CYCLIST BIKE SHARE

Ajay Shankar A

2023-11-14

Table of contents

# 1. CYCLIST BIKE SHARE

## 1.1 Introduction

The analysis is done on **Cyclist Trip Data** obtained from *Coursera Google Data Analytics* course as part of Cap Stone Project.

The data contains month wise travel usage of bikes from the year of 2015-2023. We will be concentrating on data gathered in between *July-2022* to *June-2023* which will cover an entire year.

Let’s load the required packages first

* Loading tidyverse and gt packages

library(tidyverse)  
library(gt)

### 1.1.1 Loading and Formatting Data

* Let’s look at the structure of the data in one of the downloaded .csv files.

trpdata\_july\_2022<-read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202207-divvy-tripdata/202207-divvy-tripdata.csv")

Rows: 823488 Columns: 13  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (7): ride\_id, rideable\_type, start\_station\_name, start\_station\_id, end\_...  
dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
dttm (2): started\_at, ended\_at  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

glimpse(trpdata\_july\_2022)

Rows: 823,488  
Columns: 13  
$ ride\_id <chr> "954144C2F67B1932", "292E027607D218B6", "5776585258…  
$ rideable\_type <chr> "classic\_bike", "classic\_bike", "classic\_bike", "cl…  
$ started\_at <dttm> 2022-07-05 08:12:47, 2022-07-26 12:53:38, 2022-07-…  
$ ended\_at <dttm> 2022-07-05 08:24:32, 2022-07-26 12:55:31, 2022-07-…  
$ start\_station\_name <chr> "Ashland Ave & Blackhawk St", "Buckingham Fountain …  
$ start\_station\_id <chr> "13224", "15541", "15541", "15541", "TA1307000117",…  
$ end\_station\_name <chr> "Kingsbury St & Kinzie St", "Michigan Ave & 8th St"…  
$ end\_station\_id <chr> "KA1503000043", "623", "623", "TA1307000164", "TA13…  
$ start\_lat <dbl> 41.90707, 41.86962, 41.86962, 41.86962, 41.89147, 4…  
$ start\_lng <dbl> -87.66725, -87.62398, -87.62398, -87.62398, -87.626…  
$ end\_lat <dbl> 41.88918, 41.87277, 41.87277, 41.79526, 41.93625, 4…  
$ end\_lng <dbl> -87.63851, -87.62398, -87.62398, -87.59647, -87.652…  
$ member\_casual <chr> "member", "casual", "casual", "casual", "member", "…

* Let’s look at the columns and try to understand what they represent
  + ride\_id is the unique identification token generated for each ride that was initiated.
  + rideable\_type indicates the type of bike used for the ride.
  + started\_at and ended\_at give us the time when the ride began and the ride ended respectively.
  + start\_station\_name and end\_station\_name give us the names of stations where ride began and ended respectively.
  + start\_station\_id and end\_station\_id are unique ID’s given to stations.
  + start\_lat and start\_lng represent co-ordinates where the ride began.
  + end\_lat and end\_lng represent co-ordinates where the ride stopped.
  + member\_casual identifies if the rider is a member or casual rider of the bike.

The trpdata\_july\_2022 contains 823488 rows and 13 columns. In the results we can see all the columns and their data types.

* **Lets load data of remaining 11 months.**

trpdata\_aug\_2022 <- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202208-divvy-tripdata/202208-divvy-tripdata.csv")  
  
trpdata\_sept\_2022<- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202209-divvy-tripdata/202209-divvy-publictripdata.csv")  
  
trpdata\_oct\_2022<- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202210-divvy-tripdata/202210-divvy-tripdata\_raw.csv")  
  
trpdata\_nov\_2022<- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202211-divvy-tripdata/202211-divvy-tripdata.csv")  
  
trpdata\_dec\_2022 <- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202212-divvy-tripdata/202212-divvy-tripdata.csv")  
  
trpdata\_jan\_2023 <- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202301-divvy-tripdata/202301-divvy-tripdata.csv")  
  
trpdata\_feb\_2023 <- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202302-divvy-tripdata/202302-divvy-tripdata.csv")  
  
trpdata\_mar\_2023 <- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202303-divvy-tripdata/202303-divvy-tripdata.csv")  
  
trpdata\_apr\_2023 <- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202304-divvy-tripdata/202304-divvy-tripdata.csv")  
  
trpdata\_may\_2023 <- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202305-divvy-tripdata/202305-divvy-tripdata.csv")  
  
trpdata\_june\_2023 <- read\_csv("F:/Data\_Sci/Cap\_Stone\_Project/Cyclist\_trip\_data/202306-divvy-tripdata/202306-divvy-tripdata.csv")

As structure of .csv’s is same across the all the files lets combine all the .csv files into a single data frame which contains data of all 12 months.

* **Combining all the monthly data to one previous year data(data\_one\_year\_raw).**

data\_one\_year\_raw <- rbind(trpdata\_july\_2022, trpdata\_aug\_2022,  
 trpdata\_sept\_2022, trpdata\_oct\_2022,  
 trpdata\_nov\_2022, trpdata\_dec\_2022,  
 trpdata\_jan\_2023, trpdata\_feb\_2023,  
 trpdata\_mar\_2023, trpdata\_apr\_2023,  
 trpdata\_may\_2023, trpdata\_june\_2023)  
  
glimpse(data\_one\_year\_raw)

Rows: 5,779,444  
Columns: 13  
$ ride\_id <chr> "954144C2F67B1932", "292E027607D218B6", "5776585258…  
$ rideable\_type <chr> "classic\_bike", "classic\_bike", "classic\_bike", "cl…  
$ started\_at <dttm> 2022-07-05 08:12:47, 2022-07-26 12:53:38, 2022-07-…  
$ ended\_at <dttm> 2022-07-05 08:24:32, 2022-07-26 12:55:31, 2022-07-…  
$ start\_station\_name <chr> "Ashland Ave & Blackhawk St", "Buckingham Fountain …  
$ start\_station\_id <chr> "13224", "15541", "15541", "15541", "TA1307000117",…  
$ end\_station\_name <chr> "Kingsbury St & Kinzie St", "Michigan Ave & 8th St"…  
$ end\_station\_id <chr> "KA1503000043", "623", "623", "TA1307000164", "TA13…  
$ start\_lat <dbl> 41.90707, 41.86962, 41.86962, 41.86962, 41.89147, 4…  
$ start\_lng <dbl> -87.66725, -87.62398, -87.62398, -87.62398, -87.626…  
$ end\_lat <dbl> 41.88918, 41.87277, 41.87277, 41.79526, 41.93625, 4…  
$ end\_lng <dbl> -87.63851, -87.62398, -87.62398, -87.59647, -87.652…  
$ member\_casual <chr> "member", "casual", "casual", "casual", "member", "…

* data\_one\_year\_raw data frame contains data from the month of July-2022 to June-2023.

### 1.1.2 Cleaning the data

* Checking and counting “NA” in each column of the data frame. Data is much better without “NA” as they can cause problems while aggregating data and calculating averages and sums. We can use map function to perform a function to all of the columns.

na\_in\_cols <- data\_one\_year\_raw %>% map(is.na) %>% map(sum) %>% unlist()  
  
na\_in\_cols

ride\_id rideable\_type started\_at ended\_at   
 0 0 0 0   
start\_station\_name start\_station\_id end\_station\_name end\_station\_id   
 857860 857992 915655 915796   
 start\_lat start\_lng end\_lat end\_lng   
 0 0 5795 5795   
 member\_casual   
 0

* As NA’s are not present in the times columns i.e, started\_at and ended\_at we don’t need to worry ourselves about writing na.rm during aggregation and manipulation of data but it is a good practice to do so.
* Finding the length or duration of the rides by making a new column ride\_length in minutes and making sure that the ride\_length is not negative by using if\_else function. Eliminating stations where station names and longitude and latitude co-ordinates are not present.

# As we remove all the NA's it is better to save the data as "data\_one\_year".  
data\_one\_year <- data\_one\_year\_raw %>%   
 mutate(ride\_length = difftime(ended\_at, started\_at,  
 units = "min")) %>%  
 mutate(ride\_length = as.numeric(ride\_length))  
  
data\_one\_year <- data\_one\_year %>%   
 mutate(ride\_length = if\_else(ride\_length < 0, 0, ride\_length)) %>%   
 filter(start\_station\_name != "" & end\_station\_name != "" &   
 !is.na(start\_lat) & !is.na(start\_lng) &  
 !is.na(end\_lat) & !is.na(end\_lng)) %>% arrange(ride\_length)  
  
  
glimpse(data\_one\_year)

Rows: 4,409,335  
Columns: 14  
$ ride\_id <chr> "86CD09DA24761714", "27024CD08288BD45", "029D853B5C…  
$ rideable\_type <chr> "electric\_bike", "electric\_bike", "classic\_bike", "…  
$ started\_at <dttm> 2022-07-20 16:21:48, 2022-07-30 23:42:46, 2022-07-…  
$ ended\_at <dttm> 2022-07-20 16:21:48, 2022-07-30 23:42:46, 2022-07-…  
$ start\_station\_name <chr> "Racine Ave & Fullerton Ave", "Albany Ave & 26th St…  
$ start\_station\_id <chr> "TA1306000026", "15691", "chargingstx5", "chargings…  
$ end\_station\_name <chr> "Racine Ave & Fullerton Ave", "Albany Ave & 26th St…  
$ end\_station\_id <chr> "TA1306000026", "15691", "chargingstx5", "chargings…  
$ start\_lat <dbl> 41.92556, 41.84452, 41.94335, 41.94335, 41.94335, 4…  
$ start\_lng <dbl> -87.65859, -87.70209, -87.67067, -87.67067, -87.670…  
$ end\_lat <dbl> 41.92556, 41.84448, 41.94335, 41.94335, 41.94335, 4…  
$ end\_lng <dbl> -87.65840, -87.70201, -87.67067, -87.67067, -87.670…  
$ member\_casual <chr> "member", "casual", "member", "member", "casual", "…  
$ ride\_length <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

## 1.2 Analysis of Data

### 1.2.1 Aggregating data by Rider type and Bike type.

* Aggregating data to see **“Average minutes per ride”** grouped by “bike type” and “rider type” after removing rides less than 2 minutes (As rides less than 2 minutes tend to have the same start and stop stations).

data\_one\_year\_aggregate <- data\_one\_year %>%   
 select(ride\_id, rideable\_type, member\_casual, started\_at, ended\_at,  
 ride\_length, everything()) %>%  
 filter(ride\_length >= 2) %>%   
 summarise("Number of Rides" = n(),  
 "Ride Length" = sum(ride\_length, na.rm = TRUE),  
 "Max Ride Length" = round(max(ride\_length), 2),  
 "Avg Ride Length in Minutes" = round(mean(ride\_length), 2),  
 .by = c(member\_casual, rideable\_type)) %>%   
 arrange(desc("Avg Ride Length in Minutes")) %>%   
 gt() %>% tab\_header(title = "Average length of Rides") %>%   
 cols\_label(member\_casual = "Rider type",  
 rideable\_type = "Bike type")  
  
data\_one\_year\_aggregate

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 1: Average minutes per ride  Table 1: Average length of Rides   | Rider type | Bike type | Number of Rides | Ride Length | Max Ride Length | Avg Ride Length in Minutes | | --- | --- | --- | --- | --- | --- | | member | classic\_bike | 1630991 | 21996488 | 1497.87 | 13.49 | | casual | classic\_bike | 781530 | 19383358 | 1497.75 | 24.80 | | casual | electric\_bike | 709649 | 11372659 | 479.98 | 16.03 | | member | electric\_bike | 984688 | 10968684 | 480.00 | 11.14 | | casual | docked\_bike | 136794 | 6899998 | 32035.45 | 50.44 | |

We can clearly notice in [Table 1](#tbl-avg_ride_legnth) that **member** riders have more number of rides with both **classic and electric bikes** while the average ride length is higher with **casual** riders.

* Calculating and visualizing **Average ride length** by “Rider type”.

average\_ride\_by\_rideable\_type <- data\_one\_year %>%  
 rename("Rider type" = member\_casual, "Bike type" = rideable\_type) %>%   
 summarise(ride\_length = sum(ride\_length, na.rm = TRUE),  
 ride\_count = n(),  
 avg\_ride\_length = ride\_length/ride\_count,  
 .by = c(`Rider type`, `Bike type`)) %>%   
 ggplot(aes(`Rider type`, avg\_ride\_length)) +   
 geom\_col(aes(fill = `Bike type`), position = "dodge") +   
 labs(x = "Bike type", y = "Avg Length of Ride(Minutes)",  
 title = "Average ride length by Bike type") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(size = 18),  
 legend.position = "bottom")  
  
average\_ride\_by\_rideable\_type

|  |
| --- |
| Figure 1: Average Ride Length by Rider type and Member type |

The above [Figure 1](#fig-avg_ride_length) clearly shows that members average ride lengths between bike types doesn’t differ much for **member** riders but differs with **casual** riders upto **8 minutes**.

|  |
| --- |
| Note |
| Further down in the analysis “docked\_bike” type is dropped as no proper documentation is available in the course. |

### 1.2.2 Analysing data by Time of the year and Ride Length

#### Ride Patterns Across the Weeks and Months of the Year

* Calculating and visualizing ride patterns in a week for number of rides.

rideable\_order <- c("classic\_bike", "electric\_bike", "docked\_bike")  
  
rides\_on\_days <- data\_one\_year %>%  
 filter(rideable\_type != "docked\_bike") %>%  
 mutate(month = month(started\_at, label = TRUE,  
 abbr = FALSE)) %>%   
 mutate(rideable\_type = factor(rideable\_type,  
 levels = rideable\_order)) %>% ggplot(aes(wday(started\_at, label = TRUE, abbr = FALSE))) +   
 geom\_bar(aes(fill = member\_casual), position = "dodge") +  
 facet\_wrap(~month, nrow = 3) +   
 labs(x = "Day of the Week", y = "Number of rides",  
 title = "Riding pattrens on Weekdays of each Month",  
 subtitle = "From July-2022 to June-2023",  
 fill = "Type of Rider") +  
 theme\_light() +  
 theme(legend.position = "top",  
 axis.text.x = element\_text(angle = 45, hjust = 1),  
 plot.title = element\_text(size = 18))  
  
rides\_on\_days

|  |
| --- |
| Figure 2: Riding pattrens in Weekdays of each Month |

The above [Figure 2](#fig-Ride_patterns_in_weekdays) clearly shows how the number of rides change due to seasons. In winters the number of rides decrease very drastically may be because of temperature and snow. In Summers the number of rides are at its peak.

The number of rides driven by **member** riders are increases through the week especially in working week days but for casual riders the rides increase in the weekends. The [Figure 2](#fig-Ride_patterns_in_weekdays) shows number of rides on Saturdays and Sundays by casual members overtake membership riders in the months of July and August.

#### Comparing variation in ride lengths of average and total ride lengths by bike type.

Aggregating data for the visualization.

rides\_on\_days <- data\_one\_year %>%  
 mutate(day = wday(started\_at, label = TRUE, abbr = FALSE),  
 month = month(started\_at, label = TRUE, abbr = FALSE)) %>%   
 summarise(ride\_count = n(),  
 sum\_ride\_length = sum(ride\_length, na.rm = TRUE),  
 avg\_ride\_length = mean(ride\_length, na.rm = TRUE),  
 .by = c(month, day, member\_casual))  
  
rides\_on\_days

# A tibble: 168 × 6  
 month day member\_casual ride\_count sum\_ride\_length avg\_ride\_length  
 <ord> <ord> <chr> <int> <dbl> <dbl>  
 1 July Wednesday member 47725 605175. 12.7  
 2 July Saturday casual 74543 2057158. 27.6  
 3 July Tuesday member 46360 588327. 12.7  
 4 July Tuesday casual 31415 705946. 22.5  
 5 July Saturday member 53796 817724. 15.2  
 6 July Friday casual 42333 960160 22.7  
 7 July Thursday casual 35800 759804. 21.2  
 8 July Sunday casual 61198 1715527. 28.0  
 9 July Thursday member 48572 623503. 12.8  
10 July Friday member 48221 616243. 12.8  
# ℹ 158 more rows

**Let’s visualize the aggregated data**

rides\_on\_days\_len <- rides\_on\_days %>%  
 ggplot(aes(day, sum\_ride\_length))+  
 geom\_col(aes(fill = member\_casual), position = "dodge")+  
 facet\_wrap(~month, ncol = 3)+  
 labs(x = "Day of the Week", y = "Total Length of Rides (Minutes)",  
 title = "Total Minutes driven by Riders",  
 fill = "Type of Rider") +  
 theme(legend.position = "top",  
 axis.text.x = element\_text(angle = 45, hjust = 1),  
 plot.title = element\_text(size = 18))  
  
rides\_on\_days\_len

|  |
| --- |
| Figure 3: Total Ride lengths through out the year by member types. |

rides\_on\_days\_len\_avg <- rides\_on\_days %>%  
 ggplot(aes(day, avg\_ride\_length))+  
 geom\_col(aes(fill = member\_casual), position = "dodge")+  
 facet\_wrap(~month, ncol = 3) +  
 labs(x = "Day of the Week", y = "Average Length of Rides (Minutes)",  
 title = "Average Minutes driven by Riders",  
 fill = "Type of Rider") +  
 theme(legend.position = "top",  
 axis.text.x = element\_text(angle = 45, hjust = 1),  
 plot.title = element\_text(size = 18))  
  
rides\_on\_days\_len\_avg

|  |
| --- |
| Figure 4: Average Ride lengths through out year by member types. |

The **ride length** is varying across months and seasons just as number of rides but **average ride length** is not fluctuating that much across the year.

### 1.2.3 Analysing of Stations and Routes.

* Removing “NA” and blanks from the stations columns.

data\_one\_year <- data\_one\_year %>%  
 drop\_na(start\_station\_name, end\_station\_name ) %>%   
 filter(start\_station\_name != "" & end\_station\_name != "",  
 started\_at != ended\_at)   
  
glimpse(data\_one\_year)

Rows: 4,409,072  
Columns: 14  
$ ride\_id <chr> "029D853B5C38426E", "C1D6D749139CB6C0", "D3E7C0B68E…  
$ rideable\_type <chr> "classic\_bike", "classic\_bike", "classic\_bike", "cl…  
$ started\_at <dttm> 2022-07-26 20:07:33, 2022-07-26 20:08:04, 2022-07-…  
$ ended\_at <dttm> 2022-07-26 19:59:34, 2022-07-26 19:59:34, 2022-07-…  
$ start\_station\_name <chr> "Lincoln Ave & Roscoe St\*", "Lincoln Ave & Roscoe S…  
$ start\_station\_id <chr> "chargingstx5", "chargingstx5", "chargingstx5", "ch…  
$ end\_station\_name <chr> "Lincoln Ave & Roscoe St\*", "Lincoln Ave & Roscoe S…  
$ end\_station\_id <chr> "chargingstx5", "chargingstx5", "chargingstx5", "ch…  
$ start\_lat <dbl> 41.94335, 41.94335, 41.94335, 41.94335, 41.93945, 4…  
$ start\_lng <dbl> -87.67067, -87.67067, -87.67067, -87.67067, -87.663…  
$ end\_lat <dbl> 41.94335, 41.94335, 41.94335, 41.94335, 41.93948, 4…  
$ end\_lng <dbl> -87.67067, -87.67067, -87.67067, -87.67067, -87.663…  
$ member\_casual <chr> "member", "member", "casual", "casual", "member", "…  
$ ride\_length <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

* Making a new column to identify travelled routes.

data\_one\_year <- data\_one\_year %>%   
 mutate(stations\_travelled = paste(start\_station\_name,   
 "-", end\_station\_name))  
  
glimpse(data\_one\_year)

Rows: 4,409,072  
Columns: 15  
$ ride\_id <chr> "029D853B5C38426E", "C1D6D749139CB6C0", "D3E7C0B68E…  
$ rideable\_type <chr> "classic\_bike", "classic\_bike", "classic\_bike", "cl…  
$ started\_at <dttm> 2022-07-26 20:07:33, 2022-07-26 20:08:04, 2022-07-…  
$ ended\_at <dttm> 2022-07-26 19:59:34, 2022-07-26 19:59:34, 2022-07-…  
$ start\_station\_name <chr> "Lincoln Ave & Roscoe St\*", "Lincoln Ave & Roscoe S…  
$ start\_station\_id <chr> "chargingstx5", "chargingstx5", "chargingstx5", "ch…  
$ end\_station\_name <chr> "Lincoln Ave & Roscoe St\*", "Lincoln Ave & Roscoe S…  
$ end\_station\_id <chr> "chargingstx5", "chargingstx5", "chargingstx5", "ch…  
$ start\_lat <dbl> 41.94335, 41.94335, 41.94335, 41.94335, 41.93945, 4…  
$ start\_lng <dbl> -87.67067, -87.67067, -87.67067, -87.67067, -87.663…  
$ end\_lat <dbl> 41.94335, 41.94335, 41.94335, 41.94335, 41.93948, 4…  
$ end\_lng <dbl> -87.67067, -87.67067, -87.67067, -87.67067, -87.663…  
$ member\_casual <chr> "member", "member", "casual", "casual", "member", "…  
$ ride\_length <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
$ stations\_travelled <chr> "Lincoln Ave & Roscoe St\* - Lincoln Ave & Roscoe St…

* Finding which route is most traveled by **casual riders**.

most\_travelled\_routes\_casual <- data\_one\_year %>%  
 filter(member\_casual == "casual",  
 ride\_length >= 2) %>%   
 summarise(ride\_count = n(),  
 avg\_ride\_length = round(mean(ride\_length), 2),  
 .by = c(stations\_travelled)) %>%  
 arrange(desc(ride\_count))  
  
head(most\_travelled\_routes\_casual)

# A tibble: 6 × 3  
 stations\_travelled ride\_count avg\_ride\_length  
 <chr> <int> <dbl>  
1 Streeter Dr & Grand Ave - Streeter Dr & Grand Ave 8259 46.3  
2 DuSable Lake Shore Dr & Monroe St - DuSable Lake S… 5726 38.3  
3 DuSable Lake Shore Dr & Monroe St - Streeter Dr & … 4840 27.1  
4 Michigan Ave & Oak St - Michigan Ave & Oak St 3754 50.9  
5 Millennium Park - Millennium Park 3188 45.4  
6 Streeter Dr & Grand Ave - DuSable Lake Shore Dr & … 2663 27.8

NROW(most\_travelled\_routes\_casual)

[1] 130371

**Streeter Dr & Grand Ave - Streeter Dr & Grand Ave** stands to be the most popular station with **9698 rides** by **casual** riders.

most\_travelled\_routes\_member <- data\_one\_year %>%  
 filter(member\_casual == "member") %>%   
 summarise(ride\_count = n(),  
 total\_ride\_length = sum(ride\_length),  
 ride\_length = round(mean(ride\_length), 2),  
 .by = stations\_travelled) %>% arrange(desc(ride\_count))  
  
head(most\_travelled\_routes\_member)

# A tibble: 6 × 4  
 stations\_travelled ride\_count total\_ride\_length ride\_length  
 <chr> <int> <dbl> <dbl>  
1 Ellis Ave & 60th St - University Ave… 6153 25936. 4.22  
2 University Ave & 57th St - Ellis Ave… 5786 26634. 4.6   
3 Ellis Ave & 60th St - Ellis Ave & 55… 5676 28427. 5.01  
4 Ellis Ave & 55th St - Ellis Ave & 60… 5347 27187. 5.08  
5 State St & 33rd St - Calumet Ave & 3… 4156 18014. 4.33  
6 Calumet Ave & 33rd St - State St & 3… 4027 15887. 3.95

NROW(most\_travelled\_routes\_member)

[1] 145104

**Ellis Ave & 60th St - University Ave & 57th St** stands as the most traveled route by **member** riders with **6153** rides per anum.

* Finding which station has most ride starting points and which station has most ending points.

most\_starting\_points <- data\_one\_year %>%   
 summarise(ride\_count = n(),  
 .by = start\_station\_name) %>%  
 select(start\_station\_name, ride\_count) %>%  
 slice\_max(ride\_count, n = 10)  
  
most\_starting\_points

# A tibble: 10 × 2  
 start\_station\_name ride\_count  
 <chr> <int>  
 1 Streeter Dr & Grand Ave 65892  
 2 DuSable Lake Shore Dr & Monroe St 37939  
 3 Michigan Ave & Oak St 36036  
 4 DuSable Lake Shore Dr & North Blvd 35091  
 5 Wells St & Concord Ln 33250  
 6 Clark St & Elm St 32751  
 7 Kingsbury St & Kinzie St 31876  
 8 Millennium Park 30917  
 9 Theater on the Lake 29600  
10 Wells St & Elm St 28063

most\_starting\_points$ride\_count %>% sum()

[1] 361415

most\_ending\_points <- data\_one\_year %>%   
 summarise(ride\_count = n(),  
 .by = end\_station\_name) %>%  
 select(end\_station\_name, ride\_count) %>%   
 slice\_max(ride\_count, n = 10)  
  
most\_ending\_points

# A tibble: 10 × 2  
 end\_station\_name ride\_count  
 <chr> <int>  
 1 Streeter Dr & Grand Ave 67536  
 2 DuSable Lake Shore Dr & North Blvd 38026  
 3 Michigan Ave & Oak St 36976  
 4 DuSable Lake Shore Dr & Monroe St 36806  
 5 Wells St & Concord Ln 33814  
 6 Clark St & Elm St 32325  
 7 Millennium Park 32046  
 8 Kingsbury St & Kinzie St 31058  
 9 Theater on the Lake 30214  
10 Wells St & Elm St 28212

most\_ending\_points$ride\_count %>% sum()

[1] 367013

**Streeter Dr & Grand Ave** found to be the most popular station as most rides start and end at that station.

### 1.2.4 Looking at Filtered data

Just because we filtered the data with NA’s that does not mean that the data is not helpful, it just means that it does not our fulfill specific need when calculating or manipulating data.

Let’s look at NA’s in the data once again.

na\_in\_cols <- data\_one\_year\_raw %>% map( ~sum(is.na(.))) %>% unlist()  
  
na\_in\_cols

ride\_id rideable\_type started\_at ended\_at   
 0 0 0 0   
start\_station\_name start\_station\_id end\_station\_name end\_station\_id   
 857860 857992 915655 915796   
 start\_lat start\_lng end\_lat end\_lng   
 0 0 5795 5795   
 member\_casual   
 0

* We can see that the start\_station\_name and end\_station\_name have majority of NA’s it means that rides are starting and ending where stations are not there.

prop\_na <- na\_in\_cols["start\_station\_name"]/nrow(data\_one\_year\_raw)  
  
prop\_na

start\_station\_name   
 0.148433

* 14.8432963% of data in start\_station\_name is missing and good thing is that none of the start\_lng and start\_lat have any NA’s and we can use this for find the most traveled routes.

data\_na\_one\_year <- data\_one\_year\_raw %>%   
 filter(is.na(start\_station\_name) | start\_station\_name == "") %>%   
 drop\_na(end\_lat, end\_lng)  
   
glimpse(data\_na\_one\_year)

Rows: 857,860  
Columns: 13  
$ ride\_id <chr> "DCB3D2C9B63999EC", "D1ACA8280DA02AE3", "EF98673429…  
$ rideable\_type <chr> "electric\_bike", "electric\_bike", "electric\_bike", …  
$ started\_at <dttm> 2022-07-04 15:04:26, 2022-07-12 14:43:51, 2022-07-…  
$ ended\_at <dttm> 2022-07-04 15:32:38, 2022-07-12 14:49:28, 2022-07-…  
$ start\_station\_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…  
$ start\_station\_id <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…  
$ end\_station\_name <chr> "Ashland Ave & Blackhawk St", "Cornell Ave & Hyde P…  
$ end\_station\_id <chr> "13224", "KA1503000007", "KA1503000007", "847", "48…  
$ start\_lat <dbl> 41.95, 41.80, 41.80, 41.74, 42.02, 41.95, 41.95, 41…  
$ start\_lng <dbl> -87.64, -87.59, -87.59, -87.55, -87.69, -87.67, -87…  
$ end\_lat <dbl> 41.90707, 41.80241, 41.80241, 41.73000, 42.01000, 4…  
$ end\_lng <dbl> -87.66725, -87.58692, -87.58692, -87.55000, -87.690…  
$ member\_casual <chr> "member", "member", "member", "member", "member", "…

* Now let’s make new columns start\_point with start\_lng and start\_lat and end\_point with end\_lat and end\_lng.

data\_na\_one\_year <- data\_na\_one\_year %>%  
 mutate(start\_point = paste(start\_lat, start\_lng),  
 end\_point = paste(end\_lat, end\_lng))  
  
glimpse(data\_na\_one\_year)

Rows: 857,860  
Columns: 15  
$ ride\_id <chr> "DCB3D2C9B63999EC", "D1ACA8280DA02AE3", "EF98673429…  
$ rideable\_type <chr> "electric\_bike", "electric\_bike", "electric\_bike", …  
$ started\_at <dttm> 2022-07-04 15:04:26, 2022-07-12 14:43:51, 2022-07-…  
$ ended\_at <dttm> 2022-07-04 15:32:38, 2022-07-12 14:49:28, 2022-07-…  
$ start\_station\_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…  
$ start\_station\_id <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…  
$ end\_station\_name <chr> "Ashland Ave & Blackhawk St", "Cornell Ave & Hyde P…  
$ end\_station\_id <chr> "13224", "KA1503000007", "KA1503000007", "847", "48…  
$ start\_lat <dbl> 41.95, 41.80, 41.80, 41.74, 42.02, 41.95, 41.95, 41…  
$ start\_lng <dbl> -87.64, -87.59, -87.59, -87.55, -87.69, -87.67, -87…  
$ end\_lat <dbl> 41.90707, 41.80241, 41.80241, 41.73000, 42.01000, 4…  
$ end\_lng <dbl> -87.66725, -87.58692, -87.58692, -87.55000, -87.690…  
$ member\_casual <chr> "member", "member", "member", "member", "member", "…  
$ start\_point <chr> "41.95 -87.64", "41.8 -87.59", "41.8 -87.59", "41.7…  
$ end\_point <chr> "41.907066 -87.667252", "41.802406 -87.586924", "41…

* Aggregating data to check for the most traveled routes without a start\_station name.

First join start\_point and end\_point to make route\_travelled then count the rides by routes\_travelled to see the most traveled path.

most\_travelled\_na\_routes <- data\_na\_one\_year %>%  
 filter(start\_point != end\_point) %>%   
 mutate(route\_travelled = paste(start\_point, ",", end\_point)) %>%   
 summarise(ride\_count = n(),  
 .by = route\_travelled) %>%  
 slice\_max(ride\_count, n=10)  
  
most\_travelled\_na\_routes

# A tibble: 10 × 2  
 route\_travelled ride\_count  
 <chr> <int>  
 1 41.79 -87.6 , 41.8 -87.59 1459  
 2 41.79 -87.59 , 41.79 -87.6 1354  
 3 41.8 -87.59 , 41.79 -87.6 1335  
 4 41.79 -87.6 , 41.79 -87.59 1320  
 5 41.8 -87.6 , 41.79 -87.6 1099  
 6 41.79 -87.6 , 41.78509714636 -87.6010727606 1058  
 7 41.79 -87.6 , 41.8 -87.6 999  
 8 41.79 -87.6 , 41.799568 -87.594747 917  
 9 41.79 -87.6 , 41.78 -87.6 697  
10 41.89 -87.63 , 41.9 -87.63 690

sum(most\_travelled\_na\_routes["ride\_count"])

[1] 10928

* **10928** rides are not small when compared to most traveled routes, but 10928 rides in 5 million rides is not that high.

### 1.2.5 Conclusions

* As casual members go for long rides on the weekends offers on weekend rides with membership buying may help attract more memberships and might also make membership riders to make weekend end trips.
* To increase the memberships of the **Cyclist Bike Share** the company needs to place stations where most unknown routes are traveled by the riders.