Cyclistic Case Study

Akash Kowale

9/12/2021

About Cyclistic:

Chicago’s upcoming bike sharing company, Cyclistic, is changing the way of travelling around the city. Founded in 2016, its program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. It also has options like reclining bicycle, hand tricycles and cargo bikes making the journey accessible to people with disabilities and riders who can’t use a standard two-wheeled bike.

The Problem:

Cyclistic’s finance analysts have concluded that annual members are much more profitable than casual riders. Hence the objective is to get the casual customers to get the annual membership of Cyclistic. We are targetting casual customers as they are easier to convert to annual members compared to a new audience as they are already aware of the company and how it works.

The Business Task:

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

Key stakeholders:

- Lily Moreno, Director of marketing
- Cyclistic marketing analytics team
- Cyclistic executive team

Deliverables:

1. The business task
2. Data sources used
3. Documentation of any cleaning or manipulation of data
4. A summary of the analysis
5. Supporting visualizations and key findings
6. Top three recommendations based on the analysis

The Data

Source:

(https://divvy-tripdata.s3.amazonaws.com/index.html)

The data used was sourced from Motivate International inc’s public data under this license

ROCCC approach is used to determine the credibility of the data

- Reliable – It is complete and accurate and it represents all bike rides taken in the city of Chicago for the selected duration of our analysis.
- Original - The data is made available by Motivate International Inc. which operates the city of Chicago’s Divvy bicycle sharing service which is powered by Lyft.
- Comprehensive - the data includes all information about ride details including starting time, ending time, station name, station ID, type of membership and many more.
- Current – It is up-to-date as it includes data until end of May 2021
- Cited - The data is cited and is available under Data License Agreement.

Step 1: Setting up the environment:

We will start by installing the required packages.

```
#install.packages("tidyverse")
#install.packages("lubridate")
#install.packages("ggplot2")
#install.packages("janitor")
#install.packages("Rcpp")
library(Rcpp)
library(tidyverse)

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(lubridate)

##
## Attaching package: 'lubridate'

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

```
library(ggplot2)
library(dplyr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

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

Step 2: Collecting the Data

We will now load the data of last 12 months of cyclistic.

```
m09_2020 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202009-
divvy-tripdata.csv")
m10_2020 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202010-
divvy-tripdata.csv")
m11_2020 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202011-
divvy-tripdata.csv")
m12_2020 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202012-
divvy-tripdata.csv")
m01_2021 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202101-
divvy-tripdata.csv")
m02_2021 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202102-
divvy-tripdata.csv")
m03_2021 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202103-
divvy-tripdata.csv")
m04_2021 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202104-
divvy-tripdata.csv")
m05_2021 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202105-
divvy-tripdata.csv")
m06_2021 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202106-
divvy-tripdata.csv")
m07_2021 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202107-
divvy-tripdata.csv")
m08_2021 <- read.csv("C:/Users/Wel/Desktop/Courses/Case Study/202108-
divvy-tripdata.csv")
```

Step 3: Wrangling and combining the data into one single file

To combine the data into one single file we will need to make sure that the data is consistent. We will start by comparing the column names:

```
colnames(m09_2020)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m10_2020)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m11_2020)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m12_2020)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m01_2021)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m02_2021)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m03_2021)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m04_2021)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m05_2021)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m06_2021)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m07_2021)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
colnames(m08_2021)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

We see that the column names are consistent in all the individual files.

Now we check for any difference/incongruencies in the structures of the files.

```
str(m09_2020)
```

```
## 'data.frame':    532958 obs. of  13 variables:
##  $ ride_id           : chr  "2B22BD5F95FB2629" "A7FB70B4AFC6CAF2"
"86057FA01BAC778E" "57F6DC9A153DB98C" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2020-09-17 14:27:11" "2020-09-17
15:07:31" "2020-09-17 15:09:04" "2020-09-17 18:10:46" ...
##  $ ended_at          : chr  "2020-09-17 14:44:24" "2020-09-17
15:07:45" "2020-09-17 15:09:35" "2020-09-17 18:35:49" ...
##  $ start_station_name: chr  "Michigan Ave & Lake St" "W Oakdale Ave
& N Broadway" "W Oakdale Ave & N Broadway" "Ashland Ave & Belle Plaine
```

```
Ave" ...
##  $ start_station_id  : int  52 NA NA 246 24 94 291 NA NA NA ...
##  $ end_station_name  : chr  "Green St & Randolph St" "W Oakdale Ave
& N Broadway" "W Oakdale Ave & N Broadway" "Montrose Harbor" ...
##  $ end_station_id    : int  112 NA NA 249 24 NA 256 NA NA NA ...
##  $ start_lat         : num  41.9 41.9 41.9 42 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 42 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

```
str(m10_2020)
```

```
## 'data.frame':    388653 obs. of  13 variables:
##  $ ride_id           : chr  "ACB6B40CF5B9044C" "DF450C72FD109C01"
"B6396B54A15AC0DF" "44A4AEE261B9E854" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2020-10-31 19:39:43" "2020-10-31
23:50:08" "2020-10-31 23:00:01" "2020-10-31 22:16:43" ...
##  $ ended_at          : chr  "2020-10-31 19:57:12" "2020-11-01
00:04:16" "2020-10-31 23:08:22" "2020-10-31 22:19:35" ...
##  $ start_station_name: chr  "Lakeview Ave & Fullerton Pkwy"
"Southport Ave & Waveland Ave" "Stony Island Ave & 67th St" "Clark St
& Grace St" ...
##  $ start_station_id  : int  313 227 102 165 190 359 313 125 NA
174 ...
##  $ end_station_name  : chr  "Rush St & Hubbard St" "Kedzie Ave &
Milwaukee Ave" "University Ave & 57th St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : int  125 260 423 256 185 53 125 313 199
635 ...
##  $ start_lat         : num  41.9 41.9 41.8 42 41.9 ...
##  $ start_lng         : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.8 42 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

```
str(m11_2020)
```

```
## 'data.frame':    259716 obs. of  13 variables:
##  $ ride_id           : chr  "BD0A6FF6FFF9B921" "96A7A7A4BDE4F82D"
"C61526D06582BDC5" "E533E89C32080B9E" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2020-11-01 13:36:00" "2020-11-01
10:03:26" "2020-11-01 00:34:05" "2020-11-01 00:45:16" ...
##  $ ended_at          : chr  "2020-11-01 13:45:40" "2020-11-01
10:14:45" "2020-11-01 01:03:06" "2020-11-01 00:54:31" ...
##  $ start_station_name: chr  "Dearborn St & Erie St" "Franklin St &
Illinois St" "Lake Shore Dr & Monroe St" "Leavitt St & Chicago
Ave" ...
```

```
##  $ start_station_id  : int  110 672 76 659 2 72 76 NA 58 394 ...
##  $ end_station_name  : chr  "St. Clair St & Erie St" "Noble St &
Milwaukee Ave" "Federal St & Polk St" "Stave St & Armitage Ave" ...
##  $ end_station_id    : int  211 29 41 185 2 76 72 NA 288 273 ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

```
str(m12_2020)
```

```
## 'data.frame':    131573 obs. of  13 variables:
##  $ ride_id           : chr  "70B6A9A437D4C30D" "158A465D4E74C54A"
"5262016E0F1F2F9A" "BE119628E44F871E" ...
##  $ rideable_type     : chr  "classic_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2020-12-27 12:44:29" "2020-12-18
17:37:15" "2020-12-15 15:04:33" "2020-12-15 15:54:18" ...
##  $ ended_at          : chr  "2020-12-27 12:55:06" "2020-12-18
17:44:19" "2020-12-15 15:11:28" "2020-12-15 16:00:11" ...
##  $ start_station_name: chr  "Aberdeen St & Jackson Blvd" "" ""
"" ...
##  $ start_station_id  : chr  "13157" "" "" "" ...
##  $ end_station_name  : chr  "Desplaines St & Kinzie St" "" ""
"" ...
##  $ end_station_id    : chr  "TA1306000003" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.8 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.8 ...
##  $ end_lng           : num  -87.6 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
str(m01_2021)
```

```
## 'data.frame':    96834 obs. of  13 variables:
##  $ ride_id           : chr  "E19E6F1B8D4C42ED" "DC88F20C2C55F27F"
"EC45C94683FE3F27" "4FA453A75AE377DB" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2021-01-23 16:14:19" "2021-01-27
18:43:08" "2021-01-21 22:35:54" "2021-01-07 13:31:13" ...
##  $ ended_at          : chr  "2021-01-23 16:24:44" "2021-01-27
18:47:12" "2021-01-21 22:37:14" "2021-01-07 13:42:55" ...
##  $ start_station_name: chr  "California Ave & Cortez St"
"California Ave & Cortez St" "California Ave & Cortez St" "California
Ave & Cortez St" ...
##  $ start_station_id  : chr  "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
str(m02_2021)
```

```
## 'data.frame':    49622 obs. of  13 variables:
##  $ ride_id           : chr  "89E7AA6C29227EFF" "0FEFDE2603568365"
"E6159D746B2DBB91" "B32D3199F1C2E75B" ...
##  $ rideable_type     : chr  "classic_bike" "classic_bike"
"electric_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-02-12 16:14:56" "2021-02-14
17:52:38" "2021-02-09 19:10:18" "2021-02-02 17:49:41" ...
##  $ ended_at          : chr  "2021-02-12 16:21:43" "2021-02-14
18:12:09" "2021-02-09 19:19:10" "2021-02-02 17:54:06" ...
##  $ start_station_name: chr  "Glenwood Ave & Touhy Ave" "Glenwood
Ave & Touhy Ave" "Clark St & Lake St" "Wood St & Chicago Ave" ...
##  $ start_station_id  : chr  "525" "525" "KA1503000012" "637" ...
##  $ end_station_name  : chr  "Sheridan Rd & Columbia Ave" "Bosworth
Ave & Howard St" "State St & Randolph St" "Honore St & Division
St" ...
##  $ end_station_id    : chr  "660" "16806" "TA1305000029"
"TA1305000034" ...
##  $ start_lat         : num  42 42 41.9 41.9 41.8 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num  42 42 41.9 41.9 41.8 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr  "member" "casual" "member" "member" ...
```

```
str(m03_2021)
```

```
## 'data.frame':    228496 obs. of  13 variables:
##  $ ride_id           : chr  "CFA86D4455AA1030" "30D9DC61227D1AF3"
"846D87A15682A284" "994D05AA75A168F2" ...
##  $ rideable_type     : chr  "classic_bike" "classic_bike"
"classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-03-16 08:32:30" "2021-03-28
01:26:28" "2021-03-11 21:17:29" "2021-03-11 13:26:42" ...
##  $ ended_at          : chr  "2021-03-16 08:36:34" "2021-03-28
01:36:55" "2021-03-11 21:33:53" "2021-03-11 13:55:41" ...
##  $ start_station_name: chr  "Humboldt Blvd & Armitage Ave"
"Humboldt Blvd & Armitage Ave" "Shields Ave & 28th Pl" "Winthrop Ave &
Lawrence Ave" ...
##  $ start_station_id  : chr  "15651" "15651" "15443"
"TA1308000021" ...
##  $ end_station_name  : chr  "Stave St & Armitage Ave" "Central Park
Ave & Bloomingdale Ave" "Halsted St & 35th St" "Broadway & Sheridan
Rd" ...
##  $ end_station_id    : chr  "13266" "18017" "TA1308000043"
"13323" ...
```

```
##  $ start_lat         : num  41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

```
str(m04_2021)
```

```
## 'data.frame':    337230 obs. of  13 variables:
##  $ ride_id           : chr  "6C992BD37A98A63F" "1E0145613A209000"
"E498E15508A80BAD" "1887262AD101C604" ...
##  $ rideable_type     : chr  "classic_bike" "docked_bike"
"docked_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-04-12 18:25:36" "2021-04-27
17:27:11" "2021-04-03 12:42:45" "2021-04-17 09:17:42" ...
##  $ ended_at          : chr  "2021-04-12 18:56:55" "2021-04-27
18:31:29" "2021-04-07 11:40:24" "2021-04-17 09:42:48" ...
##  $ start_station_name: chr  "State St & Pearson St" "Dorchester Ave
& 49th St" "Loomis Blvd & 84th St" "Honore St & Division St" ...
##  $ start_station_id  : chr  "TA1307000061" "KA1503000069" "20121"
"TA1305000034" ...
##  $ end_station_name  : chr  "Southport Ave & Waveland Ave"
"Dorchester Ave & 49th St" "Loomis Blvd & 84th St" "Southport Ave &
Waveland Ave" ...
##  $ end_station_id    : chr  "13235" "KA1503000069" "20121"
"13235" ...
##  $ start_lat         : num  41.9 41.8 41.7 41.9 41.7 ...
##  $ start_lng         : num  -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.8 41.7 41.9 41.7 ...
##  $ end_lng           : num  -87.7 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "casual" "casual" "member" ...
```

```
str(m05_2021)
```

```
## 'data.frame':    531633 obs. of  13 variables:
##  $ ride_id           : chr  "C809ED75D6160B2A" "DD59FDCE0ACACAF3"
"0AB83CB88C43EFC2" "7881AC6D39110C60" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2021-05-30 11:58:15" "2021-05-30
11:29:14" "2021-05-30 14:24:01" "2021-05-30 14:25:51" ...
##  $ ended_at          : chr  "2021-05-30 12:10:39" "2021-05-30
12:14:09" "2021-05-30 14:25:13" "2021-05-30 14:41:04" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.8 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...

str(m06_2021)

## 'data.frame':    729595 obs. of  13 variables:
##  $ ride_id           : chr  "99FEC93BA843FB20" "06048DCFC8520CAF"
"9598066F68045DF2" "B03C0FE48C412214" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2021-06-13 14:31:28" "2021-06-04
11:18:02" "2021-06-04 09:49:35" "2021-06-03 19:56:05" ...
##  $ ended_at          : chr  "2021-06-13 14:34:11" "2021-06-04
11:24:19" "2021-06-04 09:55:34" "2021-06-03 20:21:55" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.8 41.8 41.8 41.8 41.8 ...
##  $ start_lng         : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.8 41.8 41.8 41.8 41.8 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...

str(m07_2021)

## 'data.frame':    822410 obs. of  13 variables:
##  $ ride_id           : chr  "0A1B623926EF4E16" "B2D5583A5A5E76EE"
"6F264597DDBF427A" "379B58EAB20E8AA5" ...
##  $ rideable_type     : chr  "docked_bike" "classic_bike"
"classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-07-02 14:44:36" "2021-07-07
16:57:42" "2021-07-25 11:30:55" "2021-07-08 22:08:30" ...
##  $ ended_at          : chr  "2021-07-02 15:19:58" "2021-07-07
17:16:09" "2021-07-25 11:48:45" "2021-07-08 22:23:32" ...
##  $ start_station_name: chr  "Michigan Ave & Washington St"
"California Ave & Cortez St" "Wabash Ave & 16th St" "California Ave &
Cortez St" ...
##  $ start_station_id  : chr  "13001" "17660" "SL-012" "17660" ...
##  $ end_station_name  : chr  "Halsted St & North Branch St" "Wood St
& Hubbard St" "Rush St & Hubbard St" "Carpenter St & Huron St" ...
##  $ end_station_id    : chr  "KA1504000117" "13432" "KA1503000044"
"13196" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "member" "member" ...

str(m08_2021)
```

```
## 'data.frame':    804352 obs. of  13 variables:
##  $ ride_id           : chr  "99103BB87CC6C1BB" "EAFCCCFB0A3FC5A1"
"9EF4F46C57AD234D" "5834D3208BFAF1DA" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2021-08-10 17:15:49" "2021-08-10
17:23:14" "2021-08-21 02:34:23" "2021-08-21 06:52:55" ...
##  $ ended_at          : chr  "2021-08-10 17:22:44" "2021-08-10
17:39:24" "2021-08-21 02:50:36" "2021-08-21 07:08:13" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.8 41.8 42 42 41.8 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  41.8 41.8 42 42 41.8 ...
##  $ end_lng           : num  -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
#comparing all the columns to check any inconsistency in the data type
compare_df_cols(m09_2020, m10_2020, m11_2020, m12_2020, m01_2021,
m02_2021, m03_2021, m04_2021,
                m05_2021, m06_2021,m07_2021, m08_2021,
return="mismatch")
```

```
##        column_name m09_2020 m10_2020 m11_2020  m12_2020  m01_2021
m02_2021
## 1   end_station_id  integer  integer  integer character character
character
## 2 start_station_id  integer  integer  integer character character
character
##    m03_2021  m04_2021  m05_2021  m06_2021  m07_2021  m08_2021
## 1 character character character character character character
## 2 character character character character character character
```

We see that the “start_station_id” and “end_station_id” data type of some data frames is int instead of chr. We will convert all of them to chr to avoid any problem.

```
m09_2020 <- mutate(m09_2020, start_station_id =
as.character(start_station_id),
                   end_station_id = as.character(end_station_id))
m10_2020 <- mutate(m10_2020, start_station_id =
as.character(start_station_id),
                   end_station_id = as.character(end_station_id))
m11_2020 <- mutate(m11_2020, start_station_id =
as.character(start_station_id),
                   end_station_id = as.character(end_station_id))
head(m12_2020)
```

```
##             ride_id rideable_type          started_at
ended_at
## 1 70B6A9A437D4C30D  classic_bike 2020-12-27 12:44:29 2020-12-27
12:55:06
## 2 158A465D4E74C54A electric_bike 2020-12-18 17:37:15 2020-12-18
17:44:19
## 3 5262016E0F1F2F9A electric_bike 2020-12-15 15:04:33 2020-12-15
15:11:28
## 4 BE119628E44F871E electric_bike 2020-12-15 15:54:18 2020-12-15
16:00:11
## 5 69AF78D57854E110 electric_bike 2020-12-22 12:08:17 2020-12-22
12:10:59
## 6 C1DECC4AB488831C electric_bike 2020-12-22 13:26:37 2020-12-22
13:34:50
##           start_station_name start_station_id
end_station_name
## 1 Aberdeen St & Jackson Blvd            13157 Desplaines St &
Kinzie St
## 2

## 3

## 4

## 5

## 6

##   end_station_id start_lat start_lng  end_lat   end_lng
member_casual
## 1   TA1306000003  41.87773 -87.65479 41.88872 -87.64445
member
## 2                 41.93000 -87.70000 41.91000 -87.70000
member
## 3                 41.91000 -87.69000 41.93000 -87.70000
member
## 4                 41.92000 -87.70000 41.91000 -87.70000
member
## 5                 41.80000 -87.59000 41.80000 -87.59000
member
## 6                 41.80000 -87.59000 41.78000 -87.60000
member
```

Now we combine all the dataframes into one dataframe.

```
total_trips <-
bind_rows(m09_2020,m10_2020,m11_2020,m12_2020,m01_2021,m02_2021,

m03_2021,m04_2021,m05_2021,m06_2021,m07_2021,m08_2021)
```

```
head(total_trips[which(total_trips$started_at == "01-12-2020
00:01"), ])
```

```
##  [1] ride_id            rideable_type      started_at
ended_at
##  [5] start_station_name start_station_id   end_station_name
end_station_id
##  [9] start_lat          start_lng          end_lat
end_lng
## [13] member_casual
## <0 rows> (or 0-length row.names)
```

```
### Step 4: To organize and format and clean the data
#Now we will inspect the data and check for any corrupt data.

colnames(total_trips)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
dim(total_trips)
```

```
## [1] 4913072      13
```

```
head(total_trips)
```

```
##            ride_id rideable_type          started_at
ended_at
## 1 2B22BD5F95FB2629 electric_bike 2020-09-17 14:27:11 2020-09-17
14:44:24
## 2 A7FB70B4AFC6CAF2 electric_bike 2020-09-17 15:07:31 2020-09-17
15:07:45
## 3 86057FA01BAC778E electric_bike 2020-09-17 15:09:04 2020-09-17
15:09:35
## 4 57F6DC9A153DB98C electric_bike 2020-09-17 18:10:46 2020-09-17
18:35:49
## 5 B9C4712F78C1AE68 electric_bike 2020-09-17 15:16:13 2020-09-17
15:52:55
## 6 378BBCE1E444EB80 electric_bike 2020-09-17 18:37:04 2020-09-17
19:23:28
##               start_station_name start_station_id
end_station_name
## 1         Michigan Ave & Lake St               52     Green St &
Randolph St
## 2     W Oakdale Ave & N Broadway             <NA> W Oakdale Ave & N
Broadway
## 3     W Oakdale Ave & N Broadway             <NA> W Oakdale Ave & N
Broadway
## 4 Ashland Ave & Belle Plaine Ave              246
```

```
Montrose Harbor
## 5       Fairbanks Ct & Grand Ave               24   Fairbanks Ct &
Grand Ave
## 6        Clark St & Armitage Ave               94

##   end_station_id start_lat start_lng  end_lat   end_lng
member_casual
## 1            112  41.88669 -87.62356 41.88357 -87.64873
casual
## 2           <NA>  41.94000 -87.64000 41.94000 -87.64000
casual
## 3           <NA>  41.94000 -87.64000 41.94000 -87.64000
casual
## 4            249  41.95606 -87.66892 41.96398 -87.63822
casual
## 5             24  41.89186 -87.62101 41.89135 -87.62032
casual
## 6           <NA>  41.91826 -87.63636 41.88000 -87.62000
casual
```

```
str(total_trips)
```

```
## 'data.frame':    4913072 obs. of  13 variables:
##  $ ride_id           : chr  "2B22BD5F95FB2629" "A7FB70B4AFC6CAF2"
"86057FA01BAC778E" "57F6DC9A153DB98C" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2020-09-17 14:27:11" "2020-09-17
15:07:31" "2020-09-17 15:09:04" "2020-09-17 18:10:46" ...
##  $ ended_at          : chr  "2020-09-17 14:44:24" "2020-09-17
15:07:45" "2020-09-17 15:09:35" "2020-09-17 18:35:49" ...
##  $ start_station_name: chr  "Michigan Ave & Lake St" "W Oakdale Ave
& N Broadway" "W Oakdale Ave & N Broadway" "Ashland Ave & Belle Plaine
Ave" ...
##  $ start_station_id  : chr  "52" NA NA "246" ...
##  $ end_station_name  : chr  "Green St & Randolph St" "W Oakdale Ave
& N Broadway" "W Oakdale Ave & N Broadway" "Montrose Harbor" ...
##  $ end_station_id    : chr  "112" NA NA "249" ...
##  $ start_lat         : num  41.9 41.9 41.9 42 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 42 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

We see that there are 4913072 rows and 13 columns, 9 of them are chr and remaining 4 are num.

Before we proceed to clean the data we will make some further changes in the data:

- Create a new column, ride_length, which we can find by taking the difference between started_at and ended_at values.
- Create another column, day_of_the_week, to mention the days in the data.
- Remove incorrect data

Create a new column, ride_length

```
total_trips$ride_length <- difftime(total_trips$ended_at,
total_trips$started_at)
head(total_trips)

##            ride_id rideable_type          started_at
ended_at
## 1 2B22BD5F95FB2629 electric_bike 2020-09-17 14:27:11 2020-09-17
14:44:24
## 2 A7FB70B4AFC6CAF2 electric_bike 2020-09-17 15:07:31 2020-09-17
15:07:45
## 3 86057FA01BAC778E electric_bike 2020-09-17 15:09:04 2020-09-17
15:09:35
## 4 57F6DC9A153DB98C electric_bike 2020-09-17 18:10:46 2020-09-17
18:35:49
## 5 B9C4712F78C1AE68 electric_bike 2020-09-17 15:16:13 2020-09-17
15:52:55
## 6 378BBCE1E444EB80 electric_bike 2020-09-17 18:37:04 2020-09-17
19:23:28
##               start_station_name start_station_id
end_station_name
## 1         Michigan Ave & Lake St               52     Green St &
Randolph St
## 2     W Oakdale Ave & N Broadway             <NA> W Oakdale Ave & N
Broadway
## 3     W Oakdale Ave & N Broadway             <NA> W Oakdale Ave & N
Broadway
## 4 Ashland Ave & Belle Plaine Ave              246
Montrose Harbor
## 5       Fairbanks Ct & Grand Ave               24   Fairbanks Ct &
Grand Ave
## 6        Clark St & Armitage Ave               94

##   end_station_id start_lat start_lng  end_lat   end_lng
member_casual
## 1            112  41.88669 -87.62356 41.88357 -87.64873
casual
## 2           <NA>  41.94000 -87.64000 41.94000 -87.64000
casual
## 3           <NA>  41.94000 -87.64000 41.94000 -87.64000
casual
## 4            249  41.95606 -87.66892 41.96398 -87.63822
casual
## 5             24  41.89186 -87.62101 41.89135 -87.62032
```

```
casual
## 6           <NA>  41.91826 -87.63636 41.88000 -87.62000
casual
##   ride_length
## 1   1033 secs
## 2     14 secs
## 3     31 secs
## 4   1503 secs
## 5   2202 secs
## 6   2784 secs
```

```
# we will convert this field to numeric so that we can do calculations
on it
total_trips$ride_length <- as.numeric(total_trips$ride_length)
```

Remove lat, long, start_station_id and end_station_id as this data is not needed.

```
total_trips <- total_trips %>%
select(-c(start_lat, start_lng, end_lat, end_lng, start_station_id,
end_station_id))
```

add columns, days, months, year and day of the week

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

```
##              ride_id rideable_type          started_at
ended_at
## 1 2B22BD5F95FB2629 electric_bike 2020-09-17 14:27:11 2020-09-17
14:44:24
## 2 A7FB70B4AFC6CAF2 electric_bike 2020-09-17 15:07:31 2020-09-17
15:07:45
## 3 86057FA01BAC778E electric_bike 2020-09-17 15:09:04 2020-09-17
15:09:35
## 4 57F6DC9A153DB98C electric_bike 2020-09-17 18:10:46 2020-09-17
18:35:49
## 5 B9C4712F78C1AE68 electric_bike 2020-09-17 15:16:13 2020-09-17
15:52:55
## 6 378BBCE1E444EB80 electric_bike 2020-09-17 18:37:04 2020-09-17
19:23:28
##                 start_station_name           end_station_name
member_casual
## 1         Michigan Ave & Lake St     Green St & Randolph St
casual
## 2     W Oakdale Ave & N Broadway W Oakdale Ave & N Broadway
casual
## 3     W Oakdale Ave & N Broadway W Oakdale Ave & N Broadway
casual
```

```
## 4 Ashland Ave & Belle Plaine Ave            Montrose Harbor
casual
## 5       Fairbanks Ct & Grand Ave   Fairbanks Ct & Grand Ave
casual
## 6        Clark St & Armitage Ave
casual
##   ride_length       date month day year day_of_week
## 1        1033 2020-09-17    09  17 2020    Thursday
## 2          14 2020-09-17    09  17 2020    Thursday
## 3          31 2020-09-17    09  17 2020    Thursday
## 4        1503 2020-09-17    09  17 2020    Thursday
## 5        2202 2020-09-17    09  17 2020    Thursday
## 6        2784 2020-09-17    09  17 2020    Thursday
```

Inspecting the new table that has been created:

```
str(total_trips)

## 'data.frame':    4913072 obs. of  13 variables:
##  $ ride_id           : chr  "2B22BD5F95FB2629" "A7FB70B4AFC6CAF2"
"86057FA01BAC778E" "57F6DC9A153DB98C" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2020-09-17 14:27:11" "2020-09-17
15:07:31" "2020-09-17 15:09:04" "2020-09-17 18:10:46" ...
##  $ ended_at          : chr  "2020-09-17 14:44:24" "2020-09-17
15:07:45" "2020-09-17 15:09:35" "2020-09-17 18:35:49" ...
##  $ start_station_name: chr  "Michigan Ave & Lake St" "W Oakdale Ave
& N Broadway" "W Oakdale Ave & N Broadway" "Ashland Ave & Belle Plaine
Ave" ...
##  $ end_station_name  : chr  "Green St & Randolph St" "W Oakdale Ave
& N Broadway" "W Oakdale Ave & N Broadway" "Montrose Harbor" ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
##  $ ride_length       : num  1033 14 31 1503 2202 ...
##  $ date              : Date, format: "2020-09-17" "2020-09-17" ...
##  $ month             : chr  "09" "09" "09" "09" ...
##  $ day               : chr  "17" "17" "17" "17" ...
##  $ year              : chr  "2020" "2020" "2020" "2020" ...
##  $ day_of_week       : chr  "Thursday" "Thursday" "Thursday"
"Thursday" ...

dim(total_trips)

## [1] 4913072      13

colnames(total_trips)

##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "end_station_name"
##  [7] "member_casual"      "ride_length"        "date"
## [10] "month"              "day"                "year"
## [13] "day_of_week"
```

```
nrow(total_trips)

## [1] 4913072

summary(total_trips)

##    ride_id          rideable_type       started_at
ended_at
##  Length:4913072     Length:4913072     Length:4913072
Length:4913072
##  Class :character   Class :character   Class :character
Class :character
##  Mode  :character   Mode  :character   Mode  :character
Mode  :character
##

##

##

##  start_station_name end_station_name   member_casual
ride_length
##  Length:4913072     Length:4913072     Length:4913072     Min.   :-
1742998
##  Class :character   Class :character   Class :character   1st Qu.:
431
##  Mode  :character   Mode  :character   Mode  :character   Median :
768
##                                                           Mean   :
1269
##                                                           3rd Qu.:
1396
##                                                           Max.   :
3356649
##       date               month              day
year
##  Min.   :2020-09-01   Length:4913072     Length:4913072
Length:4913072
##  1st Qu.:2020-12-09   Class :character   Class :character
Class :character
##  Median :2021-05-26   Mode  :character   Mode  :character
Mode  :character
##  Mean   :2021-04-10

##  3rd Qu.:2021-07-17

##  Max.   :2021-08-31

##  day_of_week
##  Length:4913072
##  Class :character
##  Mode  :character
##
##
##
```

We can check if there is any null value in the data frame. We will take that out using the drop_na() function.

```
paste("Number of Rows",nrow(total_trips))

## [1] "Number of Rows 4913072"

paste("Number of Missing Values", sum(is.na(total_trips)))

## [1] "Number of Missing Values 0"

total_trips <-total_trips %>%
  drop_na()
paste("Number of Missing Values", sum(is.na(total_trips)))

## [1] "Number of Missing Values 0"

paste("Number of Rows",nrow(total_trips))

## [1] "Number of Rows 4913072"
```

Removing bad data from the table

We see that the ride_length is negative for some observations, that is because the ride was taken out of docks to check for quality and put it back in later. We will remove this negative readings. Since we are removing data we will create a new data frame; total_trips_v2

```
total_trips_v2 <- total_trips[!(total_trips$ride_length<0),]
head(total_trips_v2)

##            ride_id rideable_type          started_at
ended_at
## 1 2B22BD5F95FB2629 electric_bike 2020-09-17 14:27:11 2020-09-17
14:44:24
## 2 A7FB70B4AFC6CAF2 electric_bike 2020-09-17 15:07:31 2020-09-17
15:07:45
## 3 86057FA01BAC778E electric_bike 2020-09-17 15:09:04 2020-09-17
15:09:35
## 4 57F6DC9A153DB98C electric_bike 2020-09-17 18:10:46 2020-09-17
18:35:49
## 5 B9C4712F78C1AE68 electric_bike 2020-09-17 15:16:13 2020-09-17
15:52:55
## 6 378BBCE1E444EB80 electric_bike 2020-09-17 18:37:04 2020-09-17
19:23:28
##               start_station_name           end_station_name
member_casual
```

```
## 1         Michigan Ave & Lake St     Green St & Randolph St
casual
## 2     W Oakdale Ave & N Broadway W Oakdale Ave & N Broadway
casual
## 3     W Oakdale Ave & N Broadway W Oakdale Ave & N Broadway
casual
## 4 Ashland Ave & Belle Plaine Ave            Montrose Harbor
casual
## 5       Fairbanks Ct & Grand Ave   Fairbanks Ct & Grand Ave
casual
## 6        Clark St & Armitage Ave
casual
##   ride_length       date month day year day_of_week
## 1        1033 2020-09-17    09  17 2020    Thursday
## 2          14 2020-09-17    09  17 2020    Thursday
## 3          31 2020-09-17    09  17 2020    Thursday
## 4        1503 2020-09-17    09  17 2020    Thursday
## 5        2202 2020-09-17    09  17 2020    Thursday
## 6        2784 2020-09-17    09  17 2020    Thursday
```

Step 5: Conducting descriptive analysis on the data.

Calculating summary statistics and performing calculations to identify trends and relationships.

```
# converting ride_length into numeric value so that we can perform
calculations on it.
is.factor(total_trips_v2$ride_length)

## [1] FALSE

total_trips_v2$ride_length <-
as.numeric(as.character(total_trips_v2$ride_length))
is.numeric(total_trips_v2$ride_length)

## [1] TRUE

mean(total_trips_v2$ride_length) #calculating average ride_length of
the user.

## [1] 1402.285

median(total_trips_v2$ride_length) #midpoint number in the ascending
array of ride lengths

## [1] 769

max(total_trips_v2$ride_length) #longest ride

## [1] 3356649

min(total_trips_v2$ride_length) #shortest ride
```

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

```
#above findings can also be found using the summary function.
summary(total_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0     432     769    1402    1397 3356649
```

i)finding comparisons in terms of ride_length in respect with causal and annual members.

```
aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual, 
FUN=mean)
```

```
##   total_trips_v2$member_casual total_trips_v2$ride_length
## 1                       casual                  2053.1116
## 2                       member                   863.0715
```

```
aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual, 
FUN=median)
```

```
##   total_trips_v2$member_casual total_trips_v2$ride_length
## 1                       casual                       1029
## 2                       member                        615
```

```
aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual, 
FUN=max)
```

```
##   total_trips_v2$member_casual total_trips_v2$ride_length
## 1                       casual                    3356649
## 2                       member                    1870176
```

```
aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual, 
FUN=min)
```

```
##   total_trips_v2$member_casual total_trips_v2$ride_length
## 1                       casual                          0
## 2                       member                          0
```

ii)finding the ride_length in terms of days of the week in respect to member types:

```
aggregate(total_trips_v2$ride_length ~ 
total_trips_v2$member_casual+total_trips_v2$day_of_week, FUN=mean)
```

```
##    total_trips_v2$member_casual total_trips_v2$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
##    total_trips_v2$ride_length
## 1                   1961.7298
## 2                    851.2822
## 3                   2027.7556
## 4                    830.4723
## 5                   2217.5657
## 6                    956.9700
## 7                   2370.3948
## 8                    982.9900
## 9                   1776.9537
## 10                   808.1291
## 11                  1818.9634
## 12                   812.3113
## 13                  1827.9926
## 14                   816.9993
```

```
#printing it in order with respect to days of the week:
total_trips_v2$day_of_week <- ordered(total_trips_v2$day_of_week,
levels=c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","
Saturday"))
aggregate(total_trips_v2$ride_length ~ total_trips_v2$member_casual +
total_trips_v2$day_of_week, FUN = mean)
```

```
##    total_trips_v2$member_casual total_trips_v2$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
##    total_trips_v2$ride_length
## 1                   2370.3948
## 2                    982.9900
## 3                   2027.7556
## 4                    830.4723
## 5                   1818.9634
## 6                    812.3113
## 7                   1827.9926
## 8                    816.9993
## 9                   1776.9537
## 10                   808.1291
## 11                  1961.7298
## 12                   851.2822
## 13                  2217.5657
## 14                   956.9700
```

```
tail(total_trips_v2)
```

```
##                  ride_id rideable_type          started_at
ended_at
## 4913067 2D6861BE1B6741CF  classic_bike 2021-08-07 10:52:09 2021-08-
07 10:58:09
## 4913068 5E5C9CD681E0419C  classic_bike 2021-08-07 18:07:43 2021-08-
07 18:21:21
## 4913069 96FB57CF4AA456F6 electric_bike 2021-08-09 08:49:31 2021-08-
09 09:03:51
## 4913070 226A0910DCCE904C  classic_bike 2021-08-12 16:55:57 2021-08-
12 17:15:10
## 4913071 1A97D27AE23DE1E7  classic_bike 2021-08-08 22:47:43 2021-08-
08 23:08:12
## 4913072 BBC36E4AA3652361 electric_bike 2021-08-27 18:53:53 2021-08-
27 19:02:16
##               start_station_name            end_station_name
member_casual
## 4913067  Paulina Ave & North Ave      Leavitt St & North Ave
member
## 4913068 Wells St & Evergreen Ave Lincoln Ave & Diversey Pkwy
member
## 4913069   Broadway & Sheridan Rd      Clark St & Lincoln Ave
member
## 4913070   Dearborn St & Adams St      Clark St & Lincoln Ave
member
## 4913071   Broadway & Sheridan Rd     Clark St & Winnemac Ave
casual
## 4913072  Paulina Ave & North Ave       Dayton St & North Ave
casual
##         ride_length       date month day year day_of_week
## 4913067         360 2021-08-07    08  07 2021    Saturday
## 4913068         818 2021-08-07    08  07 2021    Saturday
## 4913069         860 2021-08-09    08  09 2021      Monday
## 4913070        1153 2021-08-12    08  12 2021    Thursday
## 4913071        1229 2021-08-08    08  08 2021      Sunday
## 4913072         503 2021-08-27    08  27 2021      Friday
```

iii) Finding number of rides with respect to weekend vs weekdays:

```
total_trips_v2 %>%
  mutate(day_type = ifelse(day_of_week %in%
c("Saturday","Sunday"),"Weekend","Weekday")) %>%
  group_by(member_casual, day_type) %>%
  summarize(number_of_rides = n())

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

## # A tibble: 4 x 3
## # Groups:   member_casual [2]
##   member_casual day_type number_of_rides
##   <chr>         <chr>              <int>
## 1 casual        Weekday          1304548
## 2 casual        Weekend           919146
## 3 member        Weekday          1957128
## 4 member        Weekend           726850
```

iv) Finding ride_length with respect to weekend vs weekdays:

```
rides_per_weekend <- total_trips_v2 %>%
    # create variable to indicate weekend or not (check the weekend
day names)
    mutate(day_type = ifelse(day_of_week %in% c("Saturday", "Sunday"),
"WEEKEND","WEEK")) %>%
    # build gouping by member type and day type
    group_by(total_trips_v2$member_casual, day_type) %>%
    # summarise total ride length
    summarize(total_ride_length = sum(ride_length, na.rm = TRUE))

## `summarise()` has grouped output by 'total_trips_v2$member_casual'.
You can override using the `.groups` argument.
```

v) Finding number of rides for everyday with respect to member type:

```
# analyze ridership data by type and weekday
total_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>% #creates
weekday field using
  group_by(member_casual, weekday) %>% #groups by usertype and weekday
  summarise(number_of_rides = n(), #calculates the number of rides and
average duration
  average_duration = mean(ride_length)) %>% # calculates the average
duration
  arrange(member_casual, weekday) # sorts

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

## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              420361            2370.
##  2 casual        Mon              250462            2028.
##  3 casual        Tue              242221            1819.
##  4 casual        Wed                241825            1828.
##  5 casual        Thu                247887            1777.
##  6 casual        Fri                322153            1962.
##  7 casual        Sat                498785            2218.
##  8 member        Sun                337186             983.
##  9 member        Mon                364532             830.
## 10 member        Tue                399923             812.
## 11 member        Wed                406416             817.
## 12 member        Thu                390381             808.
## 13 member        Fri                395876             851.
## 14 member        Sat                389664             957.
```

vi) Finding number of rides for every month with respect member types:

```
total_trips_v2 %>%
  group_by(member_casual, month) %>%
  arrange(member_casual,month) %>%
  arrange(month, member_casual) %>%
  summarize(number_of_rides = n())
```

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

```
## # A tibble: 24 x 3
## # Groups:   member_casual [2]
##    member_casual month number_of_rides
##    <chr>         <chr>           <int>
##  1 casual        01              18117
##  2 casual        02              10131
##  3 casual        03              84032
##  4 casual        04             136601
##  5 casual        05             256916
##  6 casual        06             370678
##  7 casual        07             442048
##  8 casual        08             412662
##  9 casual        09             230072
## 10 casual        10             144529
## # ... with 14 more rows
```

```
rides_per_month <-total_trips_v2 %>%
  mutate(month = month(started_at, label = TRUE)) %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()
  ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, month)
```

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

vii) Finding the top 6 most used started and end stations for casual users:

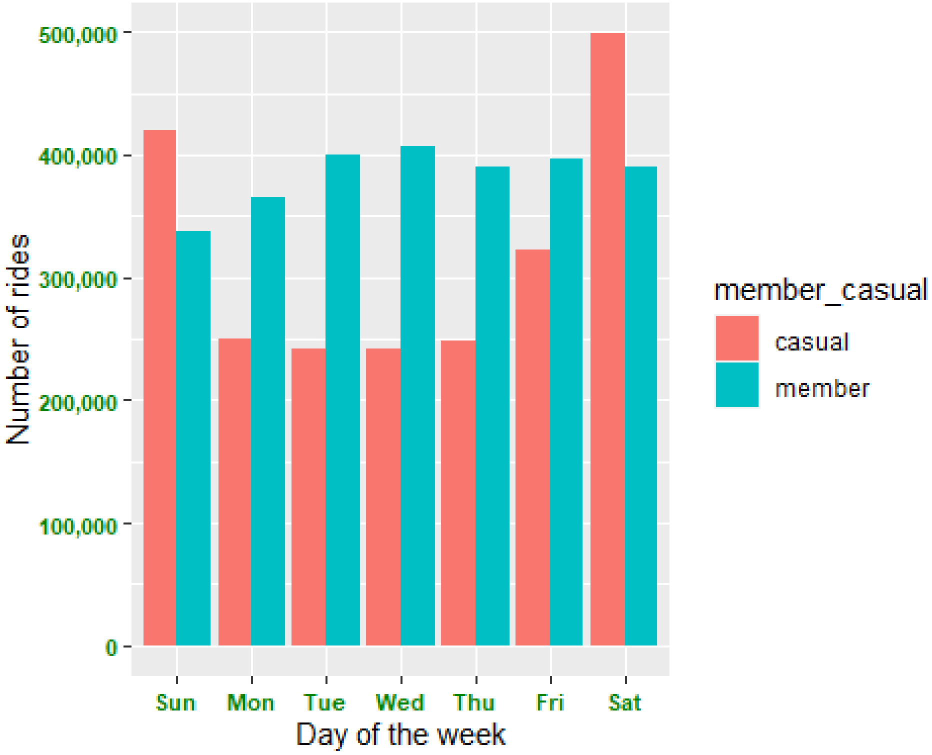
```
top_5_start_stations <- total_trips_v2 %>%
  group_by(member_casual="casual", start_station_name) %>%
  summarize(number_of_rides =n()) %>%
  arrange(desc(number_of_rides)) %>%
  head()

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

# similarly top 6 most end stations:
top_5_end_stations<- total_trips_v2 %>%
  group_by(member_casual="casual", end_station_name) %>%
  summarize(number_of_rides =n()) %>%
  arrange(desc(number_of_rides)) %>%
  head()

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

which(is.na(total_trips_v2$end_station_name), arr.ind=TRUE)

## integer(0)
```

Using the analysed data for Vizualisations:

i) visualize the number of rides by rider type

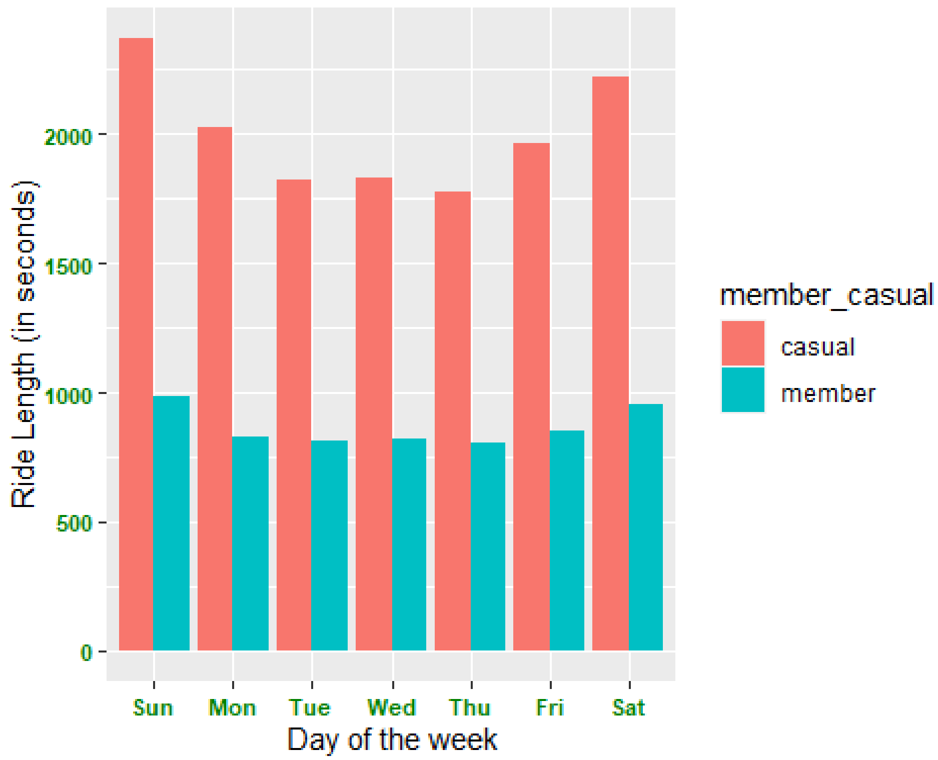
```
library(scales)

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

total_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),average_duration =
mean(ride_length)) %>%
  arrange(member_casual, weekday) %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual))
+
  geom_col(position = "dodge")+ scale_y_continuous(labels=comma,
name="Number of rides") +
  scale_x_discrete(name="Day of the week") +
theme(axis.text.x = element_text(face="bold", color="#008000",
                                 size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                                 size=8, angle=0))
```

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



ii)visualize the ride length by rider type

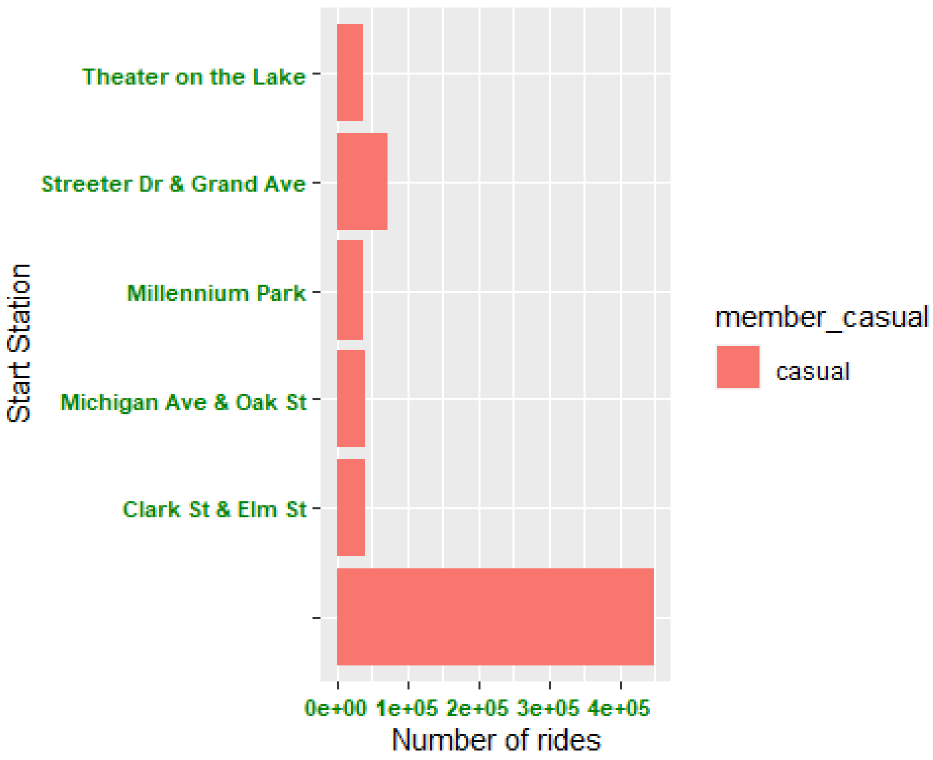
```
total_trips_v2 %>%
mutate(weekday = wday(started_at, label = TRUE)) %>%
group_by(member_casual, weekday) %>%
summarise(number_of_rides = n()
,average_duration = mean(ride_length)) %>%
arrange(member_casual, weekday) %>%
ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
geom_col(position = "dodge")+ scale_y_continuous(name="Ride Length (in
seconds)") +
  scale_x_discrete(name="Day of the week") +
theme(axis.text.x = element_text(face="bold", color="#008000",
                           size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                           size=8, angle=0))

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



iii)visualize the top 5 start stations of rides by rider type

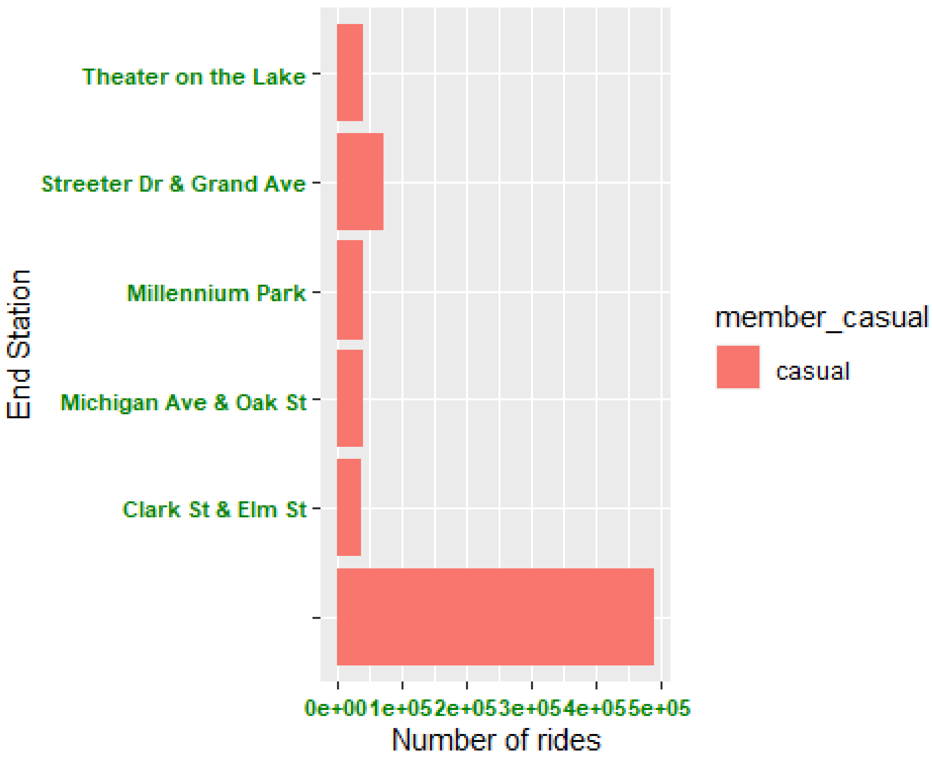
```
#creating table for the top 5 start station used by casual members.
                  head(total_trips_v2 %>%
                               group_by(member_casual="casual",
start_station_name) %>%
                               summarize(number_of_rides =n()) %>%
                               arrange(desc(number_of_rides))) %>%
ggplot(aes(x = start_station_name, y = number_of_rides, fill =
member_casual)) +
geom_col(position = "dodge")  +
coord_flip() + scale_y_continuous(name="Number of rides") +
  scale_x_discrete(name="Start Station") +
theme(axis.text.x = element_text(face="bold", color="#008000",
                           size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                           size=8, angle=0))
```

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



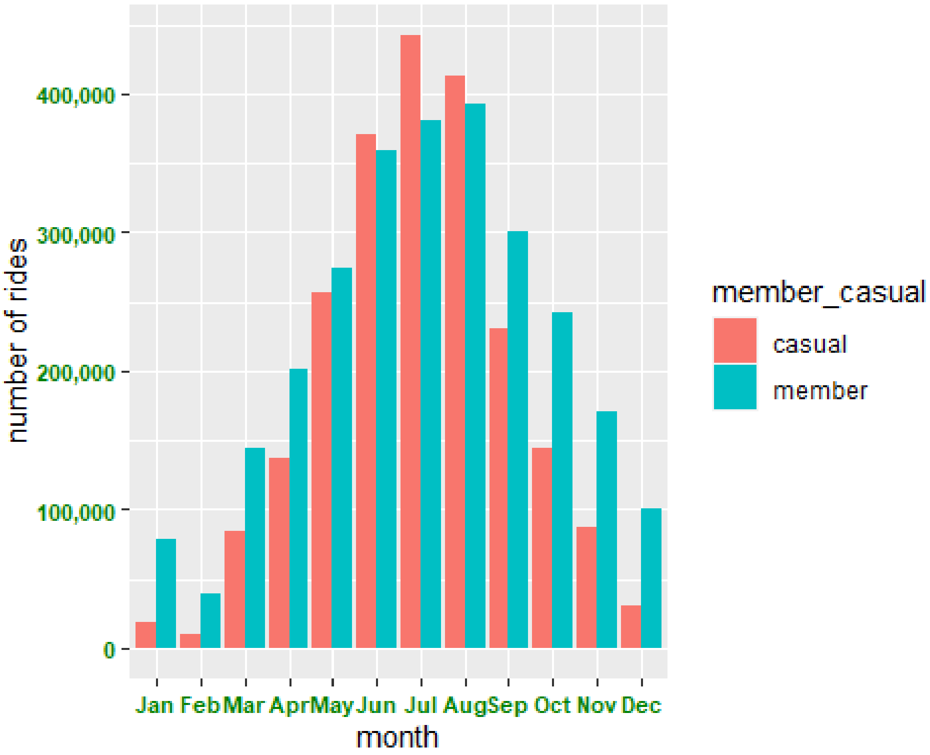
```
#creating table for the top 5 end station used by casual members.
                    head(total_trips_v2 %>%
                             group_by(member_casual="casual",
end_station_name) %>%
                             summarize(number_of_rides =n()) %>%
                             arrange(desc(number_of_rides))) %>%
ggplot(aes(x = end_station_name, y = number_of_rides, fill =
member_casual)) +
geom_col(position = "dodge")+
  coord_flip() + scale_y_continuous(name="Number of rides") +
  scale_x_discrete(name="End Station") +
theme(axis.text.x = element_text(face="bold", color="#008000",
                           size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                           size=8, angle=0))
```

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



```
#Finding number of rides for every month with respect member types:
   total_trips_v2 %>%
  mutate(month = month(started_at, label = TRUE)) %>%
  group_by(member_casual, month) %>%
  summarize(number_of_rides = n()) %>%
ggplot(aes(x = month, y = number_of_rides, fill = member_casual)) +
geom_col(position = "dodge")+ scale_y_continuous(labels=comma, name=
"number of rides") +
theme(axis.text.x = element_text(face="bold", color="#008000",
                           size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                           size=8, angle=0))
```

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



Step 6: Sharing the data (Act)

```
# Creating a csv file that we will visualize in Excel and Tableu.
counts <- aggregate(total_trips_v2$ride_length ~
total_trips_v2$member_casual +
total_trips_v2$day_of_week, FUN = mean)

#write.csv(counts, "C:/Users/Wel/Desktop/Courses/Case
Study/avg_ride_length.csv")
#write.csv(top_5_start_stations, "C:/Users/Wel/Desktop/Courses/Case
Study/start_stations.csv")
#write.csv(top_5_end_stations, "C:/Users/Wel/Desktop/Courses/Case
Study/end_stations.csv")
#write.csv(rides_per_weekend, "C:/Users/Wel/Desktop/Courses/Case
Study/rides_per_weekend.csv")
#write.csv(rides_per_month, "C:/Users/Wel/Desktop/Courses/Case
Study/rides_per_month.csv")
```

Looking into the data visualizations we can observe the following things:

1. Number of rides spike up during the weekends for the casual members where as annual members remain consistent through out the week.
2. We can see the top most picked up and dropped of stations.
3. Number of rides touch their peak during the July month.

4. Ride length also is also the highest at the weekends but almost consistent throughout the week.

5. Ride length is consistent for the annual members through out the week indicating that they are mostly using it for daily commute as going to work.

6. We see a steep fall in the number of rides for annual members when the weekend starts, stating again that their major use is to commute to work.

Recommendations based on the analysis:

1.We should have a weekend plan for the casual members where they can use the bikes just for the weekend at an affordable rates.

2.The top 5 start and end station for the casual members should be advertised even more for annual membership as those are the place where casual members are there the most.

3.June and July months overall should be used more for promotion in general as it observes the most use of the bikes.

4.Seasonal membership can also be offered as we see rise in usage during the summer season.

5.Since we see that the ride length is almost consistent through out the week for casual members, we can have some ride duration based subscription which will be at discounted price for per hour or per two hours per session, encouraging them to use it more often.

6. As we see that the casual riders use the service more on the weekends, we can offer some discounted subscription for them if they use it on the weekdays, urging them to use it more on the weekdays.