Case Study: How Does a Bike-Share Navigate Speedy Success?

Luis Prieto Uzcategui

2022-08-27

1. ASK

1.1 What is the problem you are trying to solve?

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

1.2 How can your insights drive business decisions?

Using historical data for the past 12 months from June 2020 to May 2021 to understand what annual members and casual riders use Cyclistic bikes differently.

1.3 Key tasks

- How do annual members and casual riders use Cyclistic bikes differently?
- Why would casual riders buy Cyclistic annual memberships?
- How can Cyclistic use digital media to influence casual riders to become members?

1.4 Key stakeholders

```
* Director of Marketing Lily Moreno
* Cyclistic marketing analytics team
* Cyclistic executive
```

2. PREPARE

2.1 Where is your data located?

The data collected is from link

2.2 How is the data organized?

The data is considered structured data because is organized in a certain format, like rows and columns.

2.3 Are there issues with bias or credibility in this data? Does your data ROCCC?

Data has been downloaded from Motivate International Inc. Local copies have been stored securely on Google Drive and here on Kaggle.

2.4 How are you addressing licensing, privacy, security, and accessibility?

The data has been made available by Motivate International Inc. under this link

2.5 How did you verify the data's integrity?

This is public data that you can use to explore how different customer types are using Cyclistic bikes. We are going to assume the data is credible.

2.6 Sort and filter the data

I filtered and use the "find and select" > "Go to special" > "Blanks" allowed to delete the blank row. If I found a blank row in any column, I erased the whole row because the data is public, and could not find the reason why the data was empty.

3. PROCESS

3.1 Choose your tools

R for cleaning, analysis, and data visualization. I used Janitor,here and Dplyr.

3.2 Check the data for errors

The null values were eliminated in the spreadsheets by filtering and "find and select." Originally all data frames tripdata_202006 to tripdata_202105 started_at and ended_at were considered as a character instead of DateTime. I did the changes using readr(). Empty rows or columns were cleaned back in excel. But only to add the extra step of cleaning, I used the janitor to remove empty columns and rows.

3.3 Transform the data into the right type

- For the datasets or data frames from tripdata_202006 to tripdata_202011, the started_at and ended_at were considered characters instead of DateTime, so I changed in there too.
- Additionally, start_station_id and end_station_id from 202006 to 202011 were considered double or numeric instead of characters; and from 202012 to 202105 were considered data type characters already.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(janitor)
```

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

```
library(lubridate)
```

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

```
library(dplyr)
library(here)
```

```
## here() starts at C:/Users/luisp/Documents/Case Study 1
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

3.4 Importing files using readr

```
 tripdata_202006 <- read_csv("Bike data/202006-divvy-tripdata.csv",
                             col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                              ended_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                              start_station_id = col_character(),
                                              end_station_id = col_character())
                             )
tripdata_202007 <- read_csv("Bike data/202007-divvy-tripdata.csv",
                             col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                              ended_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                              start_station_id = col_character(),
                                              end_station_id = col_character())
                             )
tripdata_202008 <- read_csv("Bike data/202008-divvy-tripdata.csv",
                             col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
```

```
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        start_station_id = col_character(),
                                        end_station_id = col_character())
                            )
tripdata_202009 <- read_csv("Bike data/202009-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        start_station_id = col_character(),
                                        end_station_id = col_character())
                            )
tripdata_202010 <- read_csv("Bike data/202010-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        start_station_id = col_character(),
                                        end_station_id = col_character())
                            )
tripdata_202011 <- read_csv("Bike data/202011-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        start_station_id = col_character(),
                                        end_station_id = col_character())
                            )
tripdata_202012 <- read_csv("Bike data/202012-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M")))

tripdata_202101 <- read_csv("Bike data/202101-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M")))

tripdata_202102 <- read_csv("Bike data/202102-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M")))

tripdata_202103 <- read_csv("Bike data/202103-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M")))

tripdata_202104 <- read_csv("Bike data/202104-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M")))

tripdata_202105 <- read_csv("Bike data/202105-divvy-tripdata.csv",
                            col_types = cols(started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
                                        ended_at = col_datetime(format = "%m/%d/%Y %H:%M")))
```

3.5 Combining each data frame to get twelve months of data bike

```
combined_databike <- bind_rows(tripdata_202006,tripdata_202007,tripdata_202008,tripdata_202009,
                               tripdata_202010,tripdata_202011,tripdata_202012,tripdata_202101,
                               tripdata_202102,tripdata_202103,tripdata_202104,tripdata_202105)
str(combined_databike)
```

```
## spec_tbl_df [3,745,465 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:3745465] "8CD5DE2C2B6C4CFC" "9A191EB2C751D85D" "F37D14B0B5659BCF" "C41
##  $ rideable_type     : chr [1:3745465] "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:3745465], format: "2020-06-13 23:24:00" "2020-06-26 07:26:00" ...
##  $ ended_at          : POSIXct[1:3745465], format: "2020-06-13 23:36:00" "2020-06-26 07:31:00" ...
##  $ start_station_name: chr [1:3745465] "Wilton Ave & Belmont Ave" "Federal St & Polk St" "Daley Cente
##  $ start_station_id  : chr [1:3745465] "117" "41" "81" "303" ...
##  $ end_station_name  : chr [1:3745465] "Damen Ave & Clybourn Ave" "Daley Center Plaza" "State St & Ha
##  $ end_station_id    : chr [1:3745465] "163" "81" "5" "294" ...
##  $ start_lat         : num [1:3745465] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:3745465] -87.7 -87.6 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num [1:3745465] 41.9 41.9 41.9 42 41.9 ...
##  $ end_lng           : num [1:3745465] -87.7 -87.6 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr [1:3745465] "casual" "member" "member" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = "%m/%d/%Y %H:%M"),
##   ..   ended_at = col_datetime(format = "%m/%d/%Y %H:%M"),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

3.6 More data cleaning

```
glimpse(combined_databike) ## With glimpse I have 3,745,465 rows and 13 columns
```

```
## Rows: 3,745,465
## Columns: 13
## $ ride_id            <chr> "8CD5DE2C2B6C4CFC", "9A191EB2C751D85D", "F37D14B0B5~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-06-13 23:24:00, 2020-06-26 07:26:00, 2020-06-~
## $ ended_at           <dttm> 2020-06-13 23:36:00, 2020-06-26 07:31:00, 2020-06-~
## $ start_station_name <chr> "Wilton Ave & Belmont Ave", "Federal St & Polk St",~
## $ start_station_id   <chr> "117", "41", "81", "303", "327", "327", "41", "115"~
## $ end_station_name   <chr> "Damen Ave & Clybourn Ave", "Daley Center Plaza", "~
## $ end_station_id     <chr> "163", "81", "5", "294", "117", "117", "81", "303",~
## $ start_lat          <dbl> 41.94018, 41.87208, 41.88424, 41.94553, 41.92154, 4~
## $ start_lng          <dbl> -87.65304, -87.62954, -87.62963, -87.64644, -87.653~
## $ end_lat            <dbl> 41.93193, 41.88424, 41.87405, 41.97835, 41.94018, 4~
## $ end_lng            <dbl> -87.67786, -87.62963, -87.62772, -87.65975, -87.653~
## $ member_casual      <chr> "casual", "member", "member", "casual", "casual", "~
```

```r
head(combined_databike) ## head allows a preview of the column names.
```

```
## # A tibble: 6 x 13
##   ride_id        ridea~1 started_at          ended_at            start~2 start~3
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 8CD5DE2C2B6C4~ docked~ 2020-06-13 23:24:00 2020-06-13 23:36:00 Wilton~ 117
## 2 9A191EB2C751D~ docked~ 2020-06-26 07:26:00 2020-06-26 07:31:00 Federa~ 41
## 3 F37D14B0B5659~ docked~ 2020-06-23 17:12:00 2020-06-23 17:21:00 Daley ~ 81
## 4 C41237B506E85~ docked~ 2020-06-20 01:09:00 2020-06-20 01:28:00 Broadw~ 303
## 5 4B51B3B0BDA77~ docked~ 2020-06-25 16:59:00 2020-06-25 17:08:00 Sheffi~ 327
## 6 D50DF288196B5~ docked~ 2020-06-17 18:07:00 2020-06-17 18:18:00 Sheffi~ 327
## # ... with 7 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names 1: rideable_type,
## #   2: start_station_name, 3: start_station_id
```

```r
combined_databike <- janitor::remove_empty(combined_databike, which = c("cols"))
combined_databike <- janitor::remove_empty(combined_databike, which = c("rows"))
dim(combined_databike) # to check if there was any changes after checking for empty spaces
```

```
## [1] 3745465      13
```

3.7 Remove columns not required or beyond the scope of project

```r
combined_databike <- combined_databike %>%
  select(-start_station_id,-end_station_id)

combined_databike <- combined_databike %>%
  select(-start_lat,-start_lng,-end_lat,-end_lng)
```

3.8 Checking combined_databike after erasing columns

```r
colnames(combined_databike) #List of column names
```

```
## [1] "ride_id"            "rideable_type"      "started_at"
## [4] "ended_at"           "start_station_name" "end_station_name"
## [7] "member_casual"
```

```r
nrow(combined_databike) #How many rows are in data frame?
```

```
## [1] 3745465
```

```r
dim(combined_databike) #Dimensions of the data frame?
```

```
## [1] 3745465       7
```

```
summary(combined_databike) # Statistical summary of data. Mainly for numerics
```

```
##    ride_id          rideable_type        started_at
##  Length:3745465     Length:3745465     Min.   :2020-06-03 05:59:00.00
##  Class :character   Class :character   1st Qu.:2020-08-03 13:43:00.00
##  Mode  :character   Mode  :character   Median :2020-09-23 13:34:00.00
##                                        Mean   :2020-11-02 01:40:18.89
##                                        3rd Qu.:2021-03-05 16:48:00.00
##                                        Max.   :2021-05-31 23:59:00.00
##     ended_at                      start_station_name end_station_name
##  Min.   :2020-06-03 06:03:00.00   Length:3745465     Length:3745465
##  1st Qu.:2020-08-03 14:13:00.00   Class :character   Class :character
##  Median :2020-09-23 13:57:00.00   Mode  :character   Mode  :character
##  Mean   :2020-11-02 02:05:03.25
##  3rd Qu.:2021-03-05 17:05:00.00
##  Max.   :2021-06-10 22:17:00.00
##  member_casual
##  Length:3745465
##  Class :character
##  Mode  :character
##
##
##
```

3.9 Data transformation and manipulation

```
combined_databike$date <- as.Date(combined_databike$started_at) # The default format is yyyy-mm-dd
combined_databike$month <- format(as.Date(combined_databike$date),"%B")
combined_databike$day <- format(as.Date(combined_databike$date), "%d")
combined_databike$year <- format(as.Date(combined_databike$date),"%Y")
combined_databike$day_of_week <- format(as.Date(combined_databike$date), "%A")
## Add a "ride_length" calculation to all_trips (in seconds)
combined_databike$ride_length <- difftime(combined_databike$ended_at,combined_databike$started_at)
```

```
str(combined_databike)
```

```
## tibble [3,745,465 x 13] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:3745465] "8CD5DE2C2B6C4CFC" "9A191EB2C751D85D" "F37D14B0B5659BCF" "C411
##  $ rideable_type     : chr [1:3745465] "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:3745465], format: "2020-06-13 23:24:00" "2020-06-26 07:26:00" ...
##  $ ended_at          : POSIXct[1:3745465], format: "2020-06-13 23:36:00" "2020-06-26 07:31:00" ...
##  $ start_station_name: chr [1:3745465] "Wilton Ave & Belmont Ave" "Federal St & Polk St" "Daley Cente
##  $ end_station_name  : chr [1:3745465] "Damen Ave & Clybourn Ave" "Daley Center Plaza" "State St & Ha
##  $ member_casual     : chr [1:3745465] "casual" "member" "member" "casual" ...
##  $ date              : Date[1:3745465], format: "2020-06-13" "2020-06-26" ...
##  $ month             : chr [1:3745465] "June" "June" "June" "June" ...
##  $ day               : chr [1:3745465] "13" "26" "23" "20" ...
##  $ year              : chr [1:3745465] "2020" "2020" "2020" "2020" ...
##  $ day_of_week       : chr [1:3745465] "Saturday" "Friday" "Tuesday" "Saturday" ...
##  $ ride_length       : 'difftime' num [1:3745465] 720 300 540 1140 ...
##   ..- attr(*, "units")= chr "secs"
```

3.10 Convert "ride_length" from Factor to numeric so we can run calculations on the data

```
is.factor(combined_databike$ride_length)
```

```
## [1] FALSE
```

```
combined_databike$ride_length <- as.numeric(as.character(combined_databike$ride_length))
is.numeric(combined_databike$ride_length)
```

```
## [1] TRUE
```

```
# # Remove "bad" data
combined_databike_2 <- combined_databike[!(combined_databike$start_station_name == "HQ QR" | combined_da
```

After removing bad data the new combined_databike_2 reduced from 3,745,465 to 3,742,263. The columns of combined_databike_2 are 13 still.

4. ANALYZE

4.1 Descriptive analysis on ride_length (all figures in seconds)

```
mean(combined_databike_2$ride_length) #straight average (total ride length / rides)
```

```
## [1] 1626.104
```

```
median(combined_databike_2$ride_length)  #midpoint number in the ascending array of ride lengths
```

```
## [1] 840
```

```
max(combined_databike_2$ride_length) #longest ride
```

```
## [1] 3256980
```

```
min(combined_databike_2$ride_length) #shortest ride
```

```
## [1] 0
```

```
summary(combined_databike_2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     480     840    1626    1560 3256980
```

Observations: The average time of ride is 1626.10 seconds between casual customer and member customer

4.2 Compare members and casual users

```
aggregate(combined_databike_2$ride_length ~ combined_databike_2$member_casual, FUN = mean)
```

```
##   combined_databike_2$member_casual combined_databike_2$ride_length
## 1                            casual                       2603.7754
## 2                            member                        915.8042
```

```
aggregate(combined_databike_2$ride_length ~ combined_databike_2$member_casual, FUN = median)
```

```
##   combined_databike_2$member_casual combined_databike_2$ride_length
## 1                            casual                            1260
## 2                            member                             660
```

```
aggregate(combined_databike_2$ride_length ~ combined_databike_2$member_casual, FUN = max)
```

```
##   combined_databike_2$member_casual combined_databike_2$ride_length
## 1                            casual                         3256980
## 2                            member                         2476260
```

```
aggregate(combined_databike_2$ride_length ~ combined_databike_2$member_casual, FUN = min)
```

```
##   combined_databike_2$member_casual combined_databike_2$ride_length
## 1                            casual                               0
## 2                            member                               0
```

Observations: We can observe that casual customers spend more time (2603.78) in their ride from started_at to ended_at than member customers (915.8)

4.3 See the average ride time by each day for members vs casual users

```
aggregate(combined_databike_2$ride_length ~ combined_databike_2$member_casual + combined_databike_2$day
```

```
##    combined_databike_2$member_casual combined_databike_2$day_of_week
## 1                             casual                          Friday
## 2                             member                          Friday
## 3                             casual                          Monday
## 4                             member                          Monday
## 5                             casual                        Saturday
## 6                             member                        Saturday
## 7                             casual                          Sunday
## 8                             member                          Sunday
## 9                             casual                        Thursday
## 10                            member                        Thursday
## 11                            casual                         Tuesday
## 12                            member                         Tuesday
## 13                            casual                       Wednesday
## 14                            member                       Wednesday
##    combined_databike_2$ride_length
```

```
## 1                          2473.0872
## 2                           893.6181
## 3                          2575.7317
## 4                           875.5580
## 5                          2702.2623
## 6                          1008.4317
## 7                          2974.7040
## 8                          1039.1421
## 9                          2447.7436
## 10                          864.1151
## 11                         2316.4070
## 12                          863.0295
## 13                         2346.7785
## 14                          870.6256
```

Notice that the days of the week are out of order. Let's fix that

```
combined_databike_2$day_of_week <- ordered(combined_databike_2$day_of_week, levels=c("Sunday", "Monday"
```

Now, let's run the average ride time by each day for members vs casual users

```
aggregate(combined_databike_2$ride_length ~ combined_databike_2$member_casual + combined_databike_2$day
```

```
##    combined_databike_2$member_casual combined_databike_2$day_of_week
## 1                             casual                           Sunday
## 2                             member                           Sunday
## 3                             casual                           Monday
## 4                             member                           Monday
## 5                             casual                          Tuesday
## 6                             member                          Tuesday
## 7                             casual                        Wednesday
## 8                             member                        Wednesday
## 9                             casual                         Thursday
## 10                            member                         Thursday
## 11                            casual                           Friday
## 12                            member                           Friday
## 13                            casual                         Saturday
## 14                            member                         Saturday
##    combined_databike_2$ride_length
## 1                        2974.7040
## 2                        1039.1421
## 3                        2575.7317
## 4                         875.5580
## 5                        2316.4070
## 6                         863.0295
## 7                        2346.7785
## 8                         870.6256
## 9                        2447.7436
## 10                        864.1151
## 11                       2473.0872
## 12                        893.6181
## 13                       2702.2623
## 14                       1008.4317
```

Observations: The average ride time by members vs casuals customers by each day were disorganized. I used ordered and levels to start from Sunday to Saturday. Each day casual customers spend more time than member customers.

4.4 See the average ride time by each month for members vs casuals users

```
aggregate(combined_databike_2$ride_length ~ combined_databike_2$member_casual + combined_databike_2$mon
```

```
##    combined_databike_2$member_casual combined_databike_2$month
## 1                             casual                     April
## 2                             member                     April
## 3                             casual                    August
## 4                             member                    August
## 5                             casual                  December
## 6                             member                  December
## 7                             casual                  February
## 8                             member                  February
## 9                             casual                   January
## 10                            member                   January
## 11                            casual                      July
## 12                            member                      July
## 13                            casual                      June
## 14                            member                      June
## 15                            casual                     March
## 16                            member                     March
## 17                            casual                       May
## 18                            member                       May
## 19                            casual                  November
## 20                            member                  November
## 21                            casual                   October
## 22                            member                   October
## 23                            casual                 September
## 24                            member                 September
##    combined_databike_2$ride_length
## 1                        2306.3654
## 2                         855.8439
## 3                        2646.9551
## 4                         990.2215
## 5                        1659.7466
## 6                         738.3983
## 7                        2828.5983
## 8                         886.7673
## 9                        1612.4128
## 10                        720.7699
## 11                       3550.7742
## 12                       1050.9757
## 13                       3070.8678
## 14                       1110.8220
## 15                       2308.9062
## 16                        819.8550
## 17                       2378.0678
## 18                        860.6578
```

```
## 19                        2005.3134
## 20                         807.0817
## 21                        1874.6899
## 22                         833.7450
## 23                        2300.0574
## 24                         913.0426
```

Notice that the days of the week are out of order. Let's fix that

```
combined_databike_2$month <- ordered(combined_databike_2$month,c("June", "July", "August", "September",
```

Now, let's run the average ride time by each day for members vs casual users

```
aggregate(combined_databike_2$ride_length ~ combined_databike_2$member_casual + combined_databike_2$mon
```

```
##    combined_databike_2$member_casual combined_databike_2$month
## 1                             casual                      June
## 2                             member                      June
## 3                             casual                      July
## 4                             member                      July
## 5                             casual                    August
## 6                             member                    August
## 7                             casual                 September
## 8                             member                 September
## 9                             casual                   October
## 10                            member                   October
## 11                            casual                  November
## 12                            member                  November
## 13                            casual                  December
## 14                            member                  December
## 15                            casual                   January
## 16                            member                   January
## 17                            casual                  February
## 18                            member                  February
## 19                            casual                     March
## 20                            member                     March
## 21                            casual                     April
## 22                            member                     April
## 23                            casual                       May
## 24                            member                       May
##    combined_databike_2$ride_length
## 1                        3070.8678
## 2                        1110.8220
## 3                        3550.7742
## 4                        1050.9757
## 5                        2646.9551
## 6                         990.2215
## 7                        2300.0574
## 8                         913.0426
## 9                        1874.6899
## 10                        833.7450
## 11                       2005.3134
## 12                        807.0817
```

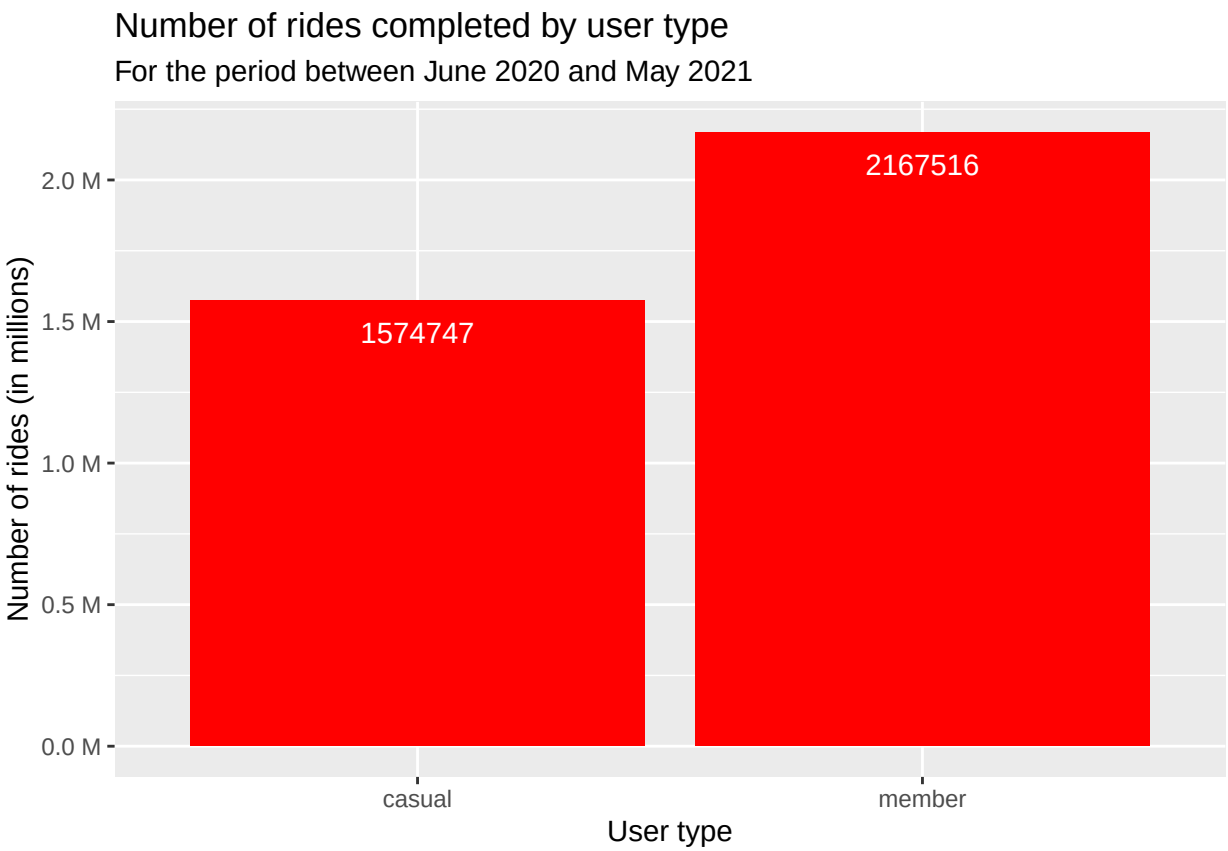
```
## 13                       1659.7466
## 14                        738.3983
## 15                       1612.4128
## 16                        720.7699
## 17                       2828.5983
## 18                        886.7673
## 19                       2308.9062
## 20                        819.8550
## 21                       2306.3654
## 22                        855.8439
## 23                       2378.0678
## 24                        860.6578
```

Observations: The average ride time by members vs casuals customers by each month were disorganized. I used ordered and levels to start from Sunday to Saturday. The month of July was the highest average time for casual members with a time spend of 3550.78 seconds compared to member customers with 1050.98 seconds of that same month. The lowest for casual customers was October with 1874.6899 seconds. The lowest for member customers was 833.7450.

5. SHARE

5.1 Number of rides completed by user type

```
ggplot(combined_databike_2, aes(x=member_casual))+
  geom_bar(fill="Red") +
  labs(
    title = "Number of rides completed by user type",
    subtitle = "For the period between June 2020 and May 2021",
    x = "User type",
    y = "Number of rides (in millions)") +
  scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6)) +
  geom_text(stat='count', aes(label=..count..), vjust=+2, color="white")
```

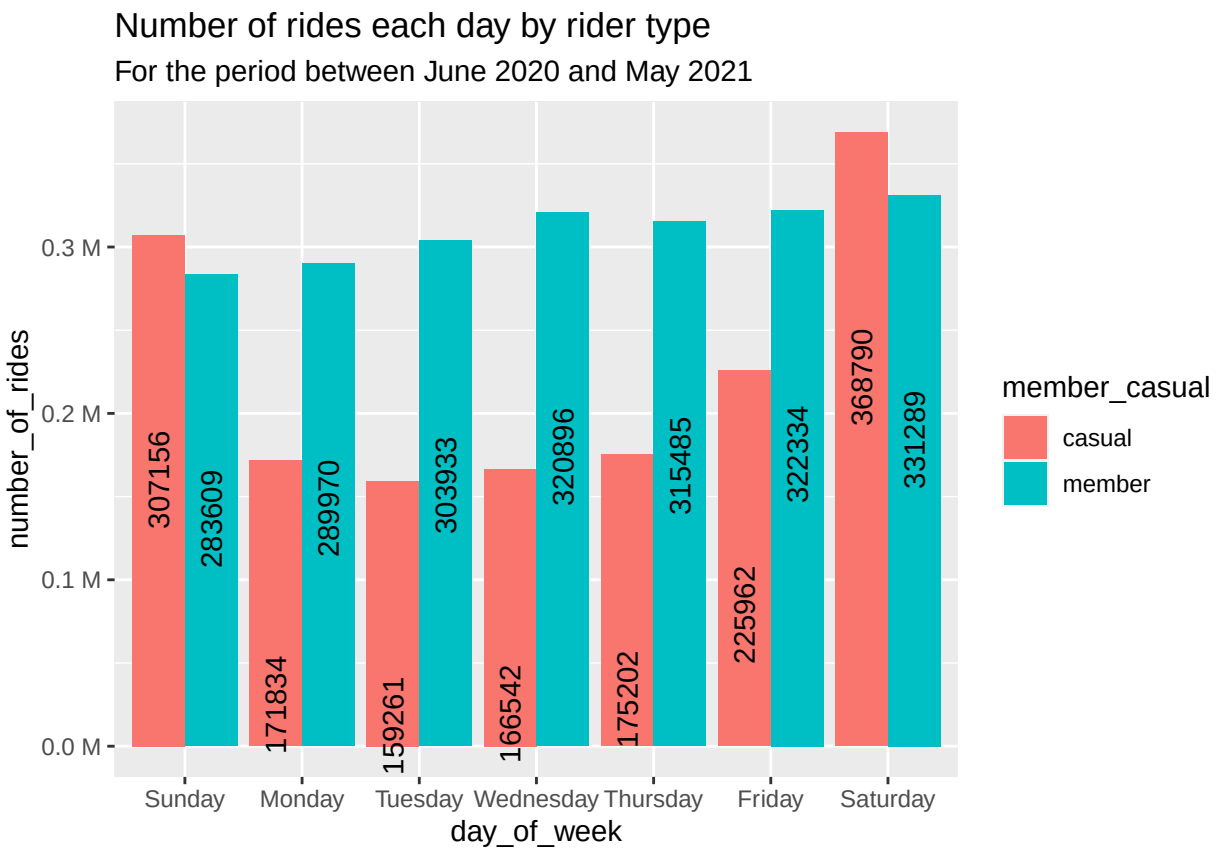


Observations: There are more members customers by 2,167,516 (58%) than casual customers 1,574,747 (42%).

5.2 Number of rides each day by rider type

```
combined_databike_2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title= "Number of rides each day by rider type",
       subtitle= "For the period between June 2020 and May 2021") +
  scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6)) +
  geom_text(aes(label=number_of_rides),position = position_dodge(0.9),hjust=+3, color="black",angle= 90)
```

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

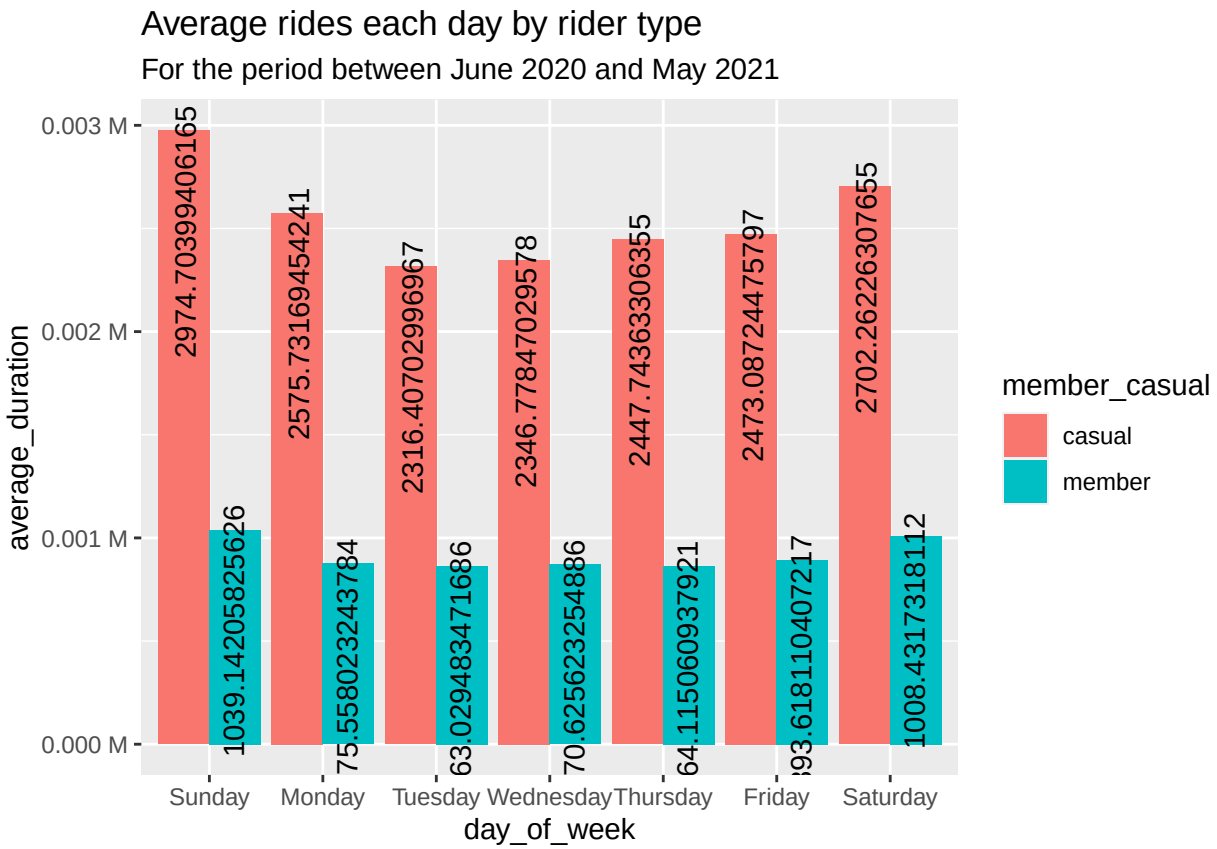


Observations: The highest number of casual customers was on Saturday with 368790 and Tuesday with 159261 the lowest for the casual members. The highest number of member customers was on Saturday with 331289 and Sunday with 283609 the lowest for the members customers. Also the members customers number of rides per day from Sunday to Saturday does not change as much as the casual members that only Sunday and Saturday are the highest and Monday to Friday drop, but increases by day until reach Saturday.

5.3 Let's create a visualization for average duration

```
combined_databike_2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title= "Average rides each day by rider type",
       subtitle= "For the period between June 2020 and May 2021") +
  scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6)) +
  geom_text(aes(label=average_duration),position = position_dodge(0.85),hjust=0.9, color="black",angle=
```

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

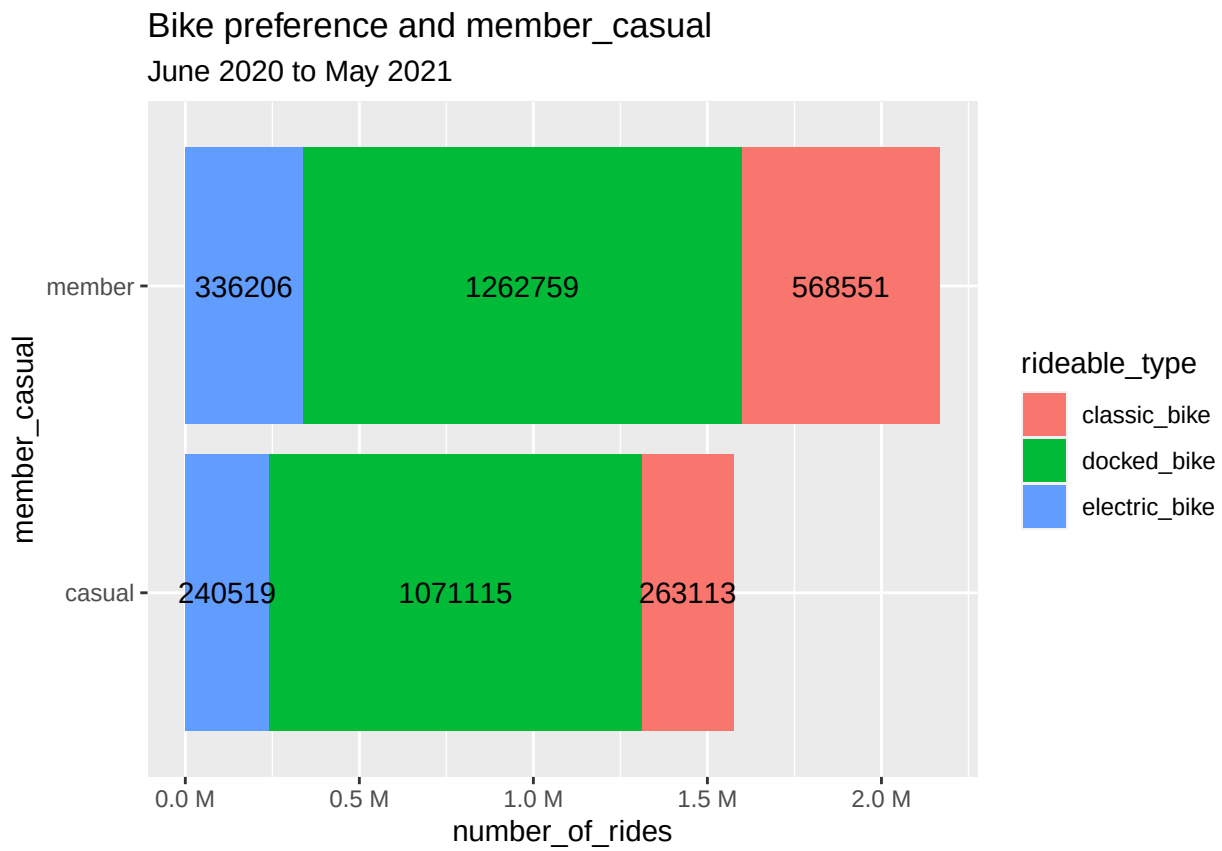


Observations: The casual customers spent more time than members customers by a differences of 1000 seconds more than member customers.

5.4 Bike preference by user type

```
combined_databike_2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, rideable_type)  %>%
  ggplot(aes(x = member_casual, y = number_of_rides, fill = rideable_type)) +
  geom_bar(stat = "identity") + labs(title= "Bike preference and member_casual",
                                     subtitle= "June 2020 to May 2021") +
  coord_flip() +
  geom_text(aes(label=number_of_rides),position = position_stack(vjust = .5), color="black") +
  scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6))
```

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

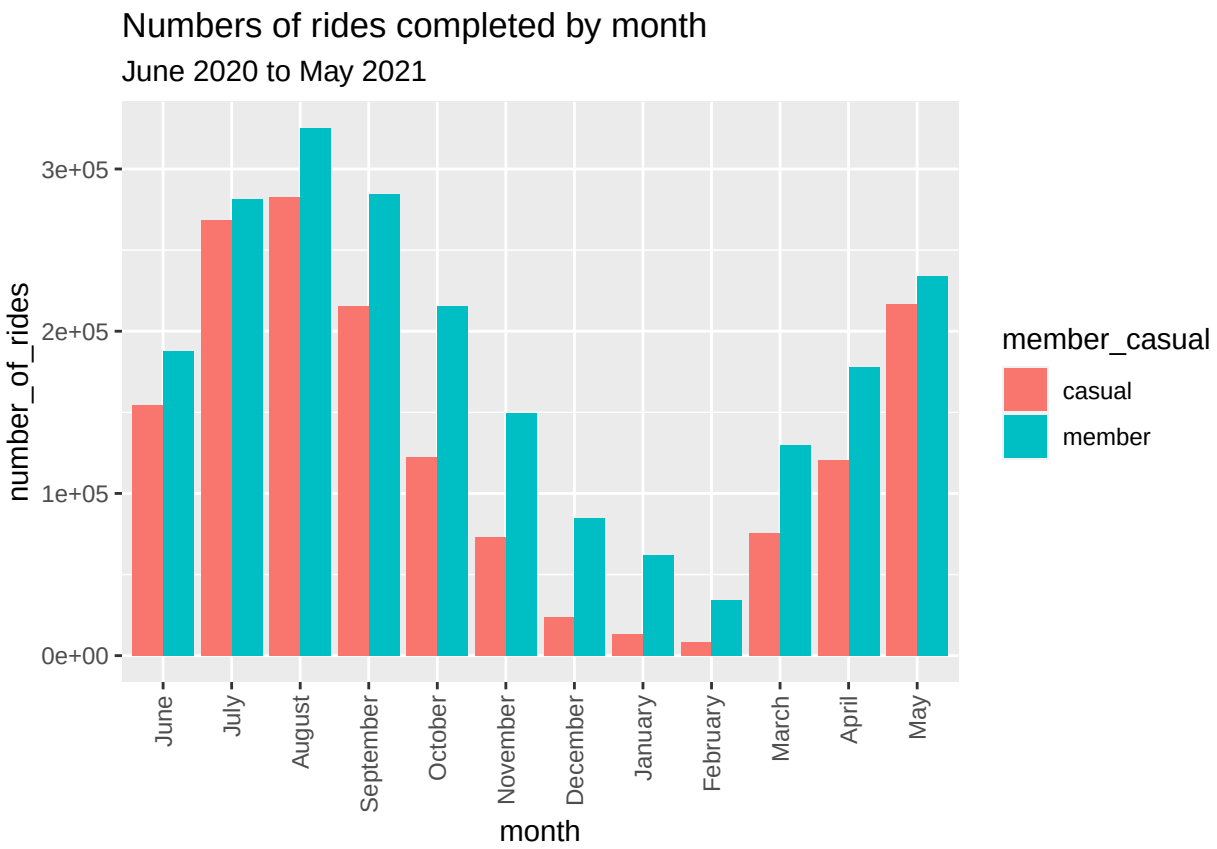


Observations: The docked bike is the most chosen rideable type by casual and member customers. The electric bike is the least chosen bike by casual and member customers.

5.5 Number of rides completed by month by user name

```
combined_databike_2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") + labs(title= "Numbers of rides completed by month",
                                      subtitle= "June 2020 to May 2021") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

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

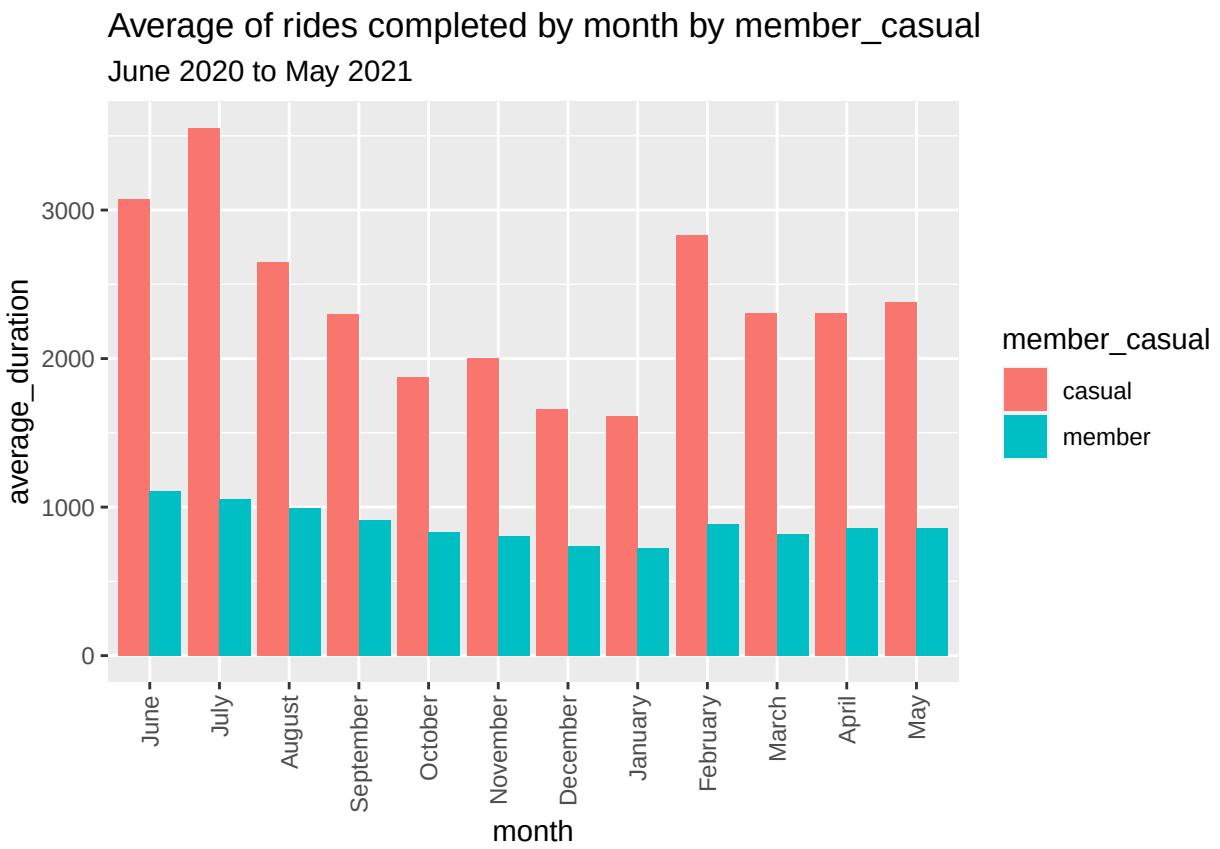


Observations: The casual and member customers highest month is August. The lowest number of rides completed by casual customers are between December to February where it picks up and the same applies to member customers too. The highest number of rides completed by month for members were July, August and September. The casual customers highest months were July and August.

5.6 Average of rides duration completed by month and member_casual

```
combined_databike_2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") + labs(title= "Average of rides completed by month by member_casual",
                                      subtitle= "June 2020 to May 2021") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

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



Observations: Again, in each month the casual customers spent more time than members customers. The differences are casual customers time spent reduces from August to January until it hits February.

5.7 Top 10 start station by user types

```
Table1 <- combined_databike_2 %>%
  group_by(member_casual, start_station_name) %>%
  summarise(count_of=n()) %>%
  arrange(desc(count_of)) %>%
  na.omit(start_station_name)
```

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

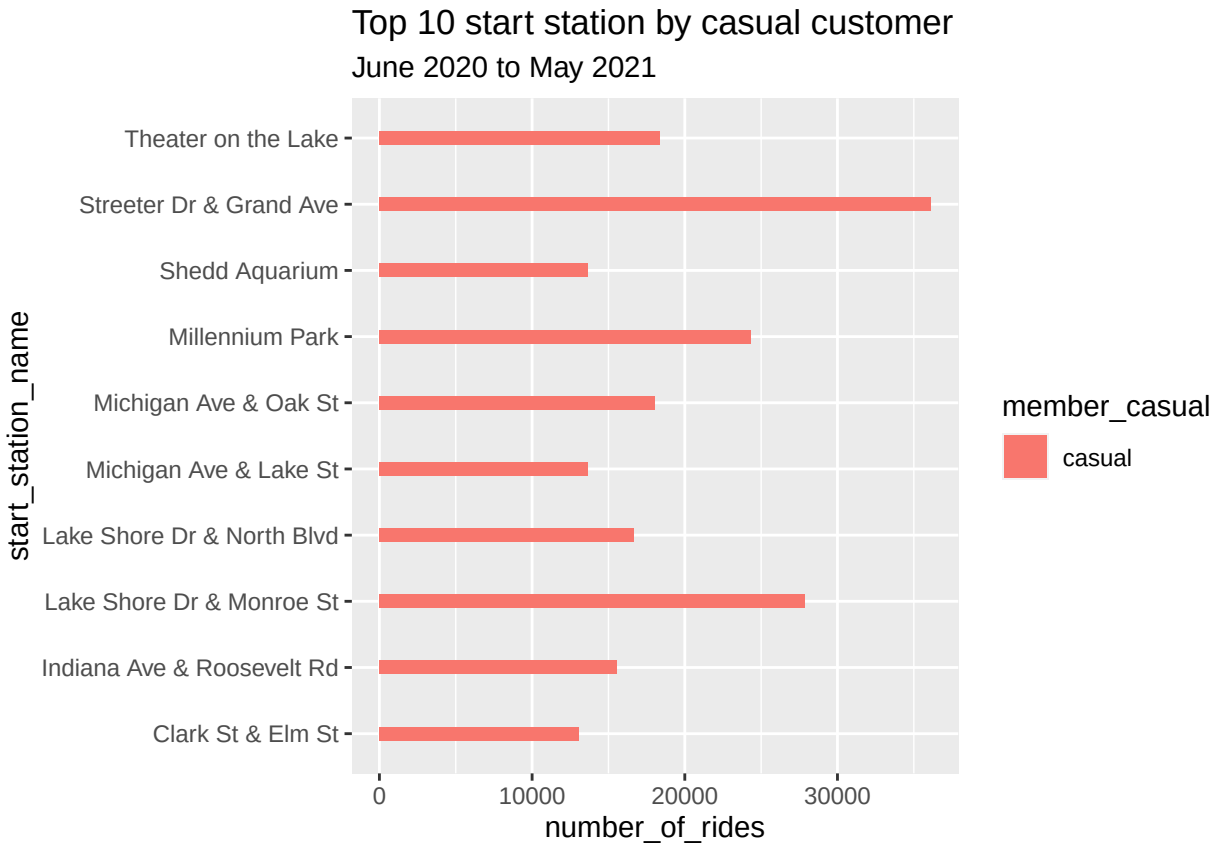
```
## Table 1.1 - By casual riders ##
Table1.1 <- filter(Table1, member_casual =="casual") %>%
  rename(number_of_rides = count_of) %>%
  slice(1:10)

#Table 1.2 - By members ##
Table1.2 <- filter(Table1, member_casual =="member") %>%
  rename(number_of_rides = count_of) %>%
  slice(1:10)
```

```
print(Table1.1)
```

```
## # A tibble: 10 x 3
## # Groups:   member_casual [1]
##    member_casual start_station_name         number_of_rides
##    <chr>         <chr>                                <int>
##  1 casual        Streeter Dr & Grand Ave              36107
##  2 casual        Lake Shore Dr & Monroe St            27882
##  3 casual        Millennium Park                      24338
##  4 casual        Theater on the Lake                  18363
##  5 casual        Michigan Ave & Oak St                18017
##  6 casual        Lake Shore Dr & North Blvd           16645
##  7 casual        Indiana Ave & Roosevelt Rd           15567
##  8 casual        Shedd Aquarium                       13681
##  9 casual        Michigan Ave & Lake St               13634
## 10 casual        Clark St & Elm St                    13046
```

```
## Top 10 start station by casual customer
  ggplot(Table1.1,aes(x = start_station_name, y = number_of_rides, fill = member_casual)) +
  geom_bar(stat = "identity", width = 0.2) +
  coord_flip() +
  labs(title = "Top 10 start station by casual customer", subtitle = "June 2020 to May 2021")
```

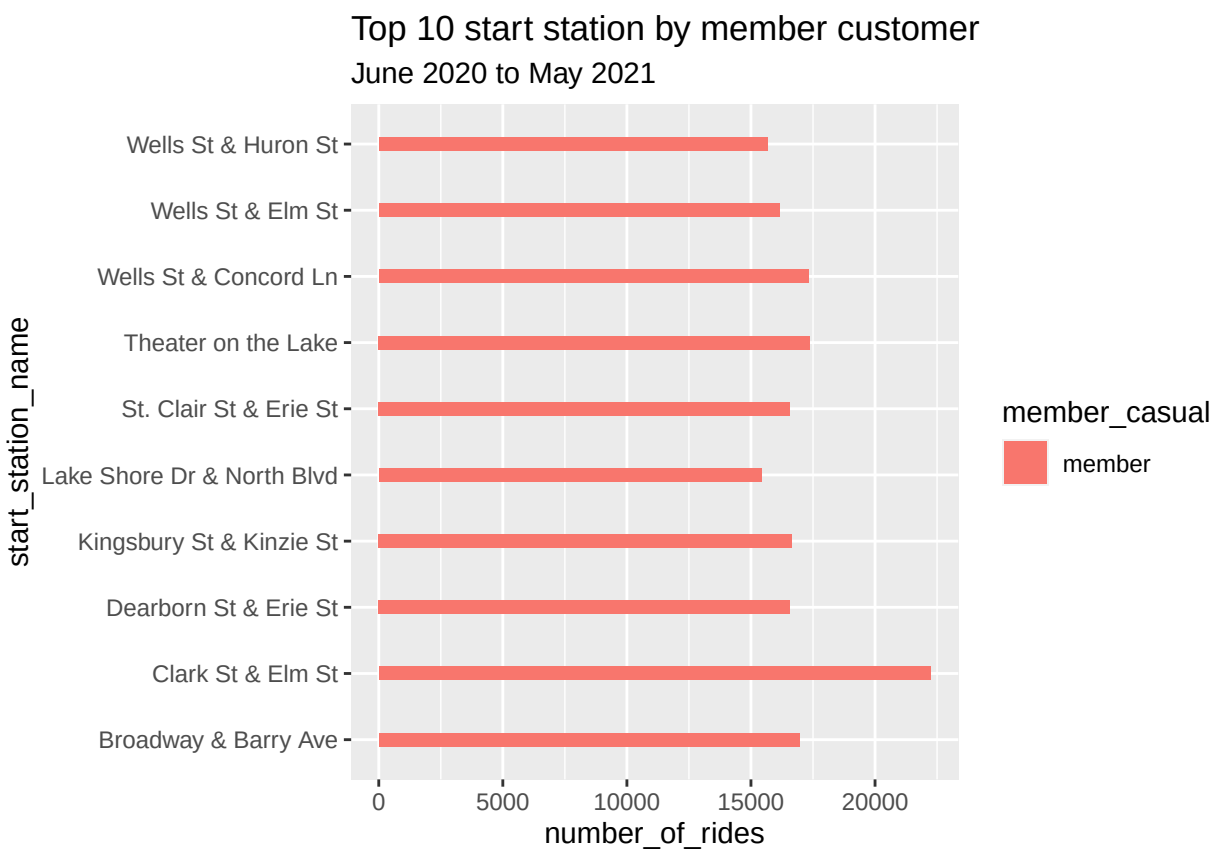


Observations: The "Streeter Dr & Grand Ave" was the most use start station by casual customers making 36107 rides from where to start.

```
print(Table1.2)
```

```
## # A tibble: 10 x 3
## # Groups:   member_casual [1]
##    member_casual start_station_name         number_of_rides
##    <chr>         <chr>                                <int>
##  1 member        Clark St & Elm St                    22243
##  2 member        Theater on the Lake                  17375
##  3 member        Wells St & Concord Ln                17310
##  4 member        Broadway & Barry Ave                 16949
##  5 member        Kingsbury St & Kinzie St             16651
##  6 member        St. Clair St & Erie St               16570
##  7 member        Dearborn St & Erie St                16568
##  8 member        Wells St & Elm St                    16160
##  9 member        Wells St & Huron St                  15657
## 10 member        Lake Shore Dr & North Blvd           15421
```

```
## top 10 start station by member customer
  ggplot(Table1.2,aes(x = start_station_name, y = number_of_rides, fill = member_casual)) +
    geom_bar(stat = "identity", width = 0.2) +
    coord_flip() +
    labs(title = "Top 10 start station by member customer",subtitle= "June 2020 to May 2021")
```



Observations: The "Clark St & Elm St" is the most use start station by members customers making 22243 rides from where to start. Also, for members customers Clark St & Elm St is the most used compared to casual customers that is the least used for them.

5.8 Top 10 end station by user types

```
Table2 <- combined_databike_2 %>%
    group_by(member_casual, end_station_name) %>%
    summarise(count_of=n()) %>%
    arrange(desc(count_of)) %>%
    na.omit(end_station_name)
```

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

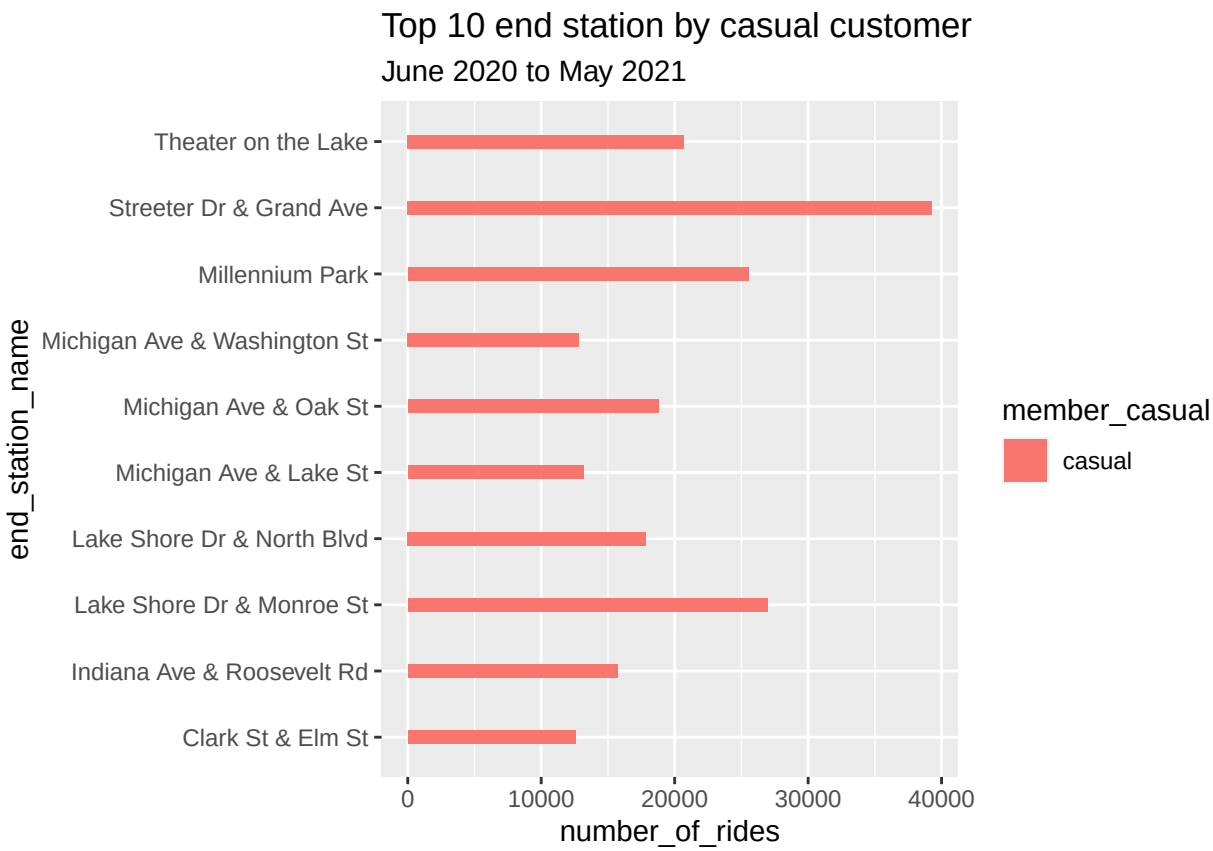
```
  ## Table 2.1 - By casual riders ##
  Table2.1 <- filter(Table2, member_casual =="casual") %>%
    rename(number_of_rides = count_of) %>%
    slice(1:10)

  #Table 2.2 - By members ##
  Table2.2 <- filter(Table2, member_casual =="member") %>%
    rename(number_of_rides = count_of) %>%
    slice(1:10)
```

```
print(Table2.1)
```

```
## # A tibble: 10 x 3
## # Groups:   member_casual [1]
##    member_casual end_station_name             number_of_rides
##    <chr>         <chr>                                  <int>
##  1 casual        Streeter Dr & Grand Ave                39288
##  2 casual        Lake Shore Dr & Monroe St              26958
##  3 casual        Millennium Park                        25547
##  4 casual        Theater on the Lake                    20697
##  5 casual        Michigan Ave & Oak St                  18770
##  6 casual        Lake Shore Dr & North Blvd             17860
##  7 casual        Indiana Ave & Roosevelt Rd             15730
##  8 casual        Michigan Ave & Lake St                 13159
##  9 casual        Michigan Ave & Washington St           12828
## 10 casual        Clark St & Elm St                      12559
```

```
ggplot(Table2.1,aes(x = end_station_name, y = number_of_rides, fill = member_casual)) +
  geom_bar(stat = "identity", width = 0.2) +
  coord_flip() +
  labs(title = "Top 10 end station by casual customer",subtitle= "June 2020 to May 2021")
```

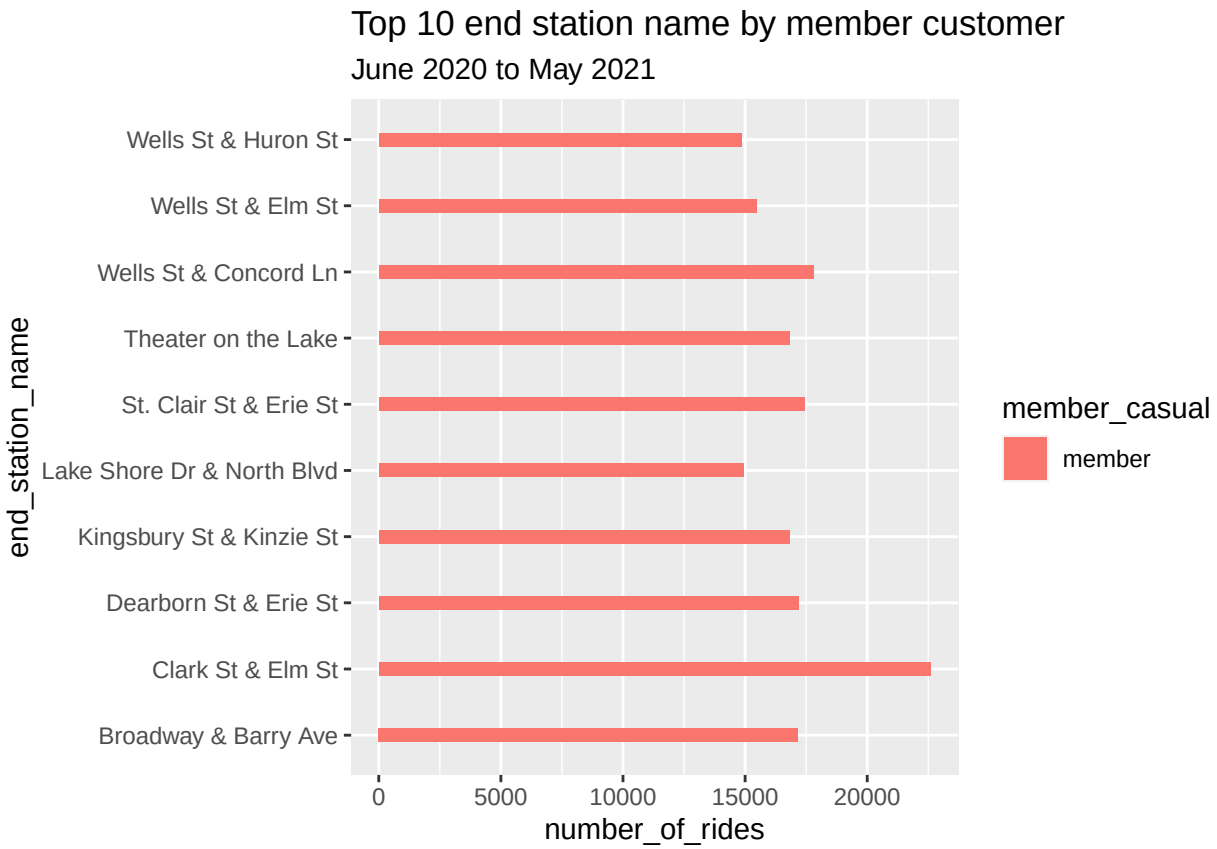


Observations: The "Streeter Dr & Grand Ave" was the most use end station by casual customers making 39288 rides. Also this station name is the most used from start to end by casual customers.

```
print(Table2.2)
```

```
## # A tibble: 10 x 3
## # Groups:   member_casual [1]
##    member_casual end_station_name           number_of_rides
##    <chr>         <chr>                                <int>
##  1 member        Clark St & Elm St                    22599
##  2 member        Wells St & Concord Ln                17791
##  3 member        St. Clair St & Erie St               17433
##  4 member        Dearborn St & Erie St                17190
##  5 member        Broadway & Barry Ave                 17171
##  6 member        Kingsbury St & Kinzie St             16814
##  7 member        Theater on the Lake                  16808
##  8 member        Wells St & Elm St                    15457
##  9 member        Lake Shore Dr & North Blvd           14941
## 10 member        Wells St & Huron St                  14843
```

```
## top 10 start station by member customer
ggplot(Table2.2,aes(x = end_station_name, y = number_of_rides, fill = member_casual)) +
  geom_bar(stat = "identity", width = 0.2) +
  coord_flip() +
  labs(title = "Top 10 end station name by member customer",subtitle= "June 2020 to May 2021")
```



Observations: The "Clark St & Elm St" is the most use start station by members customers making 22599 rides. Also, for members customers Clark St & Elm St is the most used from start to end station name by member customers.

6. ACT

6.1 Conclusions and findings

- Casual customers spent more time on average each day than members customers. To encourage casual customers to become members, Cyclistic could offer a free month of membership, which Cyclistic can use to explain the benefits of becoming member customers by showing how much time casual customers spent than member customers.
- Casual and member customers use the docked bike the most. To keep with the demand Cyclistic should keep a reserve of docked bikes ready to deploy whenever docked bikes are damaged. This is very important because casual customers use this type of bike more than the others.
- The number of casual customers peaks in July and August, but the spent time is still high than member customers. To encourage them to continue using the bike share service, special promotions of membership in other months such as October through February to raise the number of casual members, but also keep up with July and August.
- The "Streeter Dr & Grand Ave" is the most start and end station for casual customers, but there are other stations that casual customers use. One of them is "Clark St & Elm St" which is the least used of the top ten of my analysis, but the most used by members customers. To encourage casual customers to become casual to member customers, Cyclistic can partner with local stores, which can include members customers who can get discounts in these stores.

6.2 Additional data to use and expand Cyclistic findings that could give new insights

- Cyclistic can focus on time like Hours and Minutes compared to only seconds, which was used for this analysis.

7. REFERENCE PAGE

R basics

R tutorial

ggplot gallery

ggplot guide

To create data frame and use for the top start and end station

RStudio Community

Stackoverflow

Sort and Filter data

8. CONTACT PAGE

LinkedIn