station by Usertype : Top 20 Stations

summary3_df <- all_trips_19_20_10 %>%
  group_by(start_station_name, user_type) %>%
  summarize(
    total_trip_count = n()
    ) %>%
  group_by(user_type) %>%
  slice_max(order_by = total_trip_count, n = 20)
```

```
## `summarise()` has grouped output by 'start_station_name'. You can override
## using the `.groups` argument.
```

```
# Summary long table view

glimpse(summary3_df %>% pivot_wider(
    names_from = user_type,
    values_from = c(total_trip_count)) %>%
      arrange(desc(casual)))
```

```
## Rows: 38
## Columns: 3
## $ start_station_name <chr> "Streeter Dr & Grand Ave", "Lake Shore Dr & Monroe ~
## $ casual             <int> 54582, 40793, 22503, 21998, 21604, 19331, 15582, 13~
## $ member             <int> NA, NA, NA, NA, NA, NA, NA, NA, 28488, NA, NA, NA, ~
```

```
# Total number of trips from a Station by Usertype : Top 20 stations

ggplot(summary3_df, aes(x = reorder_within(start_station_name, -total_trip_count, user_type),
                        y = total_trip_count,
                        fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Total number of Trips from a Station by User Type",
       x = "Station name",
       y = "Trip count") +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))+
  scale_x_reordered() +
  facet_wrap(~user_type, scales = "free_x")
```



We can see that after the Top 10 stations, the remaining are having a low level of activity.

- **So let's look at only the *Top 10 Stations for Trip count.***

```
# Total number of trips from a Station by Usertype : Top 10 Stations

summary4_df <- all_trips_19_20_10 %>%
  group_by(start_station_name, user_type) %>%
  summarize(
    total_trip_count = n()
    ) %>%
  group_by(user_type) %>%
  slice_max(order_by = total_trip_count, n = 10)
```

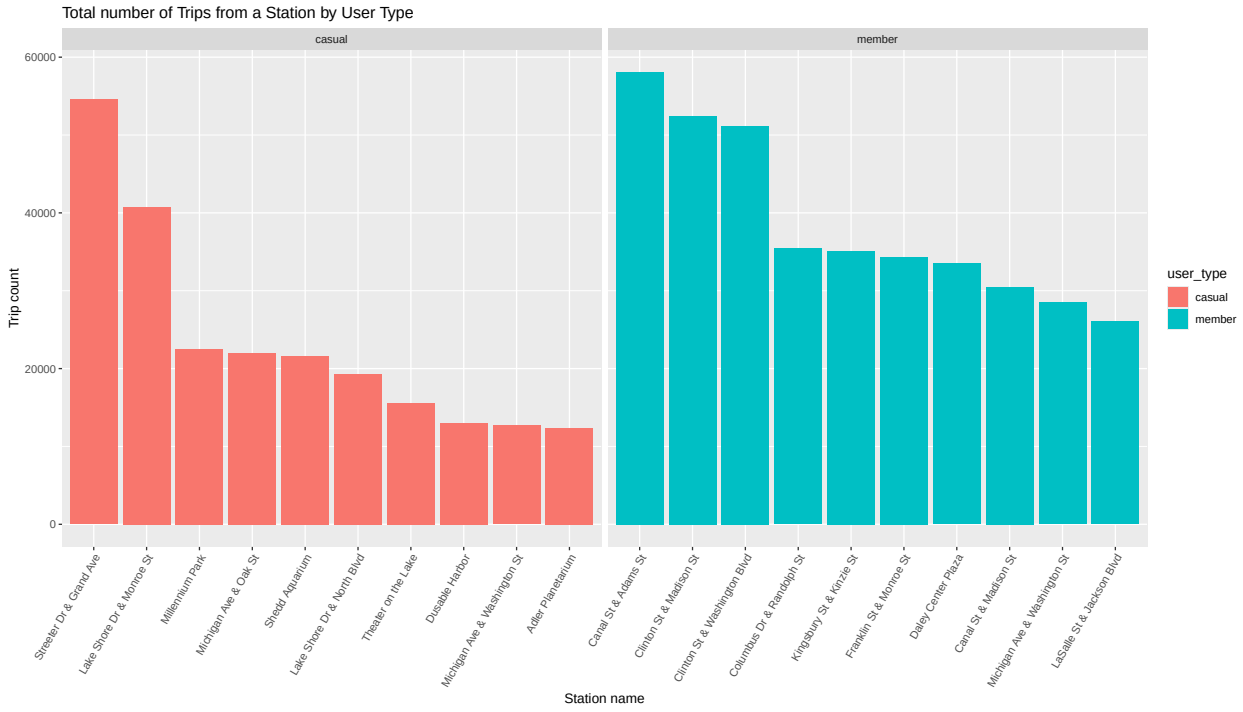
```
## `summarise()` has grouped output by 'start_station_name'. You can override
## using the `.groups` argument.
```

```
# Summary long table view

glimpse(summary4_df %>% pivot_wider(
    names_from = user_type,
    values_from = c(total_trip_count)) %>%
      arrange(desc(member)))
```

```
## Rows: 19
```

```
## Columns: 3
## $ start_station_name <chr> "Canal St & Adams St", "Clinton St & Madison St", "~
## $ casual             <int> NA, NA, NA, NA, NA, NA, NA, NA, 12709, NA, 54582, 4~
## $ member             <int> 58086, 52415, 51133, 35415, 35104, 34361, 33485, 30~
```

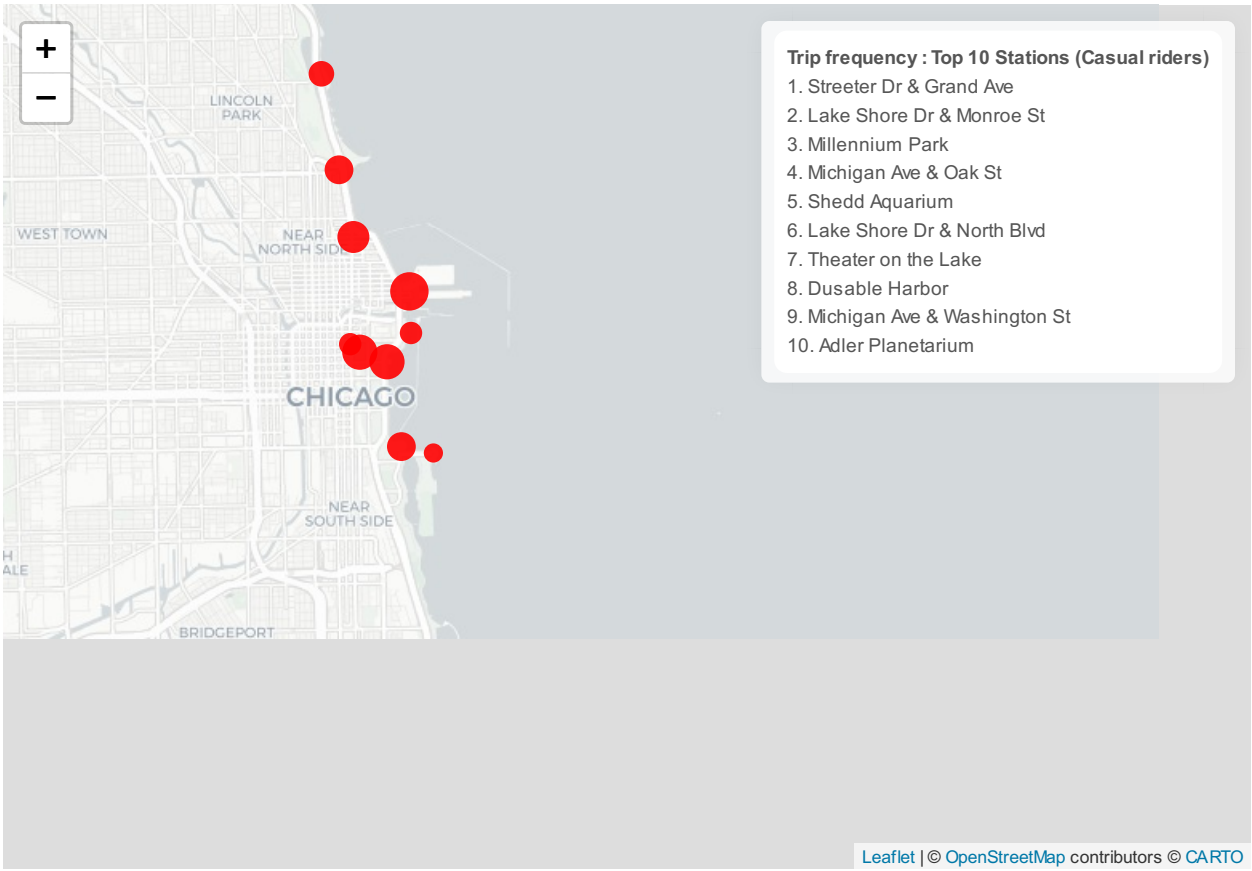
```
# Top 10 Stations by Trip count  - Column chart

ggplot(summary4_df, aes(x = reorder_within(start_station_name, -total_trip_count, user_type),
                        y = total_trip_count,
                        fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Total number of Trips from a Station by User Type",
       x = "Station name",
       y = "Trip count") +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))+
  scale_x_reordered() +
  facet_wrap(~user_type, scales = "free_x")
```



- **Total number of trips from a Station by Casual riders : Top 10 Stations - MAP**

```
# Total number of trips from a Station by Casual riders : Top 10 Stations - MAP

stations_df <- read.csv("top_stations_trips_geocoded.csv") # Extracted the coordinates of stations usin

# Step 1: Filter casual stations and rank them
casual_stations_ranked <- stations_df %>%
  filter(user_type == "casual") %>%
  arrange(desc(total_trip_count)) %>%
  mutate(
    rank = row_number(),
```

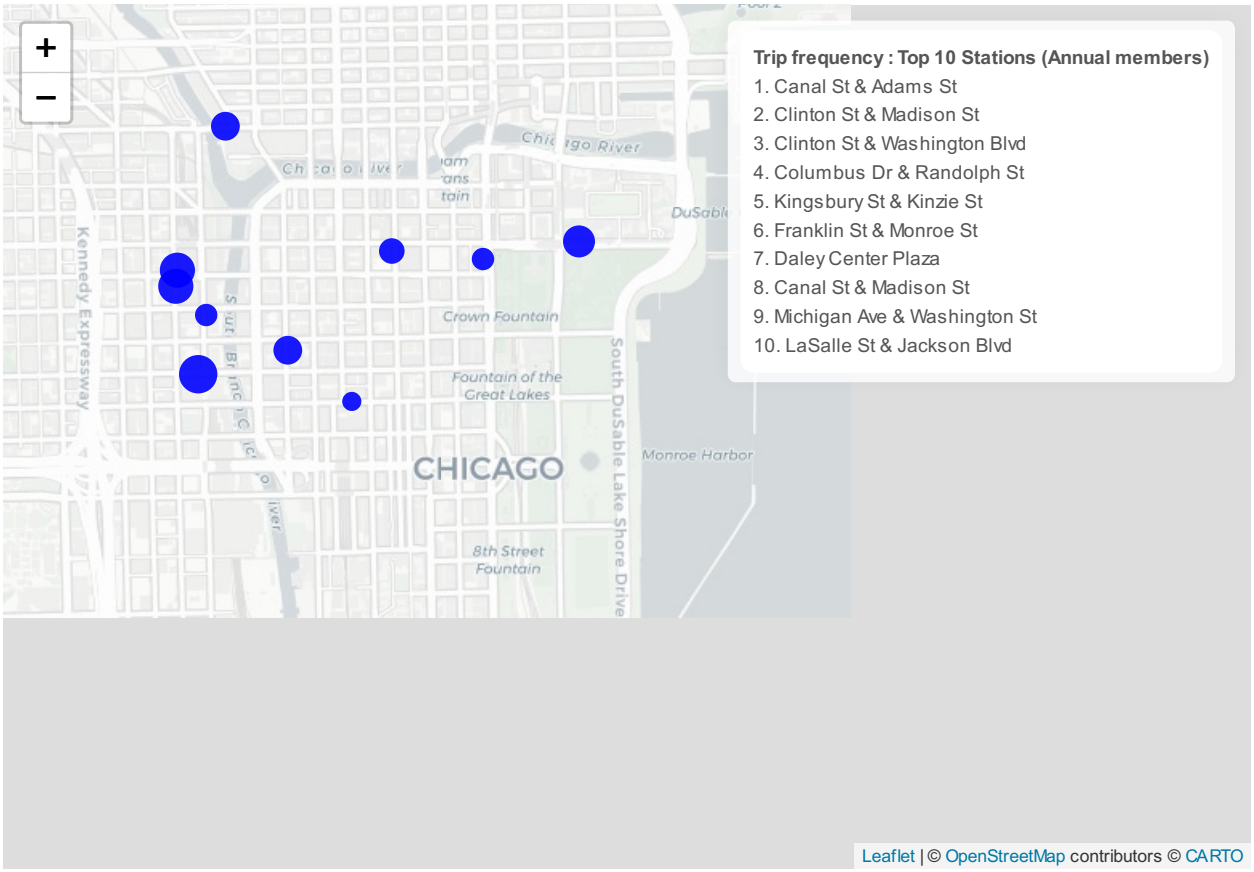
```
    marker_size = scales::rescale(max(rank) + 1 - rank, to = c(6, 12))
  )

# Step 2: Create the map
m1 <- leaflet(casual_stations_ranked) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(
    ~longitude, ~latitude,
    radius = ~marker_size,
    color = "red",
    fillOpacity = 0.9,
    stroke = FALSE,
    popup = ~paste0(
      "<strong>Rank ", rank, ":</strong> ", start_station_name, "<br>",
      "Trips: ", total_trip_count
    )
  ) %>%
  addControl(
    html = paste0(
      "<div style='text-align:left; padding:8px; font-size:12px; background:white; border-radius:8px;'>
      "<strong>Trip frequency : Top 10 Stations (Casual riders)</strong><br>",
      paste0(
        casual_stations_ranked %>%
          arrange(rank) %>%
          mutate(label = paste0(rank, ". ", start_station_name)) %>%
          pull(label),
        collapse = "<br>"
      ),
      "</div>"
    ),
    position = "topright"
  )

m1
```



- **Total number of trips from a Station by Annual members : Top 10 Stations - MAP**

```
# Total number of trips from a Station by Annual members : Top 10 Stations - MAP

# Step 1: Filter casual stations and rank them
member_stations_ranked <- stations_df %>%
  filter(user_type == "member") %>%
  arrange(desc(total_trip_count)) %>%
  mutate(
    rank = row_number(),
    marker_size = scales::rescale(max(rank) + 1 - rank, to = c(6, 12))
  )

# Step 2: Create the map
m2 <- leaflet(member_stations_ranked) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(
    ~longitude, ~latitude,
    radius = ~marker_size,
    color = "blue",
    fillOpacity = 0.9,
    stroke = FALSE,
    popup = ~paste0(
      "<strong>Rank ", rank, ":</strong> ", start_station_name, "<br>",
      "Trips: ", total_trip_count
```

```
    )
  ) %>%
  addControl(
    html = paste0(
      "<div style='text-align:left; padding:8px; font-size:12px; background:white; border-radius:8px;'>
      "<strong>Trip frequency : Top 10 Stations (Annual members)</strong><br>",
      paste0(
        member_stations_ranked %>%
          arrange(rank) %>%
          mutate(label = paste0(rank, ". ", start_station_name)) %>%
          pull(label),
        collapse = "<br>"
      ),
      "</div>"
    ),
    position = "topright"
  )

m2
```



(6A)

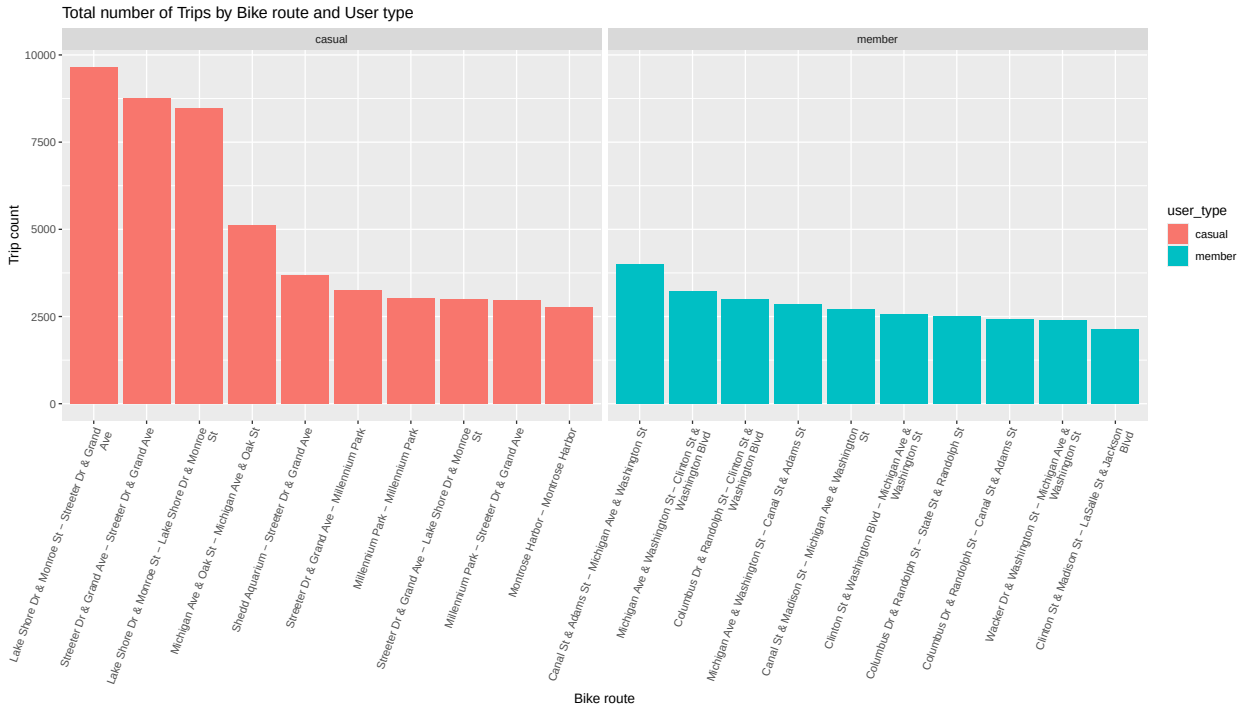
1. Casual riders : "Streeter Dr & Grand Axe" is the most busiest station for Casual riders. The Top 10 Busy stations are located near the Lake side.
2. Annual members : "Canal St & Adams St" is the most busiest station for Annual members. The Top 10 Busy stations are concentrated away from the Lake side.

4) Total number of trips by Bike route and User type (2019 - 2020 Q1) :

```
# Total number of trips from a Bike route by Usertype : Top 10 Routes

summary5_df <- all_trips_19_20_10 %>%
  group_by(bike_route, user_type) %>%
  summarize(
    total_trip_count = n(),
    .groups = "drop") %>%
  group_by(user_type) %>%
  slice_max(order_by = total_trip_count, n = 10)

# Summary long table view

glimpse(summary5_df %>% pivot_wider(
    names_from = user_type,
    values_from = c(total_trip_count)) %>%
      arrange(desc(casual)))
```

```
## Rows: 20
## Columns: 3
## $ bike_route <chr> "Lake Shore Dr & Monroe St - Streeter Dr & Grand Ave", "Str~
## $ casual     <int> 9660, 8773, 8459, 5130, 3691, 3246, 3025, 3009, 2968, 2761,~
## $ member     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 3989, 3227, 2994, 2~
```

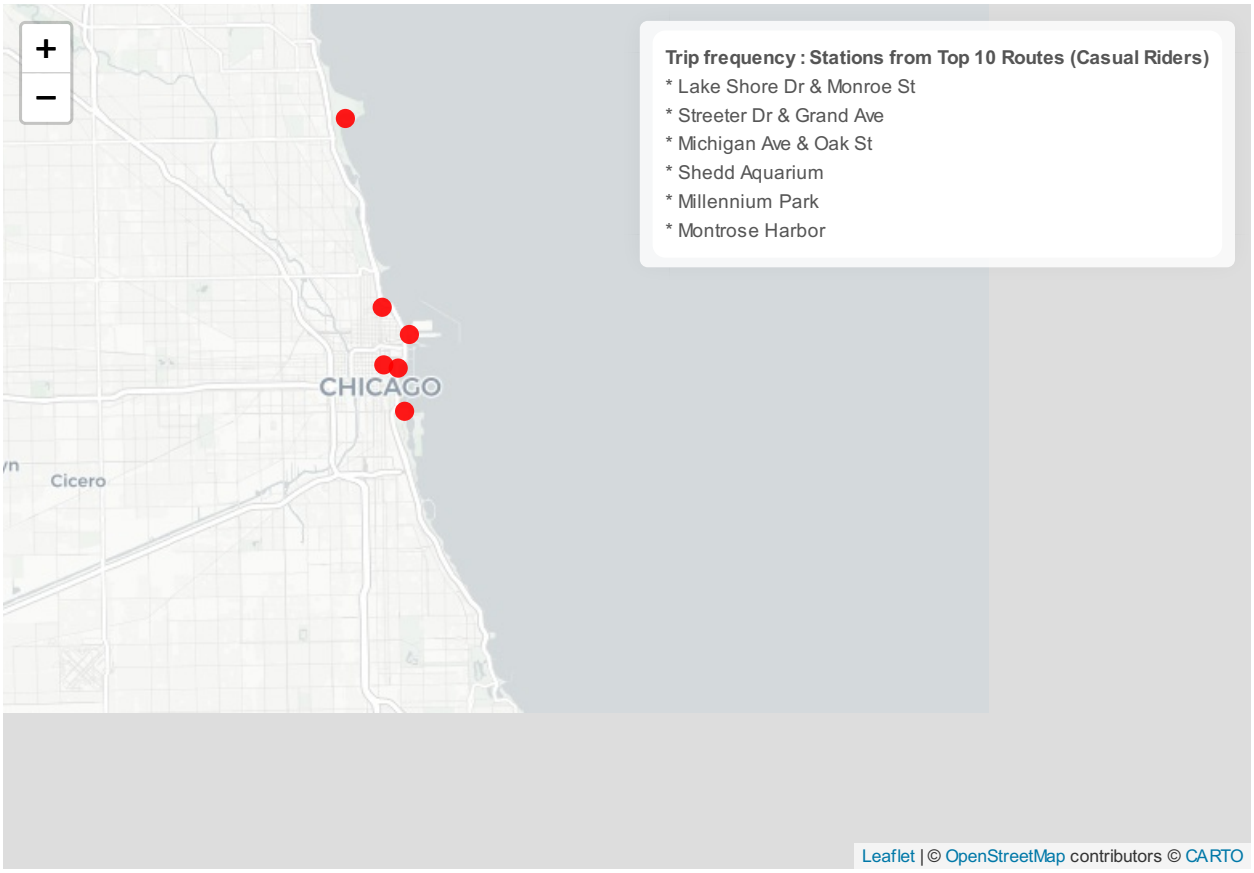
```
# Total number of trips by Bike route and User type : Top 10 routes - Column Chart

ggplot(summary5_df, aes(x = reorder_within(str_wrap(bike_route, 50), -total_trip_count, user_type),
                        y = total_trip_count,
                        fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Total number of Trips by Bike route and User type",
       x = "Bike route",
       y = "Trip count") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1))+
  scale_x_reordered() +
  facet_wrap(~user_type, scales = "free_x")
```



- **Total trip count by route for Casual riders -> MAP**

```
# Total trip count by route for Casual riders -> MAP

 ## Get top 10 routes for casual users
top_routes <- all_trips_19_20_10 %>%
  filter(user_type == "casual") %>%
  group_by(start_station_name, end_station_name) %>%
  summarise(total_trip_count = n(), .groups = "drop") %>%
  arrange(desc(total_trip_count)) %>%
  slice_head(n = 10)

## Extract start and end stations
start_stations <- top_routes %>%
  select(station_name = start_station_name)

end_stations <- top_routes %>%
  select(station_name = end_station_name)

## Combine and get unique station names
unique_stations <- bind_rows(start_stations, end_stations) %>%
  distinct(station_name)

## Extracted the coordinates of stations using geocode function, and then saved it in a csv file. Which
casual_routes_stations <- read.csv("casual_top_routes_stations_trips_geocoded.csv")


## Create the map
map_trip_count_casual_route <- leaflet(casual_routes_stations) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
```

```
  addCircleMarkers(
    ~longitude, ~latitude,
    radius = 6,
    color = "red",
    fillOpacity = 0.9,
    stroke = FALSE,
    popup = ~paste0(":</strong> ", station_name, "<br>"
    )
  ) %>%
  addControl(
    html = paste0(
      "<div style='text-align:left; padding:8px; font-size:12px; background:white; border-radius:8px;'>
      "<strong>Trip frequency : Stations from Top 10 Routes (Casual Riders)</strong><br>",
      paste0(
        casual_routes_stations %>%
          mutate(label = paste0(" * ", station_name)) %>%
          pull(label),
        collapse = "<br>"
      ),
      "</div>"
    ),
    position = "topright"
  )

map_trip_count_casual_route
```



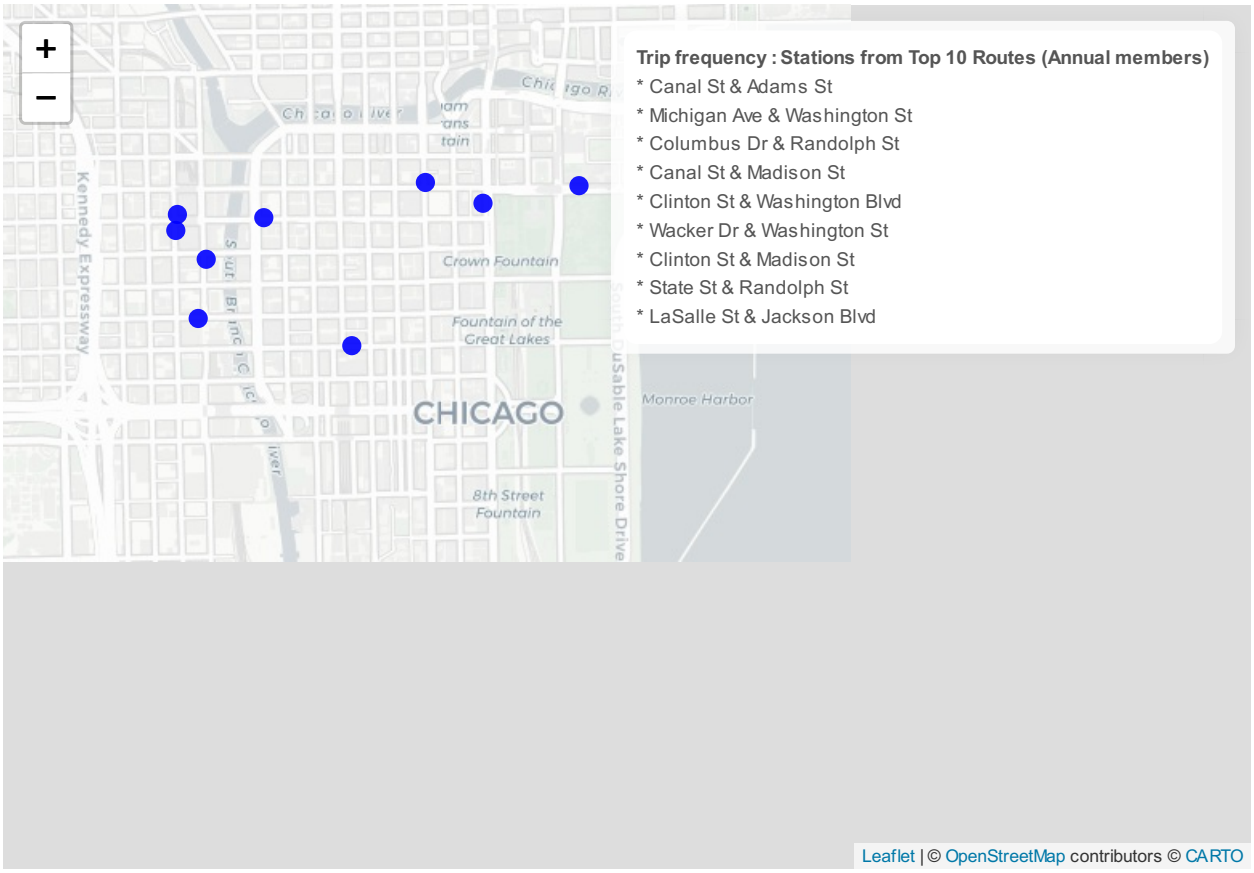
- **Total trip count by route for Annual members -> MAP**

```
# Total trip count by route for Annual members -> MAP

 ## Get top 10 routes for annual members
top_routes_1 <- all_trips_19_20_10 %>%
  filter(user_type == "member") %>%
  group_by(start_station_name, end_station_name) %>%
  summarise(total_trip_count = n(), .groups = "drop") %>%
  arrange(desc(total_trip_count)) %>%
  slice_head(n = 10)

## Extract start and end stations
start_stations_1 <- top_routes_1 %>%
  select(station_name = start_station_name)

end_stations_1 <- top_routes_1 %>%
  select(station_name = end_station_name)

## Combine and get unique station names
unique_stations_1 <- bind_rows(start_stations_1, end_stations_1) %>%
  distinct(station_name)
unique_stations_1
```

```
## # A tibble: 9 x 1
##   station_name
##   <chr>
## 1 Canal St & Adams St
## 2 Michigan Ave & Washington St
## 3 Columbus Dr & Randolph St
## 4 Canal St & Madison St
## 5 Clinton St & Washington Blvd
## 6 Wacker Dr & Washington St
## 7 Clinton St & Madison St
## 8 State St & Randolph St
## 9 LaSalle St & Jackson Blvd
```

```
## Extracted the coordinates of stations using geocode function, and then saved it in a csv file. Which
member_routes_stations <- read.csv("member_top_routes_stations_trips_geocoded.csv")


## Create the map
map_trip_count_member_route <- leaflet(member_routes_stations) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(
    ~longitude, ~latitude,
    radius = 6,
    color = "blue",
    fillOpacity = 0.9,
    stroke = FALSE,
    popup = ~paste0(":</strong> ", station_name, "<br>"
    )
  ) %>%
```

```
  addControl(
    html = paste0(
      "<div style='text-align:left; padding:8px; font-size:12px; background:white; border-radius:8px;'>
      "<strong>Trip frequency : Stations from Top 10 Routes (Annual members)</strong><br>",
      paste0(
        member_routes_stations %>%
          mutate(label = paste0(" * ", station_name)) %>%
          pull(label),
        collapse = "<br>"
      ),
      "</div>"
    ),
    position = "topright"
  )

map_trip_count_member_route
```



(7A)

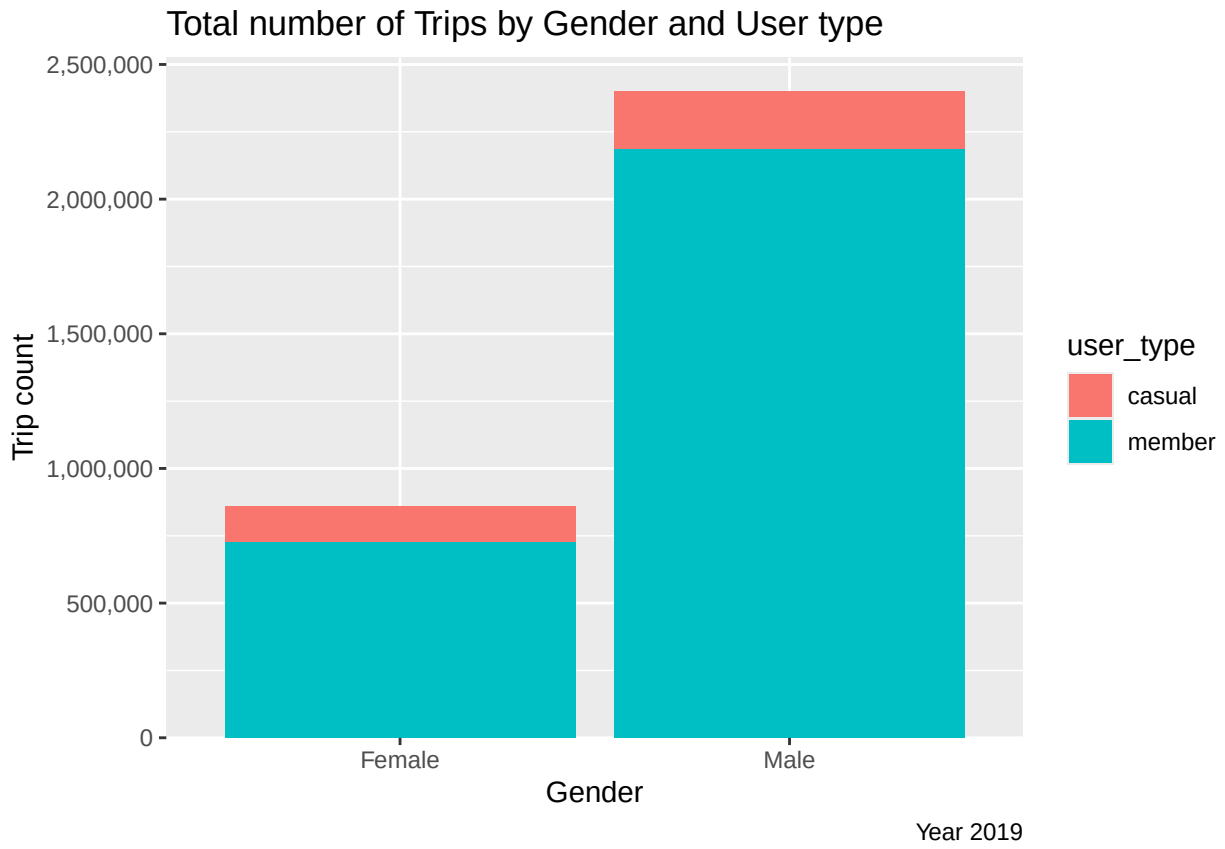
0. When Starting station and Ending station are same, then it means that the user went for a ride from the starting station, and after the ride, finished the ride at the same station
1. Casual riders : "Lake Shore Dr & Monroe St - Streeter Dr & Grand Ave" is the most busiest route for Casual riders. All stations of the Top 10 busy routes are located near the Lake side
2. Annual members : "Canal St & Adams St - Michigan Ave & Washington St" is the most busiest route for Annual members. All stations of the Top 10 busy routes are concentrated away from the Lake side.

5) Total number of trips by Gender and User type (2019 - 2020 Q1) :

```
# Total number of trips by Gender and Usertype- Summary

summary6_df <- all_trips_19_20_10 %>%
  filter(is.na(gender) == FALSE) %>%
  group_by(gender, user_type) %>%
  summarize(
    total_trip_count = n(), .groups = "drop"
    ) %>%
  group_by(user_type)

# Summary long table view

head(summary6_df %>% pivot_wider(
    names_from = user_type,
    values_from = c(total_trip_count)))
```

```
## # A tibble: 2 x 3
##   gender casual  member
##   <fct>   <int>   <int>
## 1 Female 131263  726349
## 2 Male   212493 2186707
```

```
# Total number of trips by Gender and User type  - Column Chart

ggplot(summary6_df, aes(x = gender,
                        y = total_trip_count,
                        fill = user_type)) +
  geom_col(position = "stack") +
  labs(title = "Total number of Trips by Gender and User type",
       x = "Gender",
       y = "Trip count",
       caption = "Year 2019") +
  scale_y_continuous(limits = c(0, 2500000),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01)))
```



(8A)

1. There are total 0.85 Million Trip counts for Female riders and 2.4 Million of that for Male riders in the Year 2019
2. Casual riders : Among casual riders, 38% of trips were made by females, while 62% were made by males.
3. Annual members : Among annual members, 25% of trips were made by females, whereas 75% were made by males.
4. Female riders :Among female riders, only 15% of total trips were made by casual riders, while a dominant 85% were taken by annual members.
5. Male riders : Among male riders, only 8.8% of total trips were made by casual riders, while a dominant 91.2% were taken by annual members.

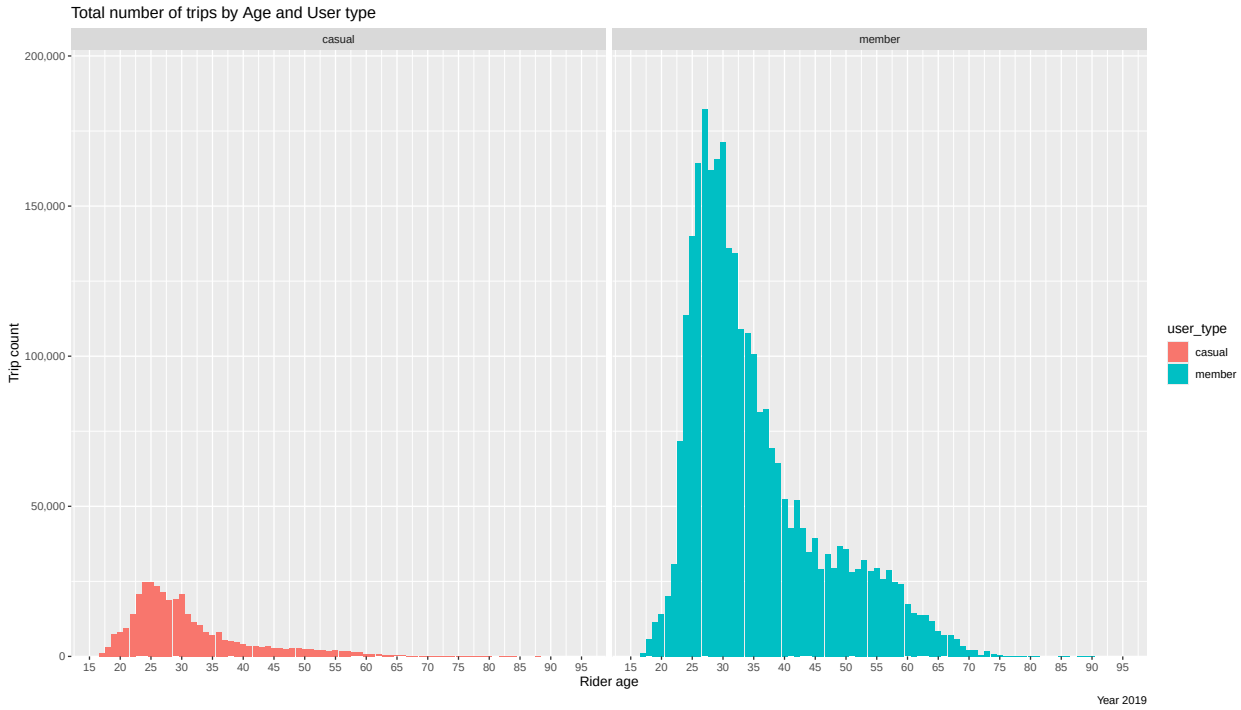
6) Total number of trips by Age and User type (2019 - 2020 Q1) :

```
# Total number of trips by Age and User type - Summary

summary7_df <- all_trips_19_20_10 %>%
  filter(is.na(rider_age) == FALSE) %>%
  group_by(rider_age, user_type) %>%
  summarize(
    total_trip_count = n(), .groups = "drop"
    ) %>%
```

```
  group_by(user_type)

# Summary long table view

glimpse(summary7_df %>% pivot_wider(
    names_from = user_type,
    values_from = c(total_trip_count)) %>%
      arrange(desc(rider_age)))
```

```
## Rows: 74
## Columns: 3
## $ rider_age <dbl> 90, 89, 88, 86, 85, 84, 83, 82, 81, 80, 79, 78, 77, 76, 75, ~
## $ casual    <int> NA, NA, 1, NA, NA, 3, 1, 4, NA, 2, 17, 4, 5, 15, 17, 9, 20, ~
## $ member    <int> 5, 35, 11, 20, 19, NA, NA, NA, 12, 88, 226, 119, 236, 93, 37~
```

```
# Total number of trips by Age and User type  - Column Chart

ggplot(summary7_df, aes(x = rider_age,
                        y = total_trip_count,
                        fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Total number of trips by Age and User type",
       x = "Rider age",
       y = "Trip count",
       caption = "Year 2019") +
  scale_y_continuous(limits = c(0, 200000),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01))) +
  scale_x_continuous(limits = c(16, 95),
                     breaks = seq(15, 95, by = 5)) +
  facet_wrap(~user_type)
```




```
# Rider age having the highest trip count.

summary7_df %>%
  group_by(user_type) %>%
  filter(total_trip_count == max(total_trip_count)) %>%
  select(user_type, rider_age, total_trip_count)
```

```
## # A tibble: 2 x 3
## # Groups:   user_type [2]
##   user_type rider_age total_trip_count
##   <fct>         <dbl>            <int>
## 1 casual           25            24778
## 2 member           27           182150
```

```
# Average trip count for a rider of any age by User type :

head(summary7_df %>%
       group_by(user_type) %>%
       summarise(average_trip_count = mean(total_trip_count)) %>%
     select(user_type, average_trip_count))
```

```
## # A tibble: 2 x 2
##   user_type average_trip_count
##   <fct>                  <dbl>
## 1 casual                 5040.
## 2 member                41260.
```

(9A)

1. Casual riders : The Age range of 18 to 36 is where there are more than 80% of the Total trip counts in the Year 2019. Age of 25 is the rider age having the highest trip count ie; approx. 25k. The Average trip count of Casuals of any age is about just 5,000

2. Annual members : The Age range of 22 to 60 is where there are more than 80% of the Total trip counts in the Year 2019. Age of 27 is the rider age having the highest trip count ie; approx. 180k. The Average trip count of an annual member any age is about 41,000.

(1) Casual riders - Analysis summary of Trip count : Most Busy => Highest Trip count

Least Busy => Least Trip count

- Out of 4.23 Million trip counts, 0.92 Million trip counts are accounted by Casual members, ie; 22% of the Total trip count.
- May, June, July, August, September & October are the months with the highest total trip count for Casuals, with August as the maximum.
- In those months or in any month, Saturday & Sunday are the most busy days for Casuals.
- On Saturday or Sunday or any other day, 'Streeter Dr & Grand Axe' and the other 9 top stations are the Top 10 *Stations* with the highest trip count for Casuals, and 'Lake Shore Dr & Monroe St - Streeter Dr & Grand Ave' and the other 9 top routes are the Top 10 *Bike routes* with the highest total trip count for Casuals, where the Top 10 Busy stations and all the stations of the Top 10 busy routes are located near the Chicago lake side.
- In the Top 10 Stations & Routes or in all of the other stations & routes, 38% of the total trip count is that of Female Casual riders, and 62% of it is that of Male riders.
- Male or Female, the highest total trip counts for casual riders are in the age range of 19 to 36 with Age 25 as the rider age having the highest trip count.

Now, if we INTERSECT ( ) & UNION ( ) intelligently all the high leverage points, then we can target precisely a narrow group for the highest impact -

(May, June, July, August, September, October) *( )* Saturday and Sunday *( )* ('Streeter Dr & Grand Axe' & other Top 9 stations *( )* 'Lake Shore Dr & Monroe St - Streeter Dr & Grand Ave' & other Top 9 routes) *( )* Male & Female (both are significant) *( )* Age range 19-36.

(2) Annual members - Analysis summary of Trip count : Most Busy => Highest Trip count

Least Busy => Least Trip count

- Out of 4.23 Million trip counts, 3.3 Million trip counts are accounted by Annual members, ie; 78% of the Total trip count.
- May, June, July, August, September & October are the months with the highest total trip count for Annual members.
- In those months or in any month, Monday through Friday are the most busy days for Annual members with Tuesday as the maximum.
- On Monday through Friday or any other day, 'Canal St & Adams St' and the other 9 top stations are the Top 10 *Stations* with the highest trip count for Annual members, 'Canal St & Adams St - Michigan Ave & Washington St' and the other 9 top routes are the Top 10 *Bike routes* with the highest total trip count for Annual members, where the Top 10 Busy stations and all the stations of the Top 10 busy routes are concentrated away from the Chicago lake side.
- In the Top 10 Stations & Routes or in all of the other stations & routes, 25% of the total trip count is that of Female Annual members, and 75% of it is that of Male annual members.
- Male or Female, the highest total trip counts for casual riders are in the age range of 22 to 60 with Age 27 as the rider age having the highest trip count.

Now, if we INTERSECT ( ) & UNION ( )intelligently all the high leverage points, then we can target precisely a narrow group for the highest impact -

(May, June, July, August, September, October) *( )* Monday through Friday *( )* ('Canal St & Adams St' & other Top 9 stations *( )* 'Canal St & Adams St - Michigan Ave & Washington St' & other Top 9 routes) *( )* Male & Female (both are significant) *( )* Age range 22-60.

---

2. Trip duration - ***0) Trip duration by Usertype (2019 - 2020 Q1) :***

(a) **Total trip duration (2019 - 2020 Q1) (Unit : %) :**

```
# User Type-wise Contribution to Overall Trip Duration (%)

summary0_df_1 <- all_trips_19_20_10 %>%
  group_by(user_type) %>%
  summarise('percentage_trip_duration' = round(sum(trip_duration)/sum(all_trips_19_20_10$trip_duration)


# View the summary

head(summary0_df_1)
```

```
## # A tibble: 2 x 2
##   user_type percentage_trip_duration
##   <fct>                        <dbl>
## 1 casual                        46.3
## 2 member                        53.7
```

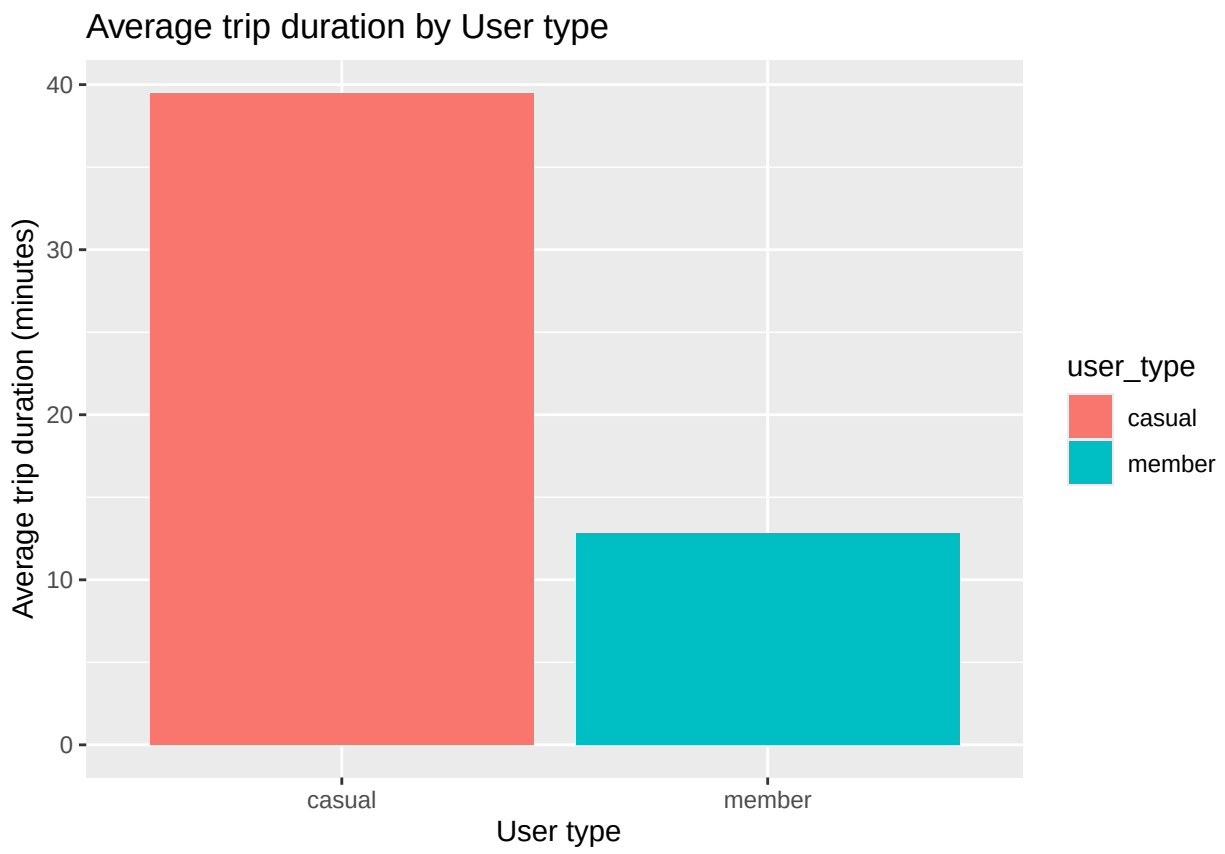
(b) **Average trip duration (2019 - 2020 Q1) (Unit : Minutes)**

```
# Average trip duration by User type

summary0_df_1avg <- all_trips_19_20_10 %>%
  group_by(user_type) %>%
  summarise('average_trip_duration_minute' = round(mean(trip_duration)/60, 1))

# View summary
head(summary0_df_1avg)
```

```
## # A tibble: 2 x 2
##   user_type average_trip_duration_minute
##   <fct>                             <dbl>
## 1 casual                             39.5
## 2 member                             12.8
```

```
# Plot

ggplot(data = summary0_df_1avg, mapping = aes(x = user_type,
                                              y = average_trip_duration_minute,
```

```
                                          fill = user_type)) +
  geom_col() +
  labs(title = "Average trip duration by User type",
       y = "Average trip duration (minutes)",
       x = " User type" )
```



(10A)

1. Casual riders : Total Trip duration is 54% of the Total Trip duration of all Users, while the average trip duration is 39.5 minutes.
2. Annual members : Total Trip duration is 46% of the Total Trip duration of all Users, while the average trip duration is 12.8 minutes.

1) Trip duration in a Month by User type (2019 - 2020 Q1) :

(a) **Total trip duration (2019 - 2020 Q1) (Unit : Days) :**

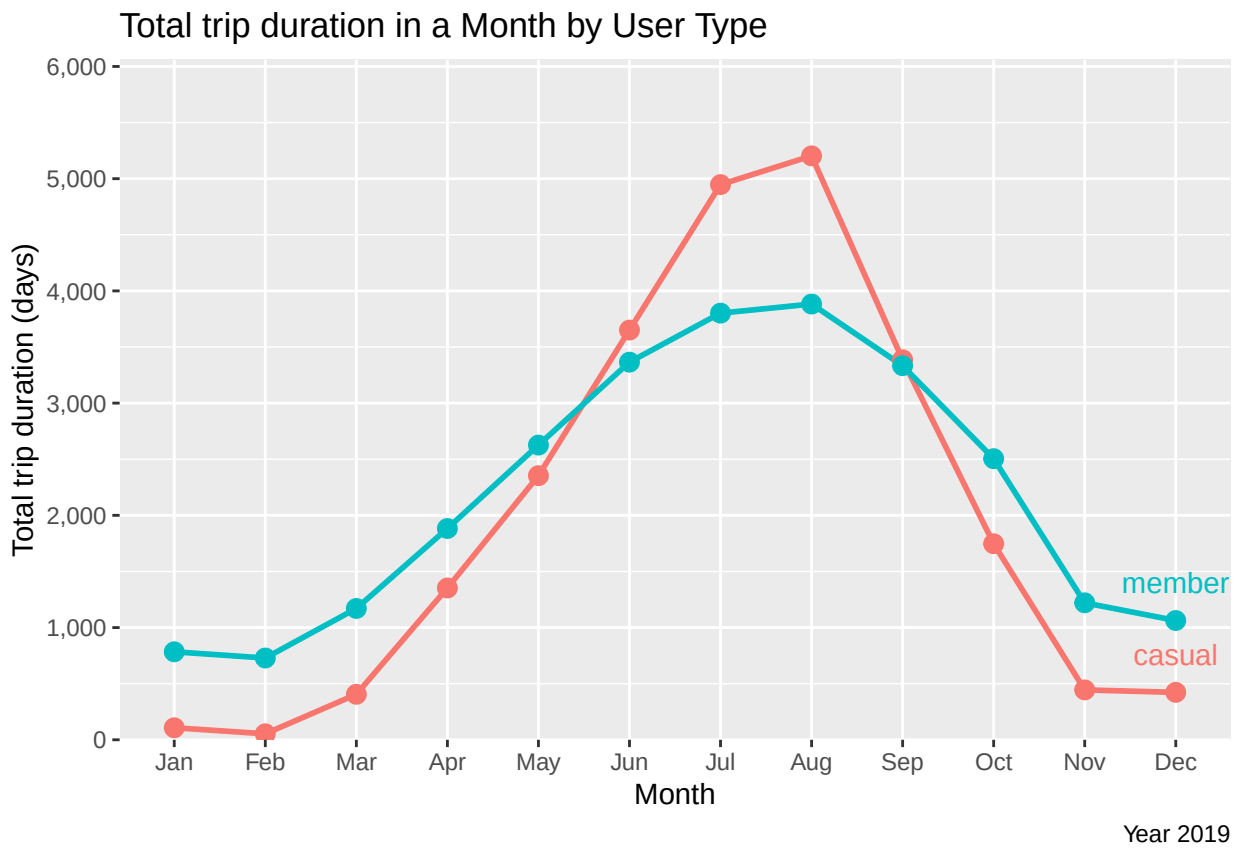
```
# Total Trip duration in a Month by User type - Summary

summary1_df_1 <- all_trips_19_20_10 %>%
  filter(year(start_time) == 2019) %>%
  group_by(month_name, user_type) %>%
  summarize(
    total_trip_duration = round(sum(trip_duration)/3600/24,1)
    )
```

```
## `summarise()` has grouped output by 'month_name'. You can override using the
## `.groups` argument.
```

```
# View Summary

glimpse(summary1_df_1 %>% pivot_wider(
    names_from = month_name,
    values_from = c(total_trip_duration)))
```

```
## Rows: 2
## Columns: 13
## $ user_type <fct> casual, member
## $ Jan       <dbl> 107.1, 783.9
## $ Feb       <dbl> 53.5, 728.6
## $ Mar       <dbl> 405.2, 1170.2
## $ Apr       <dbl> 1352.6, 1882.9
## $ May       <dbl> 2353.6, 2626.2
## $ Jun       <dbl> 3651.0, 3365.2
## $ Jul       <dbl> 4947.4, 3801.9
## $ Aug       <dbl> 5203.4, 3882.9
## $ Sep       <dbl> 3386.6, 3333.3
## $ Oct       <dbl> 1747.1, 2505.0
## $ Nov       <dbl> 444.4, 1220.5
## $ Dec       <dbl> 422.5, 1062.7
```

```
# Total trip duration (in days) inn each month by user type- Line Plot

label_data_1_1 <- summary1_df_1 %>%
  filter(month_name == "Dec")

ggplot(data = summary1_df_1,
       mapping = aes(x = month_name,
                     y = total_trip_duration,
                     colour = user_type,
                     group = user_type)) +
  geom_line(size =  1) +
  geom_point(size = 3) +
  labs(title = "Total trip duration in a Month by User Type",
       x = "Month",
       y = "Total trip duration (days)",
       caption = "Year 2019") +
  geom_text(data = label_data_1_1,
            aes(label = user_type),
            vjust = -1.3,
            size = 4,
            show.legend = FALSE) +
  scale_y_continuous(limits = c(0, 6000),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01))) +
  theme(legend.position = "none")
```



(b) **Average trip duration (2019 - 2020 Q1) (Unit : Minutes) :**

```
# Average trip duration (in minutes) in each month by User type - Summary


summary1_df_1avg <- all_trips_19_20_10 %>%
  group_by(month_name, user_type) %>%
  summarize(
    average_trip_duration_minute = round(mean(trip_duration)/60, 1)
    )
```

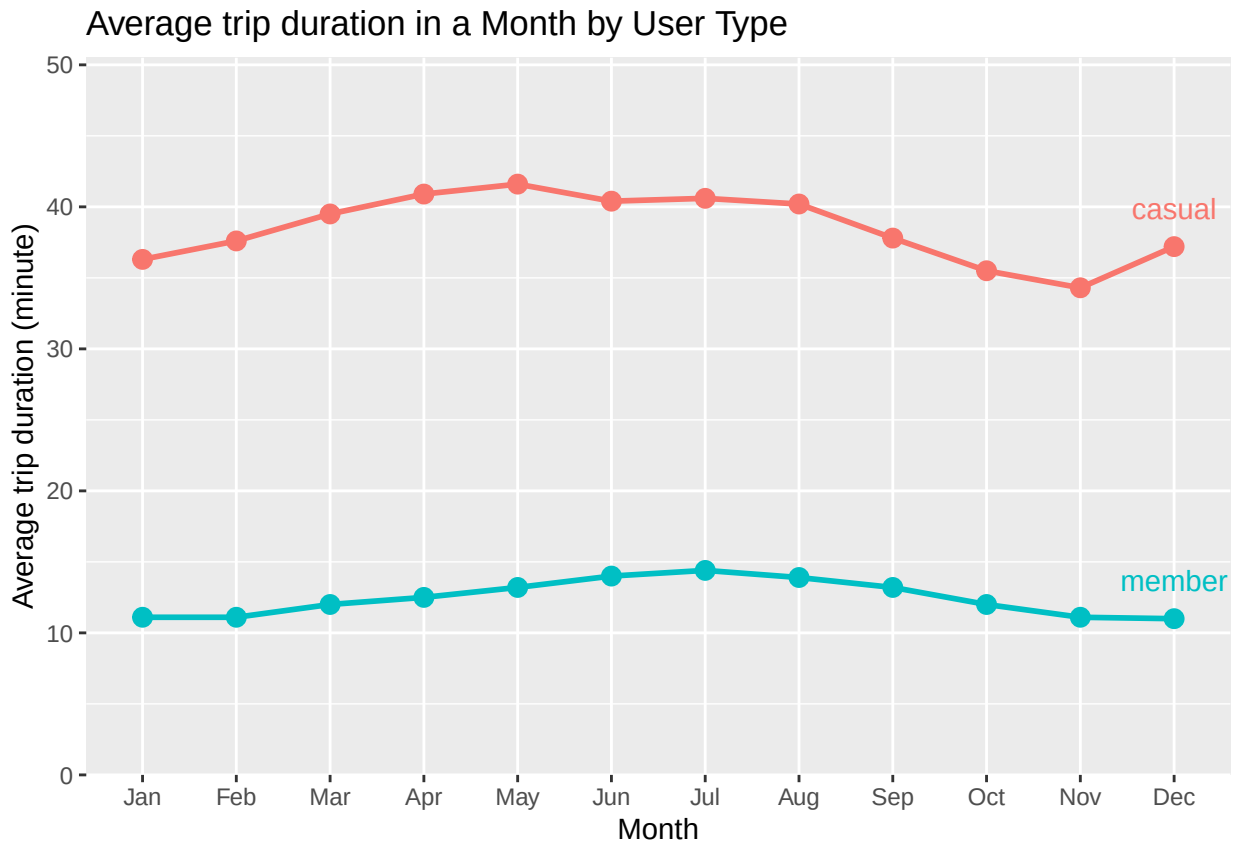
```
## `summarise()` has grouped output by 'month_name'. You can override using the
## `.groups` argument.
```

```
# View summary

glimpse(summary1_df_1avg %>% pivot_wider(
    names_from = month_name,
    values_from = c(average_trip_duration_minute)))
```

```
## Rows: 2
## Columns: 13
## $ user_type <fct> casual, member
## $ Jan       <dbl> 36.3, 11.1
```

```
## $ Feb       <dbl> 37.6, 11.1
## $ Mar       <dbl> 39.5, 12.0
## $ Apr       <dbl> 40.9, 12.5
## $ May       <dbl> 41.6, 13.2
## $ Jun       <dbl> 40.4, 14.0
## $ Jul       <dbl> 40.6, 14.4
## $ Aug       <dbl> 40.2, 13.9
## $ Sep       <dbl> 37.8, 13.2
## $ Oct       <dbl> 35.5, 12.0
## $ Nov       <dbl> 34.3, 11.1
## $ Dec       <dbl> 37.2, 11.0
```

```r
# Average trip duration (in minutes) in each month by User type - Line plot

label_data_1_1avg <- summary1_df_1avg %>%
  filter(month_name == "Dec")

ggplot(data = summary1_df_1avg,
       mapping = aes(x = month_name,
                     y = average_trip_duration_minute,
                     colour = user_type,
                     group = user_type)) +
  geom_line(size =  1) +
  geom_point(size = 3) +
  labs(title = "Average trip duration in a Month by User Type",
       x = "Month",
       y = "Average trip duration (minute)") +
  geom_text(data = label_data_1_1avg,
            aes(label = user_type),
            vjust = -1.3,
            size = 4,
            show.legend = FALSE) +
  scale_y_continuous(limits = c(0, 50),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01))) +
  theme(legend.position = "none")
```



(11A)

1. Casual riders :
    a) January and February are the least traveled months with February as the minimum (Trip duration of 53 days). Then after February, a steep and linear increase upto the Most busiest month August - a whopping 9700% increase from February to a count of approx. 5200 days.Then a linear and steep decline until November to a count of approx. 444 days. Then a slight linear decrease until December. Exceeding a total trip duration of 1500 days each, we have the months May, June, July, August, September, October.
    b) Average trip duration in each month doesn't vary substantially from the yearly average of 39.5 minutes. But it reaches above 40 minutes in April, May (maximum), June, July & August
2. Annual members :
    a) January and February are the least traveled months with February as the minimum (Trip duration of 728 days). Then after February, a linear increase upto the Most busiest month August - a 4300% increase from February to a duration of approx. 3880 days. Then a linear decrease until November and a slow decrease to December. We have the months May, June, July, August, September, October - where each has a total trip duration exceeding 2000 days.
    b) Average trip duration in each month doesn't vary substantially from the yearly average of 12.8 minutes. But it reaches above 13.5 minutes in June, July (maximum) & August.

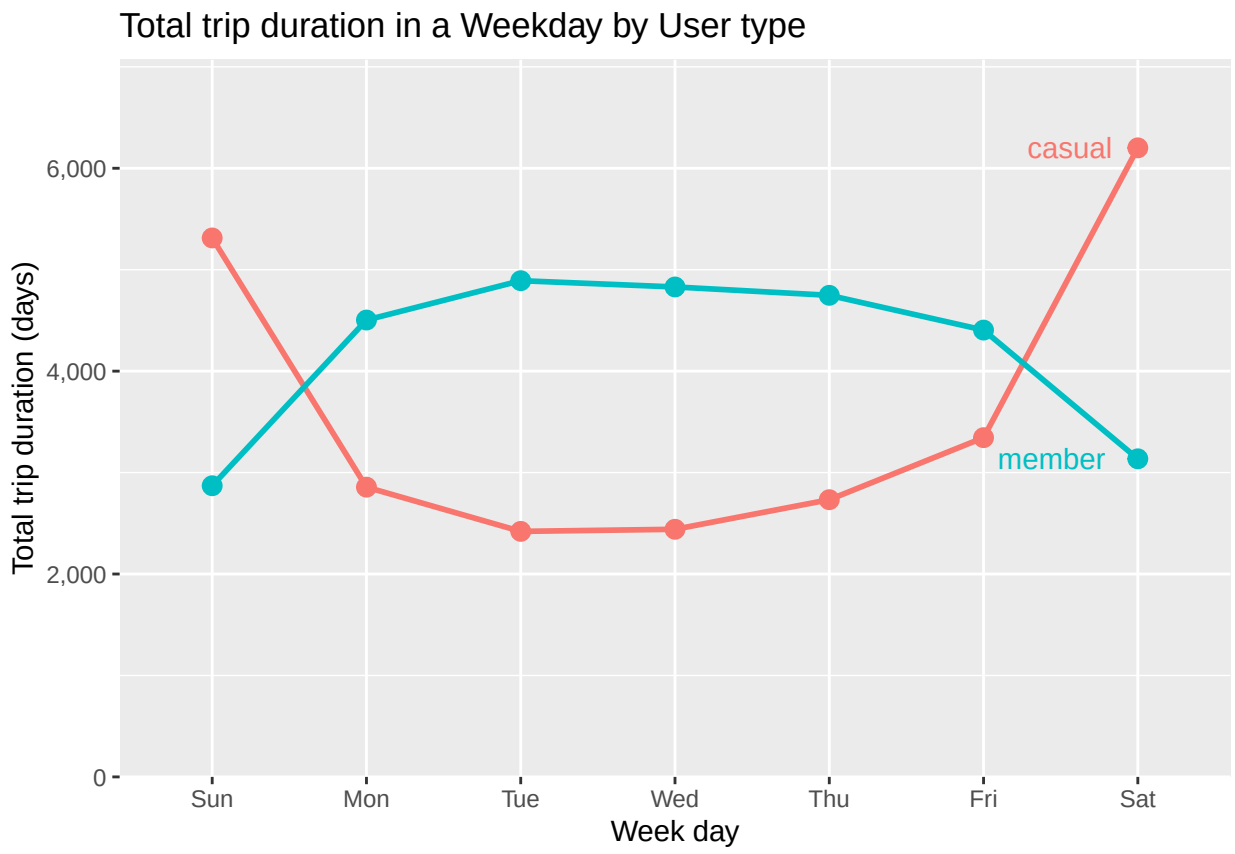
2) Trip duration in a Weekday by Usertype (2019 - 2020 Q1) :

(a) **Total trip duration (2019 - 2020 Q1) (Unit : Days) :**

```r
# Total trip duration in a Weekday by Usertype - Summary

summary2_df_1 <- all_trips_19_20_10 %>%
  group_by(week_day, user_type) %>%
  summarize(
    total_trip_duration = round(sum(trip_duration)/3600/24,1),
    .groups = "drop")

# Summary wide table view

head(summary2_df_1 %>% pivot_wider(
    names_from = week_day,
    values_from = c(total_trip_duration)))
```

```
## # A tibble: 2 x 8
##   user_type   Sun   Mon   Tue   Wed   Thu   Fri   Sat
##   <fct>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 casual    5313  2856. 2420. 2441. 2733. 3344. 6201.
## 2 member    2870. 4504. 4892. 4830. 4748. 4405. 3136.
```

```r
# Total trip duration in a Weekday by Usertype  - Line plot

label_data_2_1 <- summary2_df_1 %>%
  filter(week_day == "Sat")

ggplot(data = summary2_df_1,
       mapping = aes(x = week_day,
                     y = total_trip_duration,
                     colour = user_type,
                     group = user_type)) +
  geom_line(size =  1) +
  geom_point(size = 3) +
  labs(title = "Total trip duration in a Weekday by User type",
       x = "Week day",
       y = "Total trip duration (days)") +
  geom_text(data = label_data_2_1,
            aes(label = user_type),
            hjust = 1.3,
            size = 4,
            show.legend = FALSE) +
  scale_y_continuous(limits = c(0, 7000),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01))) +
  theme(legend.position = "none")
```



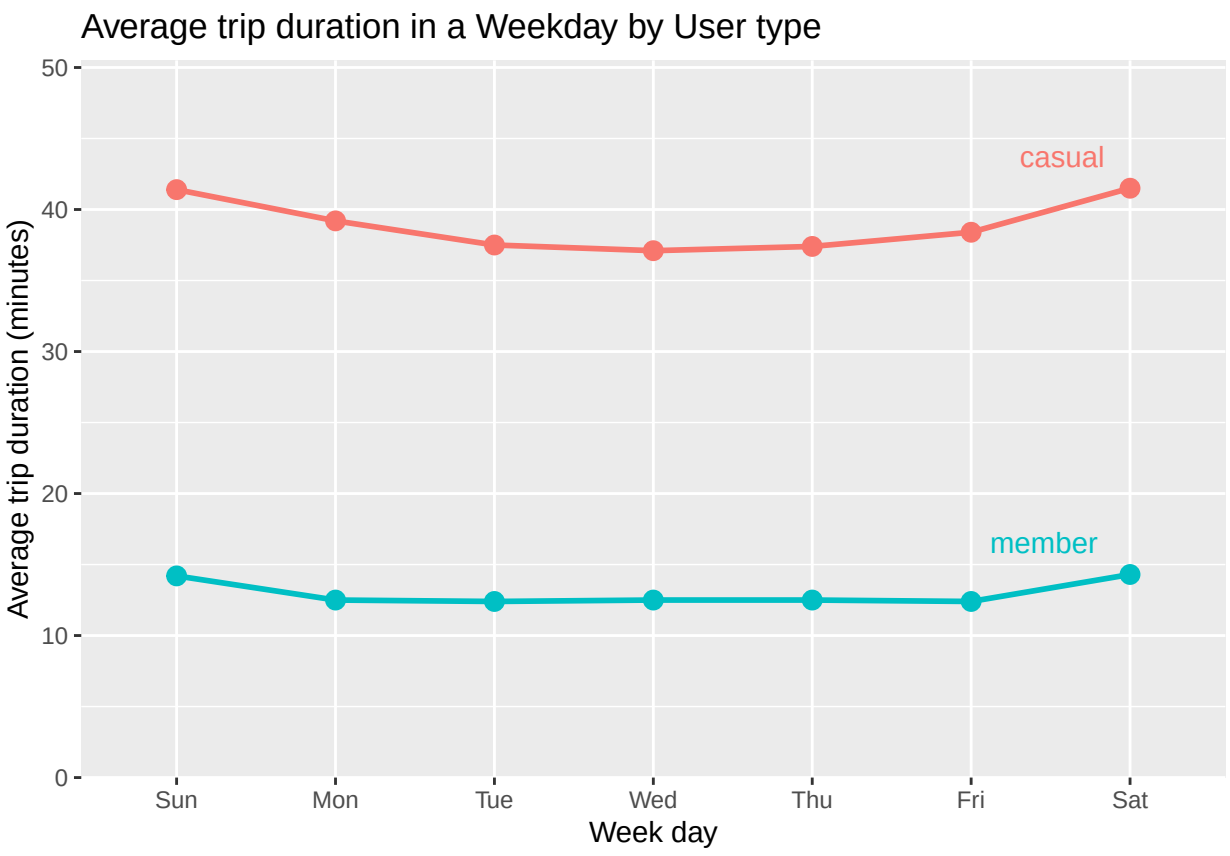
(b) **Average trip duration (2019 - 2020 Q1) (Unit : Minutes) :**

```
# Average trip duration in a Weekday by User type - Summary

summary2_df_1avg <- all_trips_19_20_10 %>%
  group_by(week_day, user_type) %>%
  summarize(
    average_trip_duration_minute = round(mean(trip_duration)/60, 1),
    .groups = "drop")

# Summary wide table view

head(summary2_df_1avg %>% pivot_wider(
    names_from = week_day,
    values_from = c(average_trip_duration_minute)))
```

```
## # A tibble: 2 x 8
##   user_type   Sun   Mon   Tue   Wed   Thu   Fri   Sat
##   <fct>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 casual     41.4  39.2  37.5  37.1  37.4  38.4  41.5
## 2 member     14.2  12.5  12.4  12.5  12.5  12.4  14.3
```

```
# Average trip duration in a Weekday by Usertype  - Line plot

label_data_2_1avg <- summary2_df_1avg %>%
```

```
  filter(week_day == "Sat")

ggplot(data = summary2_df_1avg,
       mapping = aes(x = week_day,
                     y = average_trip_duration_minute,
                     colour = user_type,
                     group = user_type)) +
  geom_line(size =  1) +
  geom_point(size = 3) +
  labs(title = "Average trip duration in a Weekday by User type",
       x = "Week day",
       y = "Average trip duration (minutes)") +
  geom_text(data = label_data_2_1avg,
            aes(label = user_type),
            hjust = 1.3,
            vjust = -1,
            size = 4,
            show.legend = FALSE) +
  scale_y_continuous(limits = c(0, 50),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01))) +
  theme(legend.position = "none")
```



(12A)

1. Casual riders :

a) Saturday is the day with the highest total trip duration (approx. 6200 days), then a slight decrease on Sunday followed by a steep descent on Monday until Tuesday to reach a minimum total trip duration of approx. 2400 days. then a slight climb until Friday.
b) Saturday and Sunday are the days with the highest trip duration per trip (approx. 14 minutes), then the average trip duration reduces slightly and stays at approx. 12.5 minutes for the rest of the week.

2. Annual members :

a) Saturday and Sunday are the days with the least total trip duration (approx. 3000 days), then it rises to an average of approx. 4500 days and stays there for the rest of the week.
b) Saturday and Sunday are the days with the highest trip duration per trip (approx. 14 minutes), then the average trip duration reduces slightly and stays at approx. 12.5 minutes for the rest of the week.

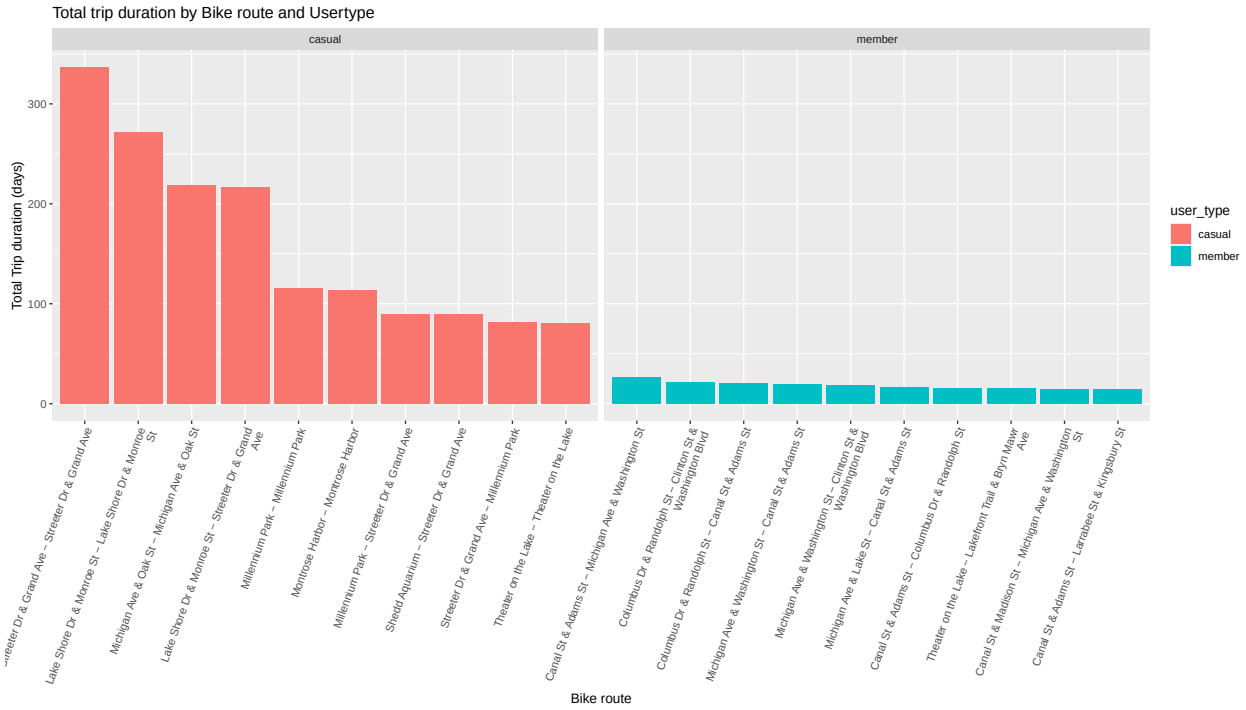
3) Trip duration by Bike route and User type (2019 - 2020 Q1) :

(a) **Total trip duration (2019 - 2020 Q1) (Unit : Days) :**

```
# Total trip duration by Bike route and Usertype : Top 10 Routes- Summary

summary5_df_1 <- all_trips_19_20_10 %>%
  group_by(bike_route, user_type) %>%
  summarize(
    total_trip_duration = round(sum(trip_duration)/3600/24, 1),
    .groups = "drop") %>%
  group_by(user_type) %>%
  slice_max(order_by = total_trip_duration, n = 10)

# Summary long table view

glimpse(summary5_df_1 %>% pivot_wider(
    names_from = user_type,
    values_from = c(total_trip_duration)) %>%
      arrange(desc(member)))
```

```
## Rows: 20
## Columns: 3
## $ bike_route <chr> "Canal St & Adams St - Michigan Ave & Washington St", "Colu~
## $ casual     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 337.1, 271.9, 218.8~
## $ member     <dbl> 26.2, 21.5, 20.8, 19.6, 18.8, 16.5, 15.6, 15.4, 15.0, 14.9,~
```

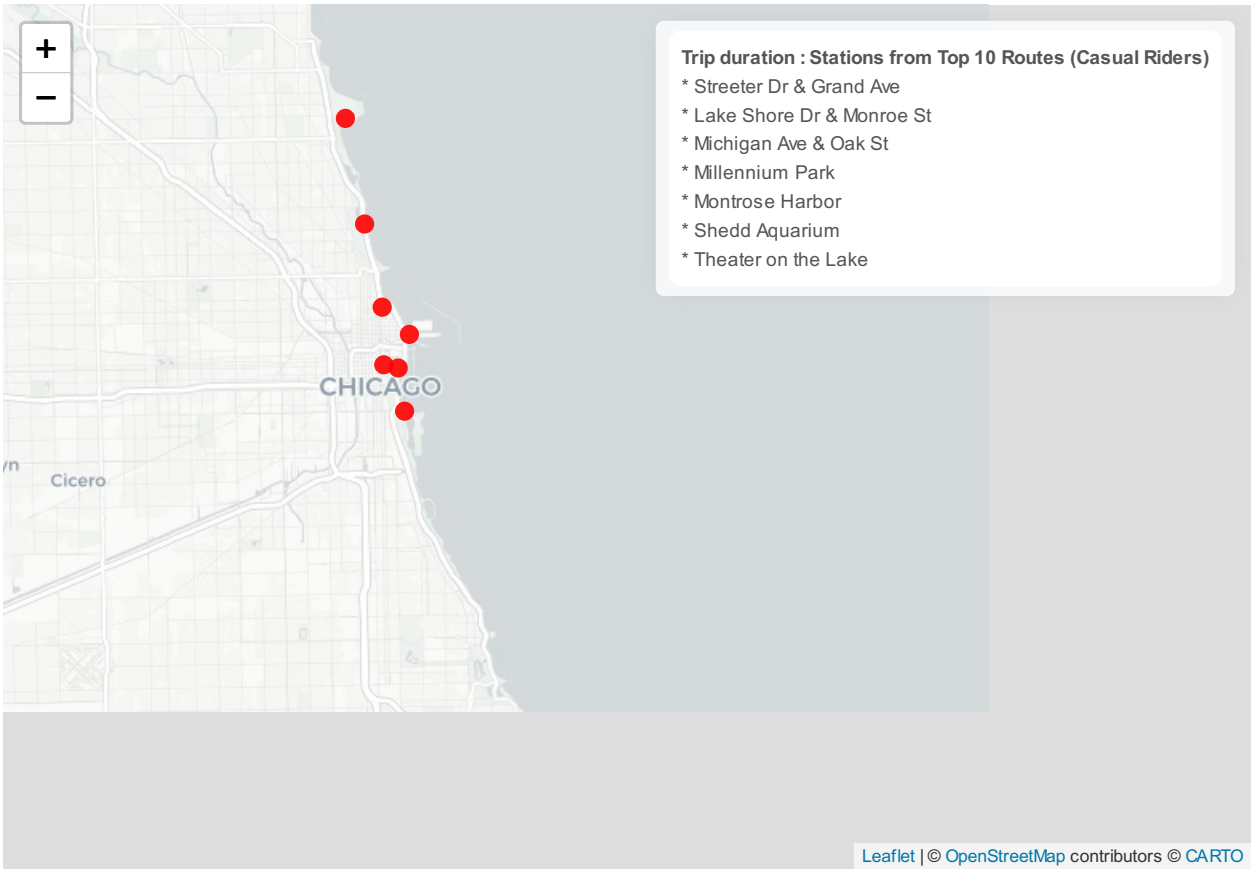
```
# Total trip duration by Bike route and Usertype : Top 10 Routes - Column Chart

ggplot(summary5_df_1, aes(x = reorder_within(str_wrap(bike_route, 50), -total_trip_duration, user_type)
                          y = total_trip_duration,
                          fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Total trip duration by Bike route and Usertype",
       x = "Bike route",
       y = "Total Trip duration (days)") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1))+
  scale_x_reordered() +
  facet_wrap(~user_type, scales = "free_x")
```



- **Total trip duration by routes for Casual riders -> MAP**

```
# Total trip duration by routes for Casual riders -> MAP

 ## Get top 10 routes for casual riders
top_routes_2 <- all_trips_19_20_10 %>%
  filter(user_type == "casual") %>%
  group_by(start_station_name, end_station_name) %>%
  summarise(total_trip_duration = sum(trip_duration), .groups = "drop") %>%
  arrange(desc(total_trip_duration)) %>%
  slice_head(n = 10)

## Extract start and end stations
start_stations_2 <- top_routes_2 %>%
  select(station_name = start_station_name)

end_stations_2 <- top_routes_2 %>%
  select(station_name = end_station_name)

## Combine and get unique station names
unique_stations_2 <- bind_rows(start_stations_2, end_stations_2) %>%
  distinct(station_name)

## Geocode unique station names (done using geocode function)
casual_routes_stations_duration <- read.csv("casual_top_routes_stations_durations_geocoded.csv")

## Create the map
map_trip_duration_casual_route <- leaflet(casual_routes_stations_duration) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(
```

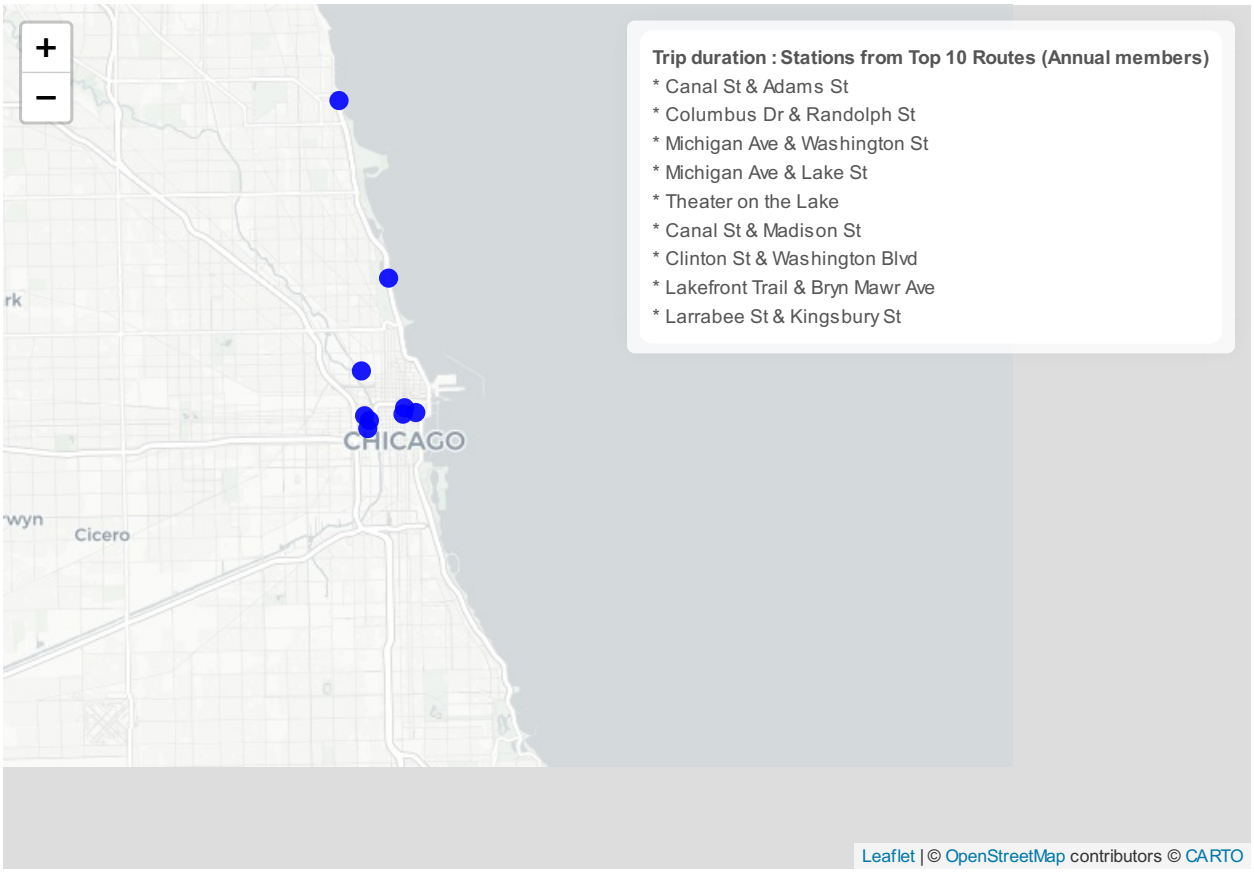
```
    ~longitude, ~latitude,
    radius = 6,
    color = "red",
    fillOpacity = 0.9,
    stroke = FALSE,
    popup = ~paste0(":</strong> ", station_name, "<br>"
    )
  ) %>%
  addControl(
    html = paste0(
      "<div style='text-align:left; padding:8px; font-size:12px; background:white; border-radius:8px;'>
      "<strong>Trip duration : Stations from Top 10 Routes (Casual Riders)</strong><br>",
      paste0(
        casual_routes_stations_duration %>%
          mutate(label = paste0(" * ", station_name)) %>%
          pull(label),
        collapse = "<br>"
      ),
      "</div>"
    ),
    position = "topright"
  )

map_trip_duration_casual_route
```



- **Total trip duration by routes for Annual members -> MAP**

```
# Total trip duration by routes for Annual members -> MAP

 ## Get top 10 routes for annual members
top_routes_3 <- all_trips_19_20_10 %>%
  filter(user_type == "member") %>%
  group_by(start_station_name, end_station_name) %>%
  summarise(total_trip_duration = sum(trip_duration), .groups = "drop") %>%
  arrange(desc(total_trip_duration)) %>%
  slice_head(n = 10)

## Extract start and end stations
start_stations_3 <- top_routes_3 %>%
  select(station_name = start_station_name)

end_stations_3 <- top_routes_3 %>%
  select(station_name = end_station_name)

## Combine and get unique station names
unique_stations_3 <- bind_rows(start_stations_3, end_stations_3) %>%
  distinct(station_name)

## Geocode unique station names (done using geocode function)
member_routes_stations_duration <- read.csv("member_top_routes_stations_durations_geocoded.csv")


## Create the map
map_trip_duration_member_route <- leaflet(member_routes_stations_duration) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(
    ~longitude, ~latitude,
    radius = 6,
    color = "blue",
    fillOpacity = 0.9,
    stroke = FALSE,
    popup = ~paste0(":</strong> ", station_name, "<br>"
    )
  ) %>%
  addControl(
    html = paste0(
      "<div style='text-align:left; padding:8px; font-size:12px; background:white; border-radius:8px;'>
      "<strong>Trip duration : Stations from Top 10 Routes (Annual members)</strong><br>",
      paste0(
        member_routes_stations_duration %>%
          mutate(label = paste0(" * ", station_name)) %>%
          pull(label),
        collapse = "<br>"
      ),
      "</div>"
    ),
    position = "topright"
  )
```

```
map_trip_duration_member_route
```



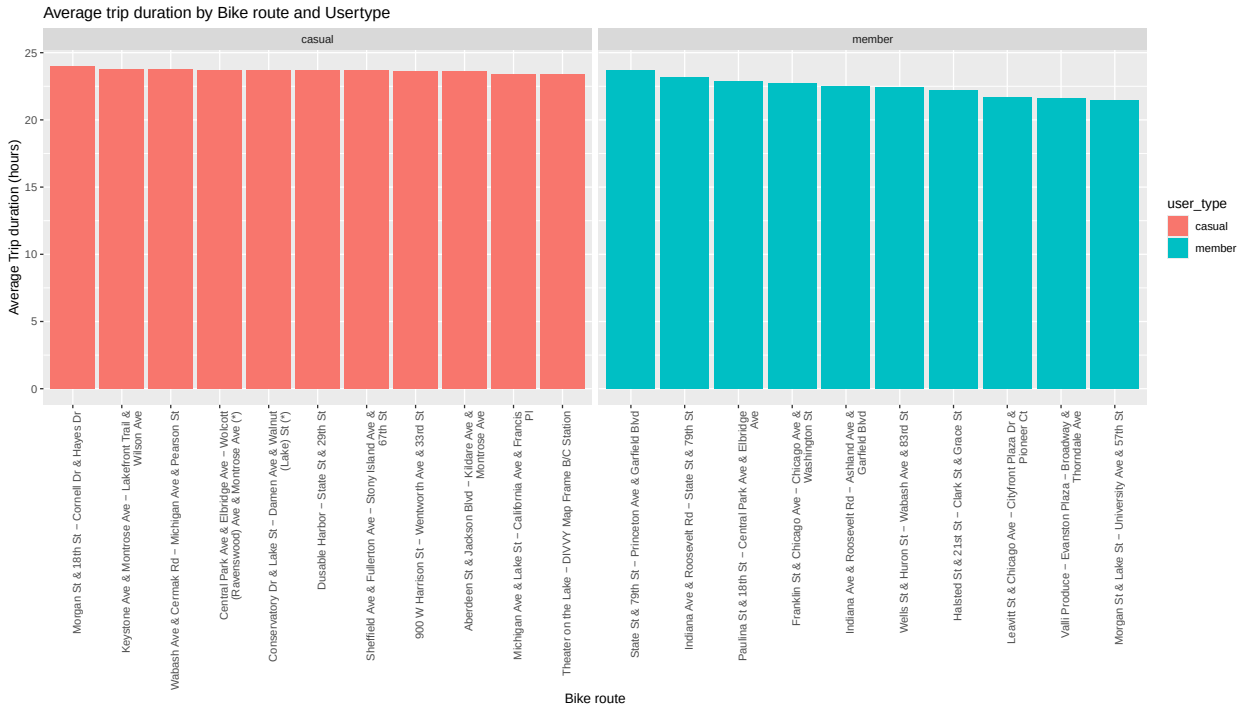
(b) **Average trip duration (2019 - 2020 Q1) (Unit : Hours) :**

```
# Average trip duration by Bike route and Usertype : Top 10 Routes- Summary

summary5_df_1avg <- all_trips_19_20_10 %>%
  group_by(bike_route, user_type) %>%
  summarize(
    average_trip_duration_hour = round(mean(trip_duration, na.rm = TRUE)/3600, 1),
    .groups = "drop") %>%
  group_by(user_type) %>%
  slice_max(order_by = average_trip_duration_hour, n = 10)

# Summary long table view

glimpse(summary5_df_1avg %>% pivot_wider(
    names_from = user_type,
    values_from = c(average_trip_duration_hour)) %>%
      arrange(desc(member)))
```

```
## Rows: 21
## Columns: 3
## $ bike_route <chr> "State St & 79th St - Princeton Ave & Garfield Blvd", "Indi~
```

```
## $ casual      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 24.0, 23.8, 23.8, 2~
## $ member      <dbl> 23.7, 23.2, 22.9, 22.7, 22.5, 22.4, 22.2, 21.7, 21.6, 21.5,~
```

```
# Average trip duration by Bike route and Usertype : Top 10 Routes - Column Chart

ggplot(summary5_df_1avg, aes(x = reorder_within(str_wrap(bike_route, 50), -average_trip_duration_hour, u
                         y = average_trip_duration_hour,
                         fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Average trip duration by Bike route and Usertype",
       x = "Bike route",
       y = "Average Trip duration (hours)") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  scale_x_reordered() +
  facet_wrap(~user_type, scales = "free_x")
```



(13A)

1. Casual riders : "Streeter Dr & Grand Ave - Streeter Dr & Grand Ave" and "Lake Shore Dr & Monroe St - Lake Shore Dr & Monroe St" has the highest total trip durations for Casual riders. But not even one of the Top 10 routes with highest average trip durations (approx. 23 hours) are in the routes with the Top 10 highest total trip durations. All stations of the Top 10 routes with the highest trip durations are located near the Lake side.

2. Annual members : "Canal St & Adams St - Michigan Ave & Washington St" has the highest total trip duration for Annual members. But not even one of the Top 10 routes with highest average trip durations (approx. 22 hours) are in the routes with the Top 10 highest total trip durations. All stations of the Top 10 routes with the highest trip durations are concentrated away from the Lake side.

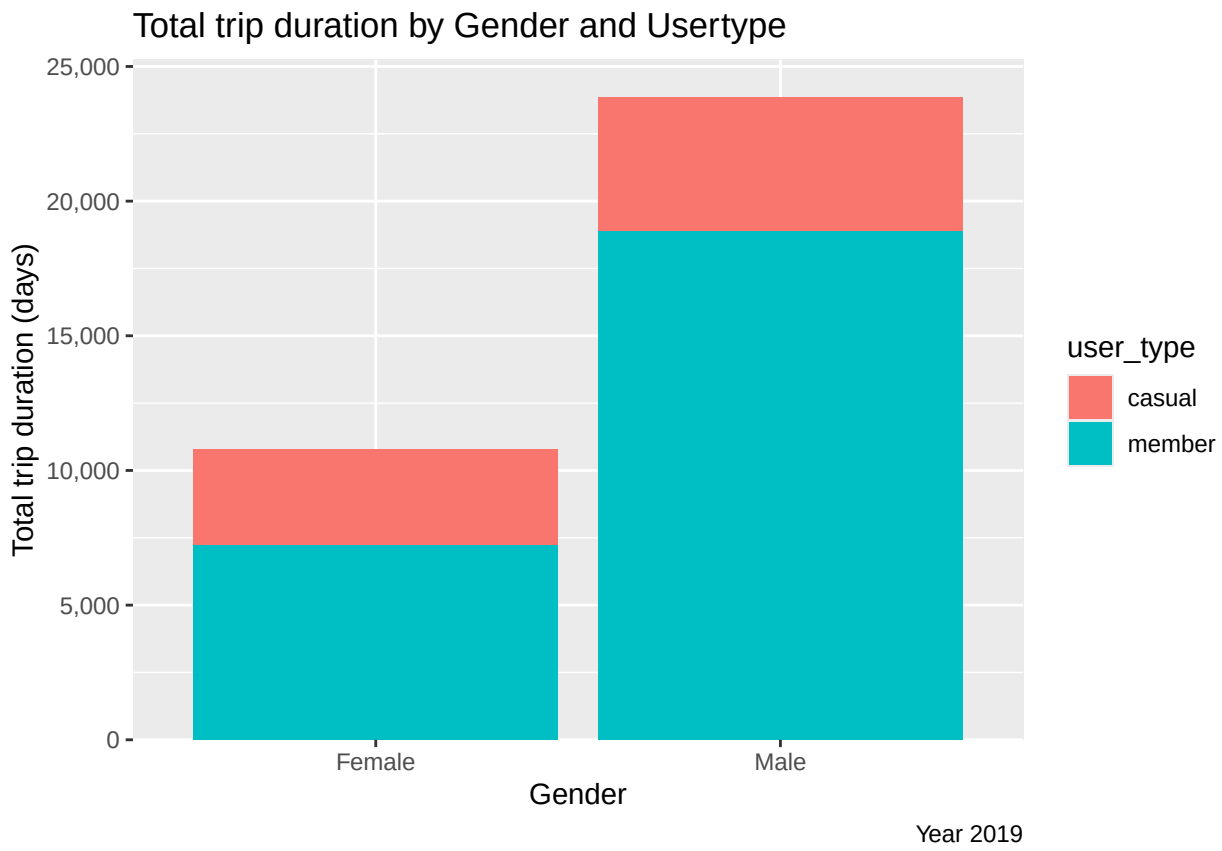
4) Trip duration by Gender and User type (2019 - 2020 Q1) :

(a) **Total trip duration (2019 - 2020 Q1) (Unit : Days) :**

```
# Total trip duration by Gender and Usertype - Summary

summary6_df_1 <- all_trips_19_20_10 %>%
  filter(is.na(gender) == FALSE) %>%
  group_by(gender, user_type) %>%
  summarize(
    total_trip_duration = round(sum(trip_duration)/3600/24, 1)
    , .groups = "drop"
    ) %>%
  group_by(user_type)

# Summary long table view

head(summary6_df_1 %>% pivot_wider(
    names_from = user_type,
    values_from = c(total_trip_duration)))
```

```
## # A tibble: 2 x 3
##   gender casual member
##   <fct>   <dbl>  <dbl>
## 1 Female  3546.  7239.
## 2 Male    4976. 18876.
```

```
# Total trip duration by Gender and User type - Column plot

ggplot(summary6_df_1, aes(x = gender,
                          y = total_trip_duration,
                          fill = user_type)) +
  geom_col(position = "stack") +
  labs(title = "Total trip duration by Gender and Usertype ",
       x = "Gender",
       y = "Total trip duration (days)",
       caption = "Year 2019") +
  scale_y_continuous(limits = c(0, 25000),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01)))
```

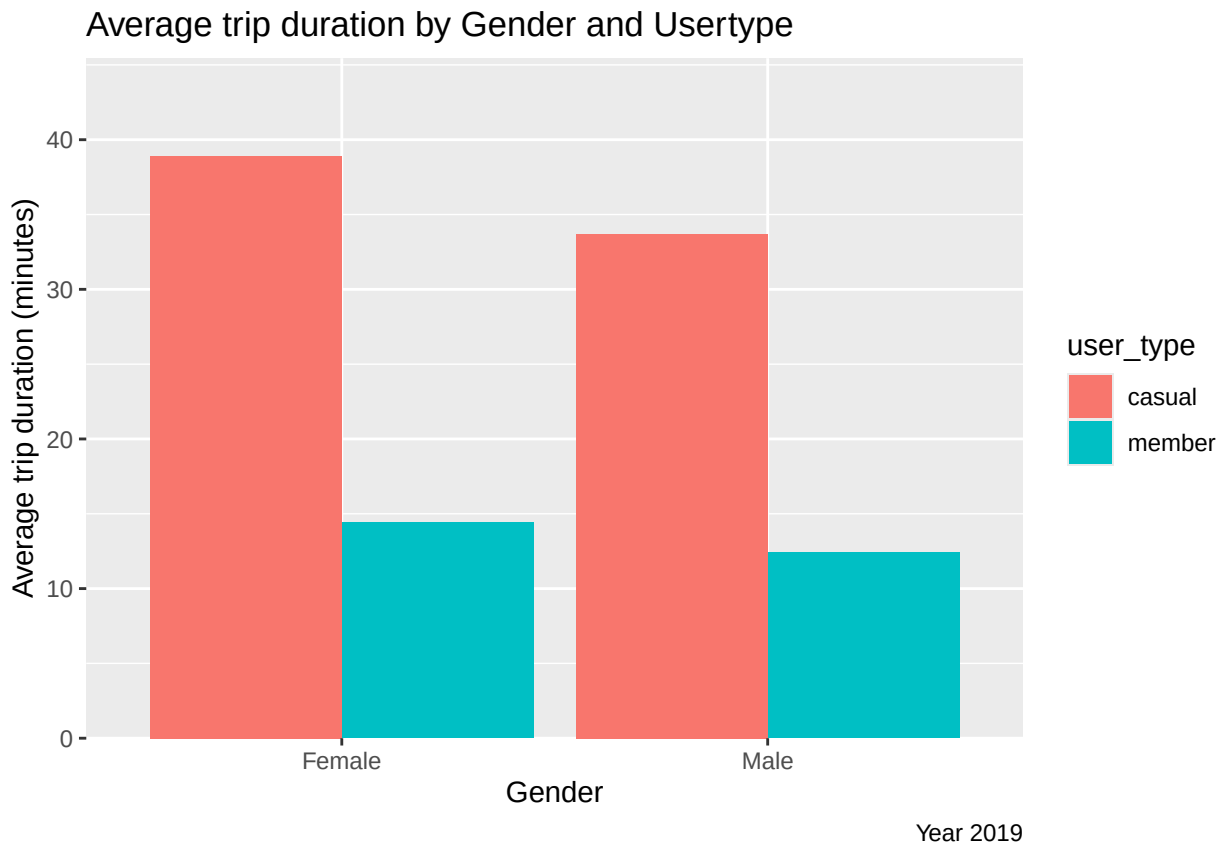


(b) **Average trip duration (2019 - 2020 Q1) (Unit : Minutes) :**

```
# Average trip duration by Gender and Usertype - Summary

summary6_df_1avg <- all_trips_19_20_10 %>%
  filter(is.na(gender) == FALSE) %>%
  group_by(gender, user_type) %>%
  summarize(
    average_trip_duration_minute = round(mean(trip_duration)/60, 1)
    , .groups = "drop"
    ) %>%
  group_by(user_type)

# Summary long table view

head(summary6_df_1avg %>% pivot_wider(
    names_from = user_type,
    values_from = c(average_trip_duration_minute)))
```

```
## # A tibble: 2 x 3
##   gender casual member
##   <fct>   <dbl>  <dbl>
## 1 Female   38.9   14.4
## 2 Male     33.7   12.4
```

```r
# Average trip duration by Gender and Usertype - Column plot

ggplot(summary6_df_1avg, aes(x = gender,
                         y = average_trip_duration_minute,
                         fill = user_type)) +
  geom_col(position = "dodge") +
  labs(title = "Average trip duration by Gender and Usertype",
       x = "Gender",
       y = "Average trip duration (minutes)",
       caption = "Year 2019") +
  scale_y_continuous(limits = c(0, 45),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01)))
```



(14A)

0. For Female riders, the Total trip duration is 10,784 days, while it is 23,850 days for Male riders in the Year 2019.

1. Casual riders : Males account for 58% of the total trip duration and Females account for 42%. But the Average trip duration of a Female rider (38.9 minutes) is 5 minutes greater than that of a Male rider.

2. Annual members : Males account for 72% of the total trip duration and Females account for 28%. But the Average trip duration of a Female rider (14.4 minutes) is 2 minutes greater than that of a Male rider.

3. Among Female riders, only 33% of total trip duration were accounted by casual riders, while a dominant 67% were by annual members.

4. Among Male riders, only 21% of total trip duration were accounted by casual riders, while a dominant 79% were by annual members.

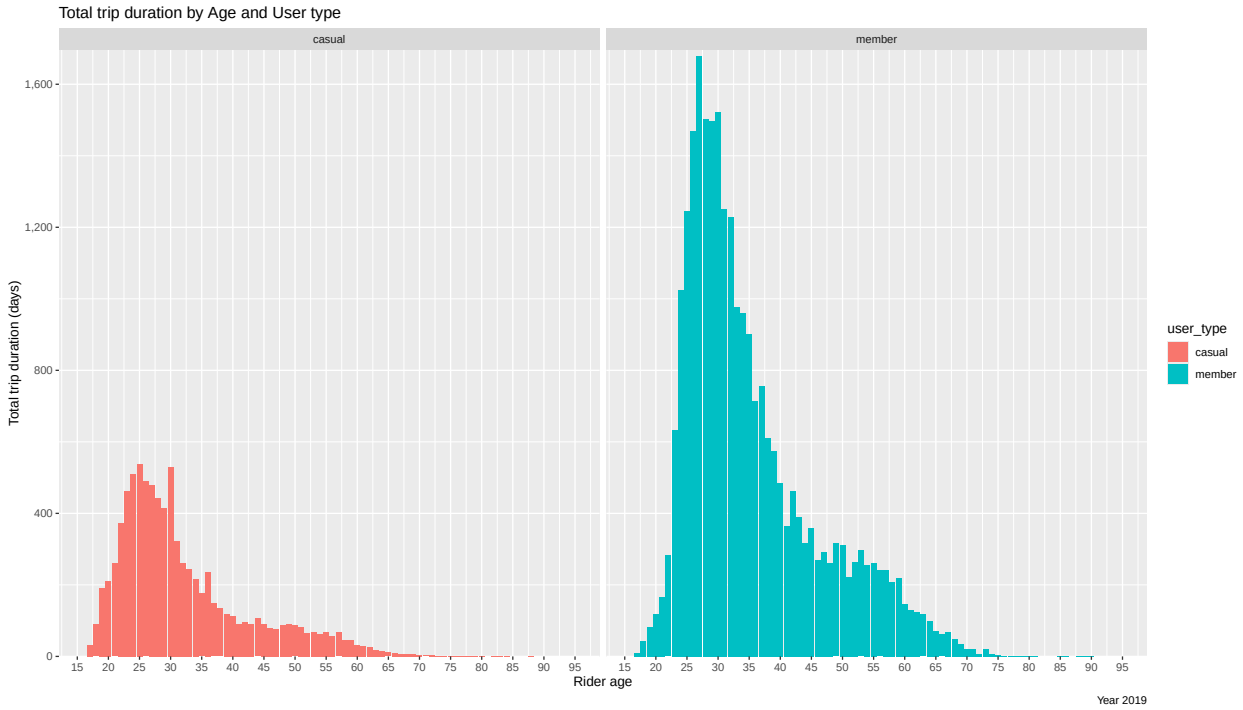
5) Trip duration by Age and User type :

(a) **Total trip duration (2019 - 2020 Q1) (Unit : Days) :**

```
# Total trip duration by Age and User type - Summary

summary7_df_1 <- all_trips_19_20_10 %>%
  filter(is.na(rider_age) == FALSE) %>%
  group_by(rider_age, user_type) %>%
  summarize(
    total_trip_duration = round(sum(trip_duration)/3600/24, 1),
    .groups = "drop"
    ) %>%
  group_by(user_type)

# Summary long table view

glimpse(summary7_df_1 %>% pivot_wider(
    names_from = user_type,
    values_from = c(total_trip_duration)) %>%
      arrange(rider_age))
```

```
## Rows: 74
## Columns: 3
## $ rider_age <dbl> 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, ~
## $ casual    <dbl> 8.2, 31.9, 89.3, 190.3, 211.9, 260.0, 372.9, 461.8, 509.8, 5~
## $ member    <dbl> 7.1, 10.2, 42.7, 82.9, 118.2, 165.3, 282.4, 634.2, 1023.9, 1~
```

```
# Total trip duration by Age and User type - Column Chart

ggplot(summary7_df_1, aes(x = rider_age,
                          y = total_trip_duration,
                          fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Total trip duration by Age and User type",
       x = "Rider age",
       y = "Total trip duration (days)",
       caption = "Year 2019") +
  scale_y_continuous(limits = c(0, NA),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01))) +
  scale_x_continuous(limits = c(16, 95),
                     breaks = seq(15, 95, by = 5)) +
  facet_wrap(~user_type)
```




```
# Rider age having the highest trip duration (Unit in days).

summary7_df_1 %>%
  group_by(user_type) %>%
  filter(total_trip_duration == max(total_trip_duration)) %>%
  select(user_type, rider_age, total_trip_duration)
```

```
## # A tibble: 2 x 3
## # Groups:   user_type [2]
##   user_type rider_age total_trip_duration
##   <fct>         <dbl>               <dbl>
## 1 casual           25                539.
## 2 member           27               1679.
```

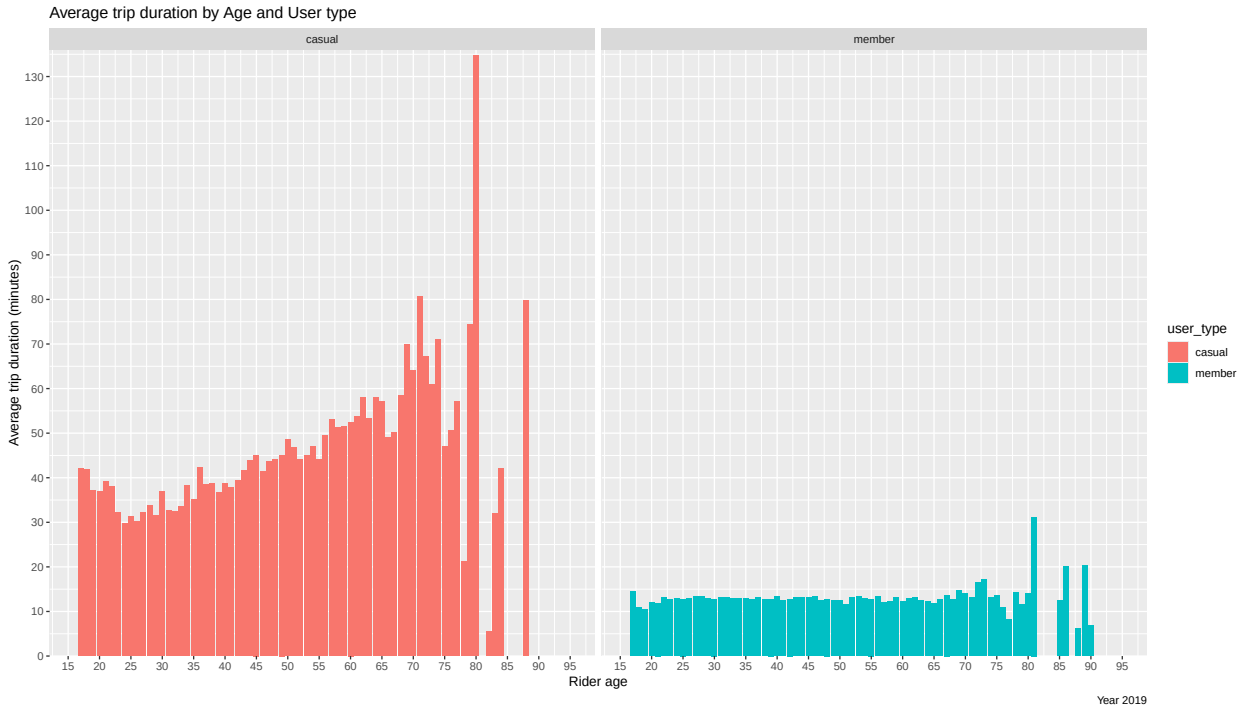
(b) **Average trip duration (2019 - 2020 Q1) (Unit : Minutes) :**

```
# Average trip duration by Age and User type - Summary

summary7_df_1avg <- all_trips_19_20_10 %>%
  filter(is.na(rider_age) == FALSE) %>%
  group_by(rider_age, user_type) %>%
  summarize(
    average_trip_duration_minute = round(mean(trip_duration)/60, 1),
    .groups = "drop"
    ) %>%
  group_by(user_type)

# Summary long table view
```

```
glimpse(summary7_df_1avg %>% pivot_wider(
    names_from = user_type,
    values_from = c(average_trip_duration_minute)) %>%
      arrange(rider_age))
```

```
## Rows: 74
## Columns: 3
## $ rider_age <dbl> 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, ~
## $ casual    <dbl> 40.4, 42.0, 41.8, 37.2, 37.0, 39.2, 38.1, 32.3, 29.8, 31.3, ~
## $ member    <dbl> 21.8, 14.5, 10.9, 10.5, 12.1, 11.9, 13.2, 12.7, 13.0, 12.8, ~
```

```
# Average trip duration by Age and User type - Column Chart

ggplot(summary7_df_1avg, aes(x = rider_age,
                        y = average_trip_duration_minute,
                        fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Average trip duration by Age and User type",
       x = "Rider age",
       y = "Average trip duration (minutes)",
       caption = "Year 2019") +
  scale_y_continuous(limits = c(0, NA),
                     labels = scales::comma,
                     expand = expansion(mult = c(0, 0.01)),
                     breaks = seq(0, 150, by = 10)) +
  scale_x_continuous(limits = c(16, 95),
                     breaks = seq(15, 95, by = 5)) +
  facet_wrap(~user_type)
```



```r
# Average of the average trip duration by Age and User type

summary7_df_1avg  %>%
  group_by(user_type) %>%
  summarise(average_trip_duration_minute_avg = round(mean(average_trip_duration_minute), 1))
```

```
## # A tibble: 2 x 2
##   user_type average_trip_duration_minute_avg
##   <fct>                                <dbl>
## 1 casual                                46.5
## 2 member                                13.3
```

(15A)

1. Casual riders :
    a) The age range of 18 to 51 is where the Total trip duration exceeds 80% of the Total trip duration of the whole age range in the Year 2019. Age of 25 is the rider age having the highest trip duration ie; approx. 538 days.
    b) The Average trip duration decreases from the age of 16 to the age of 24 (29.8 minutes). Then it gradually increases until the age of 75, and then there are noticeable spikes at the age of 80 and 87. Age ranges 16-22, 34-76, 79-80 have the highest average trip durations. Average trip duration is lowest in the age range of 23-33
2. Annual members :
    a) The Age range of 22 to 60 is where the Total trip duration exceeds 80% of the Total trip duration of the whole age range in the Year 2019. Age of 27 is the rider age having the highest trip duration ie; approx. 1679 days.
    b) The Average trip duration stays constant from the age of 16 to the age of 80, at an average of 13.3 minutes. Then a noticeable spike at the age of 81.

(1) Casual riders - Analysis summary of Trip duration :

- Casual riders contribute to 54% of the Total trip duration of all users, and have average trip duration of 39.5 minutes.
- May, June, July, August, September & October are the months with the highest total trip durations for Casuals with August as the Maximum, where April, May, June, July and August have the highest average trip duration.
- In those months or in any other months, Saturday & Sunday are the days with the highest total trip duration and highest average trip duration for Casuals. On Saturday or Sunday or any other day, "Streeter Dr & Grand Ave - Streeter Dr & Grand Ave" and "Lake Shore Dr & Monroe St - Lake Shore Dr & Monroe St" and the other 8 top routes are the Top 10 *Bike routes* with the highest total trip duration for Casuals, where all stations of the Top 10 routes with the highest trip durations are located near the Chicago lake side.
- In the Top 10 Routes or in all of the other routes, Male Casual riders account for 58% of the total trip duration and Females account for 42%, while Female casual riders ride 5 minutes more than Males per trip.
- Male or Female, casual riders with the highest trip durations are in the age range of 18 to 51 with Age 25 as the rider age having the highest total trip duration, while the average trip duration is highest for the age ranges 16-22 and 34-76.

Now, if we INTERSECT ( ) & UNION ( ) intelligently all the high leverage points, then we can target precisely a narrow group for the highest impact -

(May, June, July, August, September, October) *( )* Saturday and Sunday *( )* 'Streeter Dr & Grand Ave - Streeter Dr & Grand Ave' & other Top 9 routes *( )* Male & Female (both are significant) *( )* Age range 18-51.

(2) Annual riders - Analysis summary of Trip duration :

- Annual riders contribute to 46% of the Total trip duration of all users, and have average trip duration of 12.8 minutes.
- May, June, July, August, September & October are the months with the highest total trip durations for Annual members with August as the Maximum, where June, July and August have the highest average trip duration.
- In those months or in any other months, the days of highest total trip durations are from Monday to Friday, while Saturday & Sunday are the days with the highest average trip duration for Annual members.
- On a day, "Canal St & Adams St - Michigan Ave & Washington St" and the other 9 top routes are the Top 10 *Bike routes* with the highest total trip duration for Annual members, where all stations of the Top 10 routes with the highest trip durations are concentrated away from the Chicago Lake side.
- In the Top 10 Routes or in all of the other routes, Male annual members account for 72% of the total trip duration and Females account for 28%, while Female annual members ride 2 minutes more than Males per trip.
- Male or Female, Annual members with the highest trip durations are in the age range of 22 to 60 with Age 27 as the rider age having the highest total trip duration, while the average trip duration stays approximately constant in the age range of 16-80.

Now, if we INTERSECT ( ) & UNION ( ) intelligently all the high leverage points, then we can target precisely a narrow group for the highest impact -

(May, June, July, August, September, October) *( )* Monday through Friday *( )* 'Canal St & Adams St - Michigan Ave & Washington St' & other Top 9 routes *( )* Male & Female (both are significant) *( )* Age range 22-60.

Total Summary of Analysis :

Note :

Most Busy => Highest total trip count

Least Busy => Lowest total trip count

1.

- Casual riders : Ride for 39 minutes on an average.
- Annual members : Ride an average of 12.8 minutes

2.

- Casual riders : Saturday is the most busy day, while Tuesday is the least busy day
- Annual members : Tuesday is the most busy day, while Sunday is the least busy day

Trip count by User type -

3.

- Casual riders : Trip count is approx. 0.92 Million, which is 22% of the Total trip count.
- Annual members : Trip count is approx. 3.3 Million, which is 78% of the Total trip count.

4.

- Casual riders : January and February are the least busy months with February as the minimum (Below the count of 5,000). Then after February, a linear increase upto the Most busiest month August - a 5900% increase from February to a count of approx. 185,000.Then a linear decrease until November to a count of approx. 18,000. Then a slight linear decrease until December. Trip counts exceeded 50,000 in May, June, July, August, September, and October
- Annual members : January and February are the least busy months with February as the minimum (Below the count of 10,000). Then after February, a linear increase upto the Most busiest month August - a 3900% increase from February to a count of approx. 400,000. Then a linear decrease until October, and a sudden drop in count by 50% in November. Then a linear decrease until December. Trip counts exceeded 250,000 in May, June, July, August, September, and October.

5.

- Casual riders : Saturday and Sunday has a count of approx. 200,000. Then, the count reduces sharply by 35% through Monday ie; below 130,000 count and reaches a Minimum at Tuesday. Then it slowly and steadily increase until Friday, and then a sharp increase on Saturday.
- Annual members : Sunday is the least busy day (approx. 300,000), then the count sharply increase by 70% on Monday ie; above 500,000 count, and then reaches a Maximum on Tuesday. Then a slow & steady decrease until Friday, but the count stays above 500,000. Then a sharp decrease by Saturday to an approx. count of 300,000.

6.

- Casual riders : "Streeter Dr & Grand Axe" is the most busiest station for Casual riders. The Top 10 Busy stations are located near the Lake side
- Annual members : "Canal St & Adams St" is the most busiest station for Annual members. The Top 10 Busy stations are concentrated away from the Lake side

7.

- *When Starting station and Ending station are same, then it means that the user went for a ride from the starting station, and after the ride, finished the ride at the same station*
- Casual riders : "Lake Shore Dr & Monroe St - Streeter Dr & Grand Ave" is the most busiest route for Casual riders. All stations of the Top 10 busy routes are located near the lake side.
- Annual members : "Canal St & Adams St - Michigan Ave & Washington St" is the most busiest route for Annual members. All stations of the Top 10 busy routes are concentrated away from the lake side.

8.

- There are total 0.85 Million Trip counts for Female riders and 2.4 Million of that for Male riders in the Year 2019
- Casual riders : Among casual riders, 38% of trips were made by females, while 62% were made by males.

- Annual members : Among annual members, 25% of trips were made by females, whereas 75% were made by males.
- Female riders :Among female riders, only 15% of total trips were made by casual riders, while a dominant 85% were taken by annual members
- Male riders : Among male riders, only 8.8% of total trips were made by casual riders, while a dominant 91.2% were taken by annual members

9.

- Casual riders : The Age range of 18 to 36 is where there are more than 80% of the Total trip counts in the Year 2019. Age of 25 is the rider age having the highest trip count ie; approx. 25k. The Average trip count of Casuals of any age is about just 5,000.
- Annual members : The Age range of 22 to 60 is where there are more than 80% of the Total trip counts in the Year 2019. Age of 27 is the rider age having the highest trip count ie; approx. 180k. The Average trip count of an annual member any age is about 41,000.

Trip duration by User type -

10.

- Casual riders : Total Trip duration is 54% of the Total Trip duration of all Users, while the average trip duration is 39.5 minutes.
- Annual members : Total Trip duration is 46% of the Total Trip duration of all Users, while the average trip duration is 12.8 minutes.

11.

- Casual riders :
  - January and February are the least traveled months with February as the minimum (Trip duration of 53 days). Then after February, a steep and linear increase upto the Most busiest month August - a whopping 9700% increase from February to a count of approx. 5200 days.Then a linear and steep decline until November to a count of approx. 444 days. Then a slight linear decrease until December. Exceeding a total trip duration of 1500 days each, we have the months May, June, July, August, September, October
  - Average trip duration in each month doesn't vary substantially from the yearly average of 39.5 minutes. But it reaches above 40 minutes in April, May (maximum), June, July & August
- Annual members :
  - January and February are the least traveled months with February as the minimum (Trip duration of 728 days). Then after February, a linear increase upto the Most busiest month August - a 4300% increase from February to a duration of approx. 3880 days. Then a linear decrease until November and a slow decrease to December. We have the months May, June, July, August, September, October - where each has a total trip duration exceeding 2000 days.
  - Average trip duration in each month doesn't vary substantially from the yearly average of 12.8 minutes. But it reaches above 13.5 minutes in June, July (maximum) & August

12.

- Casual riders :
  - Saturday is the day with the highest total trip duration (approx. 6200 days), then a slight decrease on Sunday followed by a steep descent on Monday until Tuesday to reach a minimum total trip duration of approx. 2400 days. then a slight climb until Friday.

  - Saturday and Sunday are the days with the highest trip duration per trip (approx. 14 minutes), then the average trip duration reduces slightly and stays at approx. 12.5 minutes for the rest of the week.

- Annual members :
  - Saturday and Sunday are the days with the least total trip duration (approx. 3000 days), then it rises to an average of approx. 4500 days and stays there for the rest of the week**
  - Saturday and Sunday are the days with the highest trip duration per trip (approx. 14 minutes), then the average trip duration reduces slightly and stays at approx. 12.5 minutes for the rest of the week.

13.

- Casual riders : "Streeter Dr & Grand Ave - Streeter Dr & Grand Ave" and "Lake Shore Dr & Monroe St - Lake Shore Dr & Monroe St" has the highest total trip durations for Casual riders. But not even one of the Top 10 routes with highest average trip durations (approx. 23 hours) are in the routes with the Top 10 highest total trip durations. All stations of the Top 10 routes with the highest trip durations are located near the Lake side.
- Annual members : "Canal St & Adams St - Michigan Ave & Washington St" has the highest total trip duration for Annual members. But not even one of the Top 10 routes with highest average trip durations (approx. 22 hours) are in the routes with the Top 10 highest total trip durations. All stations of the Top 10 routes with the highest trip durations are concentrated away from the Lake side.

14.

- For Female riders, the Total trip duration is 10,784 days, while it is 23,850 days for Male riders in the Year 2019
- Casual riders : Males account for 58% of the total trip duration and Females account for 42%. But the Average trip duration of a Female rider (38.9 minutes) is 5 minutes greater than that of a Male rider.
- Annual members : Males account for 72% of the total trip duration and Females account for 28%. But the Average trip duration of a Female rider (14.4 minutes) is 2 minutes greater than that of a Male rider.
- Among Female riders, only 33% of total trip duration were accounted by casual riders, while a dominant 67% were by annual members
- Among Male riders, only 21% of total trip duration were accounted by casual riders, while a dominant 79% were by annual members

15.

- Casual riders :
  - The age range of 18 to 51 is where the Total trip duration exceeds 80% of the Total trip duration of the whole age range in the Year 2019. Age of 25 is the rider age having the highest trip duration ie; approx. 538 days.
  - The Average trip duration decreases from the age of 16 to the age of 24 (29.8 minutes). Then it gradually increases until the age of 75, and then there are noticeable spikes at the age of 80 and 87. Age ranges 16-22, 34-76, 79-80 have the highest average trip durations. Average trip duration is lowest in the age range of 23-33.

- Annual members :
  - The Age range of 22 to 60 is where the Total trip duration exceeds 80% of the Total trip duration of the whole age range in the Year 2019. Age of 27 is the rider age having the highest trip duration ie; approx. 1679 days.
  - The Average trip duration stays constant from the age of 16 to the age of 80, at an average of 13.3 minutes. Then a noticeable spike at the age of 81.

---

5. SHARE PHASE (Deliverable = Final report)

Final slideshow report :

The Final report is created in a clear and easy to understand manner for the stakeholders - specifically, the marketing executives - by distilling the key insights from the analysis and providing actionable recommendations.

Final slideshow report can be viewed here : Click Here to view the Final Slideshow Report

Additional Visualizations generated :

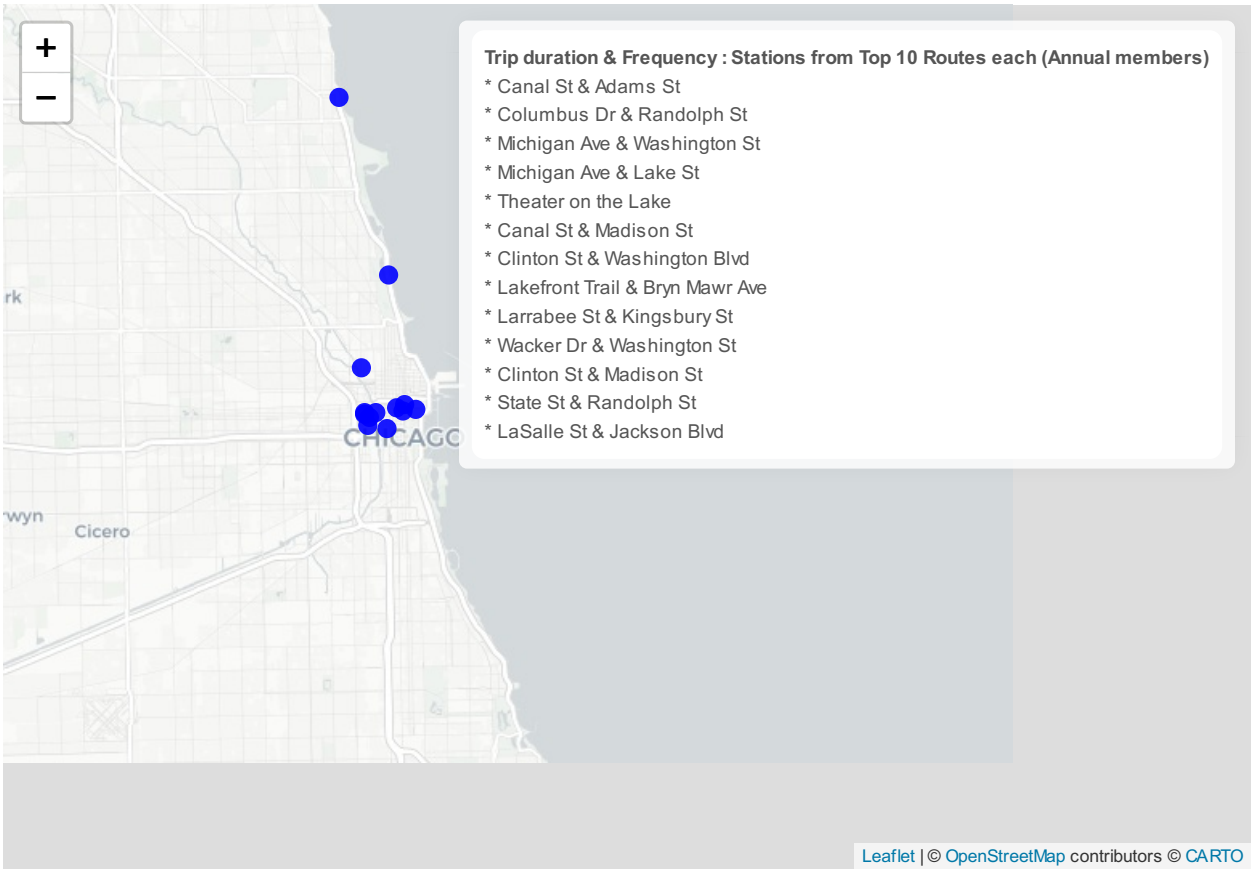
1) **Trip duration & Frequency : Stations from Top 10 Routes each (Annual members)**

```
member_duration <- read.csv("member_top_routes_stations_durations_geocoded.csv")
member_trip <- read.csv("member_top_routes_stations_trips_geocoded.csv")

member_top_routes_stations_duration_trip <- bind_rows(member_duration, member_trip) %>%
  distinct(station_name, .keep_all = TRUE)

member_top_routes_stations_duration_trip
```

```
##                       station_name latitude longitude
## 1              Canal St & Adams St 41.87925 -87.64001
## 2        Columbus Dr & Randolph St 41.88461 -87.61953
## 3     Michigan Ave & Washington St 41.88388 -87.62469
## 4           Michigan Ave & Lake St 41.88605 -87.62435
## 5              Theater on the Lake 41.92718 -87.63074
## 6            Canal St & Madison St 41.88165 -87.63957
## 7     Clinton St & Washington Blvd 41.88341 -87.64114
## 8  Lakefront Trail & Bryn Mawr Ave 41.98404 -87.65230
## 9       Larrabee St & Kingsbury St 41.89774 -87.64288
## 10       Wacker Dr & Washington St 41.88329 -87.63652
## 11         Clinton St & Madison St 41.88278 -87.64120
## 12          State St & Randolph St 41.88469 -87.62778
## 13       LaSalle St & Jackson Blvd 41.87820 -87.63176
```

```
## Create the map

map_member_route_stations <- leaflet(member_top_routes_stations_duration_trip) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(
    ~longitude, ~latitude,
    radius = 6,
    color = "blue",
    fillOpacity = 0.9,
    stroke = FALSE,
    popup = ~paste0(":</strong> ", station_name, "<br>"
    )
  ) %>%
  addControl(
    html = paste0(
      "<div style='text-align:left; padding:8px; font-size:12px; background:white; border-radius:8px;'>
```

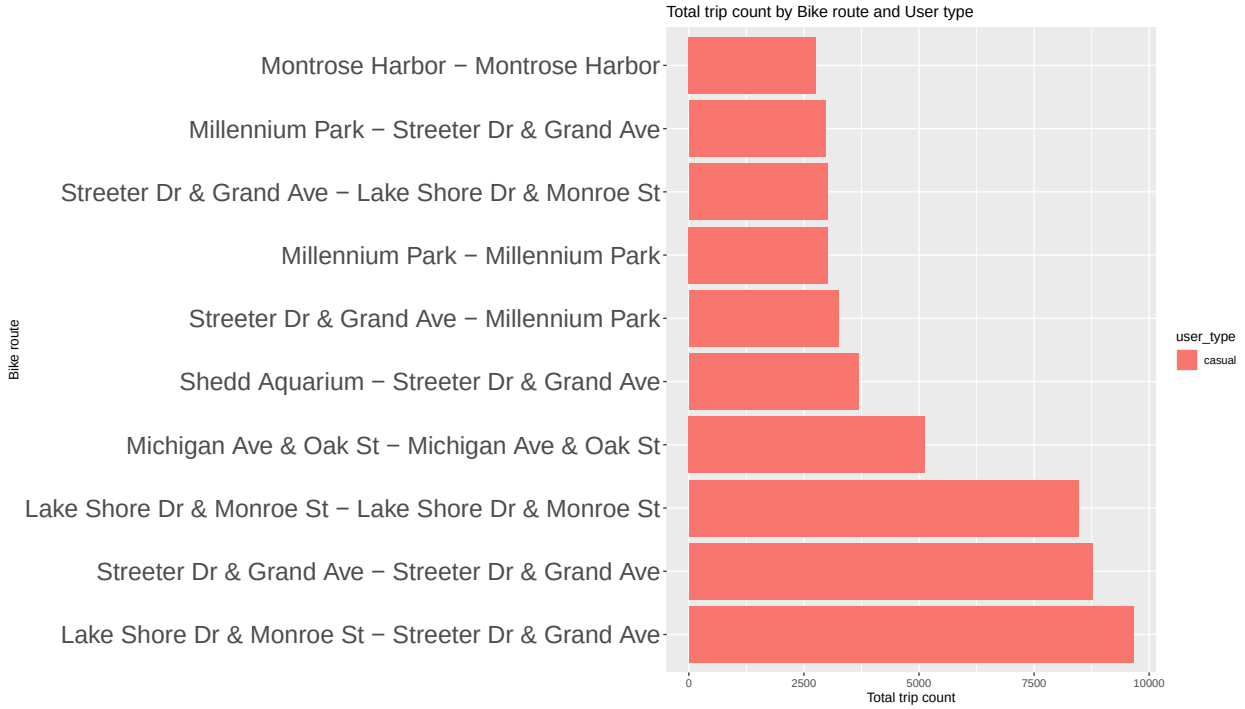
```
      "<strong>Trip duration & Frequency : Stations from Top 10 Routes each (Annual members)</strong><br>",
      paste0(
        member_top_routes_stations_duration_trip %>%
          mutate(label = paste0(" * ", station_name)) %>%
          pull(label),
        collapse = "<br>"
      ),
      "</div>"
    ),
    position = "topright"
  )

map_member_route_stations
```



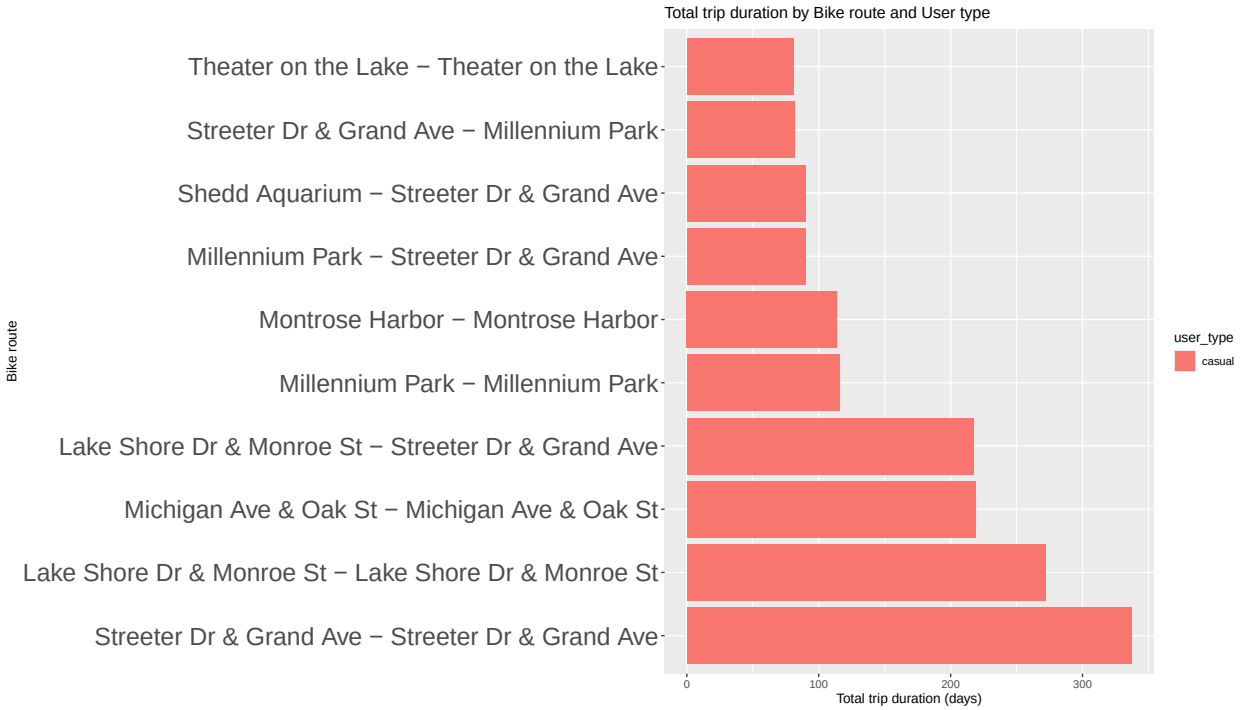
2) **(Horizontal bar charts) Total trip count & Total trip duration of Casual users by Bike route**

```
ggplot(summary5_df %>%
  filter(user_type == 'casual') %>%
    arrange(total_trip_count), aes(y = forcats::fct_reorder(bike_route, -total_trip_count),
                        x = total_trip_count,
                        fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Total trip count by Bike route and User type",
       x = "Total trip count",
```

```
      y = "Bike route")+
  theme(axis.text.y = element_text(size = 20))
```




```
ggplot(summary5_df_1 %>%
  filter(user_type == 'casual') %>%
    arrange(total_trip_duration), aes(y = forcats::fct_reorder(bike_route, -total_trip_duration),
                        x = total_trip_duration,
                        fill = user_type)) +
  geom_col(position = "identity") +
  labs(title = "Total trip duration by Bike route and User type",
       x = "Total trip duration (days)",
       y = "Bike route")+
  theme(axis.text.y = element_text(size = 20))
```



6. ACT PHASE (Deliverable = Top 3 recommendations based on the analysis)

Here are my Top 3 recommendations for the Marketing executives :

1. **Launch referral-based promotions to convert students into annual members on Chicago college campuses near the Lake Michigan shoreline during peak months from May to October.**

- Casual riders are primarily active in the 18–25 age range, and the top stations for casual riders are located along Chicago's Lake Michigan shoreline.

2. **Offer seasonal weekend promotions from late spring through summer at bike stations along the Chicago shoreline, also targeting top routes of the casual riders.**

- Casual rider activity peaks on weekends and during late spring (May) through the summer season (June to September).

3. **Host contests for adult and middle-aged riders at top casual rider stations during peak months, offering discounted annual memberships as prizes along with mid-event campaigns highlighting the health and budget benefits of annual memberships.**

- Casual riders are highly active in the age range of 26-51 (Young adults, Middle-aged adults).

Refer the Final slideshow report for more details : Click Here to view the Final Slideshow Report

Remarks :

With the completion of this case study and the development of an infant problem solving system, I will now focus on solving more complex challenges while iteratively enhancing the system's capabilities.

Ciao.

If you found this analysis 'really' interesting or not, you can get to know me or contact me :

LinkedIn : https://www.linkedin.com/in/r-amarthya-sreechand-3223351b4/.

Email : amarthyasreechand@gmail.com.