Google Data Analytics Capstone Project. Case Study 1: How Does a Bike-Share Navigate Speedy Success?

Paul W

January 29, 2024

Company Background

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime. Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.

Business Task

Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, the marketing director believes that maximizing the number of annual members will be key to future growth. The marketing director has a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the marketing team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. The director and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

The purpose of this case study and report is to do an exploratory data analysis of the provided data and identify any data points, descriptive statistics or trends that may help the marketing team achieve their goals.

The analysis is broken up into these sections that are steps of the data analysis process from the course:

- Ask
- Prepare
- Process
- Analyze
- Share
- Act

Ask - Three Questions

Three questions will guide the future marketing program:

- 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

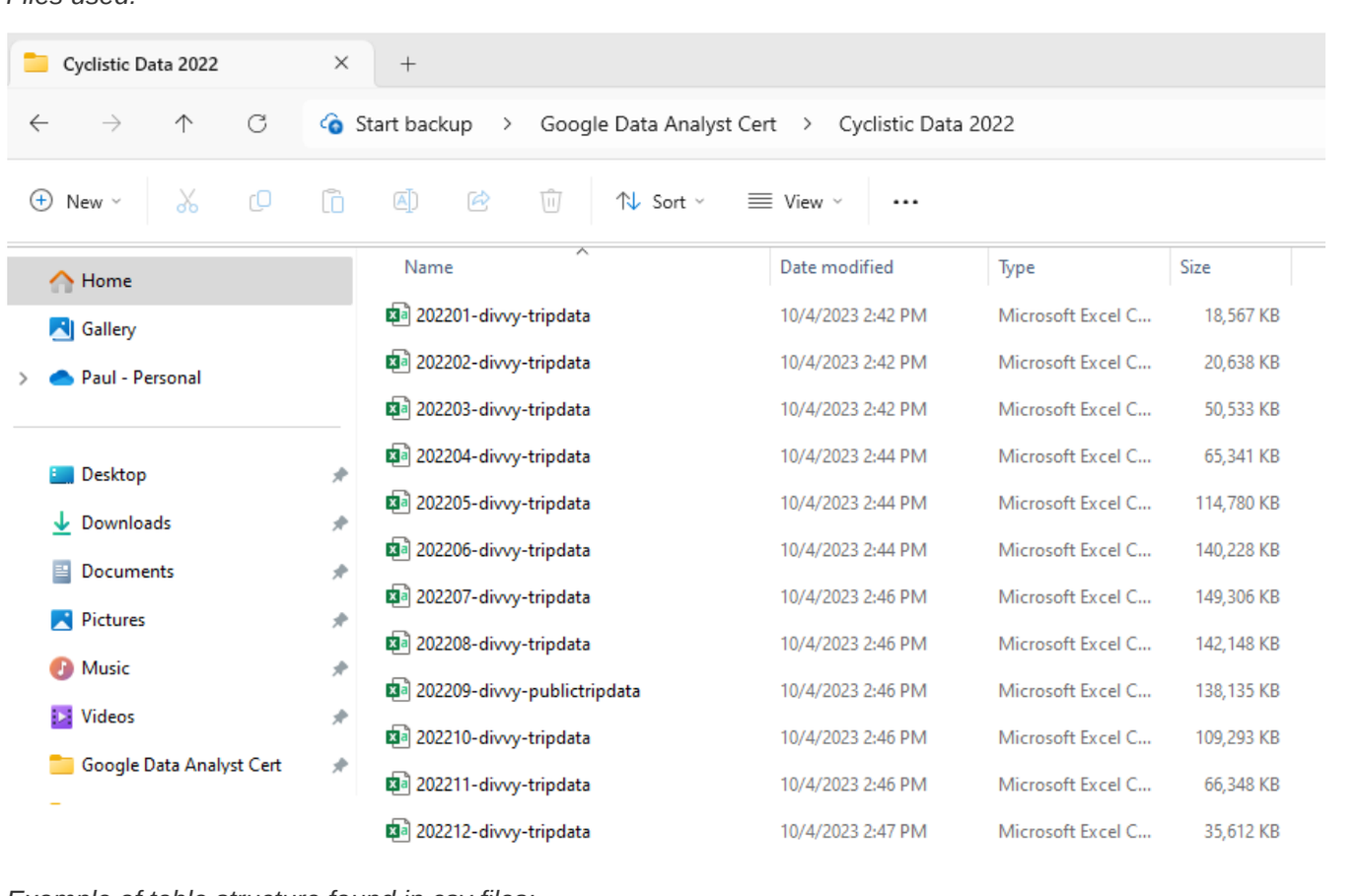
Prepare - Preparation of Data

Description of all data sources used and technical information

The data for this project was provided by Coursera in the form of many csv files. Each csv file has bike ride data with a unique id to represent each bike trip made and each file has data for a particular month and year. These files can be made to create one large fact table of bike ride data. No files were offered to create dimension tables. Also sql was not needed to pull the files - they were made already available in several folders. Instead of using sql to prepare the files, R was used to prepare the data and perform an analysis. R Markdown was used to create the report document and further analysis and presentation was done with Tableau Public. Only one year's worth of data could be used for this report because of the limits to how much data can be uploaded to Tableau Public. The documentation from the project also notes that we will not be able to identify individual users and purchases because there are no credit card numbers or identification numbers given to identify them. There were no data dictionaries or layouts provided, further complicating the analysis performed.

Here are some screenshots of the csv files and a sneak peak view of how the tables with their columns appear:

Files used:



Example of table structure found in csv files:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual
1	C2F7DD78E82EC875	electric_bike	2022-01-13 11:59:47	2022-01-13 12:02:44	Glenwood Ave & Touhy Ave	525	Clark St & Touhy Ave	RP-007	42.01280	-87.66591	42.01256	-87.67437	casual
2	A6CF8980A652D272	electric_bike	2022-01-10 08:41:56	2022-01-10 08:46:17	Glenwood Ave & Touhy Ave	525	Clark St & Touhy Ave	RP-007	42.01276	-87.66597	42.01256	-87.67437	casual
3	BD0F91DFF741C66D	classic_bike	2022-01-25 04:53:40	2022-01-25 04:58:01	Sheffield Ave & Fullerton Ave	TA1306000016	Greenview Ave & Fullerton Ave	TA1307000001	41.92560	-87.65371	41.92533	-87.66580	member
4	CBB80ED419105406	classic_bike	2022-01-04 00:18:04	2022-01-04 00:33:00	Clark St & Bryn Mawr Ave	KA1504000151	Paulina St & Montrose Ave	TA1309000021	41.98359	-87.66915	41.96151	-87.67139	casual
5	DDC963BFDDA51EEA	classic_bike	2022-01-20 01:31:10	2022-01-20 01:37:12	Michigan Ave & Jackson Blvd	TA1309000002	State St & Randolph St	TA1305000029	41.87785	-87.62408	41.88462	-87.62783	member

Install and Load Packages

```
# Load libraries
library("tidyverse")
library("leaflet")
library("dplyr")
library("magrittr")
library("readr")
library("ggplot2")
library("stringr")
library("geosphere")
library("sf")
library("maps")
library("lubridate")
library("scales")
library("viridis")
library("gridExtra")
```

Combine csv files into one dataframe. Check csv files for issues that would prevent combining them

Instead of inspecting each csv file individually, this method combines all of the csv files for 2022 into one dataframe. It also checks for anything that can produce errors such as differences found in column names, column types, missing or extra columns etc. The console will give a report for each file showing if there were errors or if the file successfully passed inspection. No errors were found and files were combined successfully:

```
Data_Combined_2022 <- list.files(path = "C:/Users/paulw/Desktop/Google Data Analyst Cert/Cyclistic Data 2022",
                                 pattern="*.csv",
                                 full.names = T) %>%
                                 map_df(~read_csv(.))

#Data_Combined_2022 <- Data_Combined_2022 %>% slice(1:100000)
```

Process - Data Processing and Data Validation

Initial Review of Tables and Any Documents

For many projects, a data dictionary/layout is normally reviewed by the analyst. This project did not include one which would be very helpful when inspecting the fields and checking for errors. Also, it is good to know exactly what is being reported in a field and to know what lengths, data types etc. are to be expected. We will have to look at the fields and try to get an idea of what data types should be in those columns and if they contain facets, see if the correct facet values are there. Here we can start with str() to get a summary of what is going on in the dataframe and get names of columns, the data types found in the columns and a few sample values of what is found in each column:

```
str(Data_Combined_2022)
```

```
## spc_tbl_ [5,667,717 × 13] (S3: spc_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5667717] "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED419105
406" ...
##  $ rideable_type     : chr [1:5667717] "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:5667717], format: "2022-01-13 11:59:47" "2022-01-10 08:41:56" ...
##  $ ended_at          : POSIXct[1:5667717], format: "2022-01-13 12:02:44" "2022-01-10 08:46:17" ...
##  $ start_station_name: chr [1:5667717] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave &
Fullerton Ave" "Clark St & Bryn Mawr Ave" ...
##  $ start_station_id  : chr [1:5667717] "525" "525" "TA1306000016" "KA1504000151" ...
##  $ end_station_name  : chr [1:5667717] "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerto
n Ave" "Paulina St & Montrose Ave" ...
##  $ end_station_id    : chr [1:5667717] "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...
##  $ start_lat         : num [1:5667717] 42 42 41.9 42 41.9 ...
##  $ start_lng         : num [1:5667717] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:5667717] 42 42 41.9 42 41.9 ...
##  $ end_lng           : num [1:5667717] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:5667717] "casual" "casual" "member" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   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>
```

Initial Cleaning of Columns with Strings

Clean leading and trailing spaces that may be before or after text in columns that have strings

```
Columns_with_strings = c("ride_id", "rideable_type", "start_station_name", "start_station_id", "end_station_nam
e", "end_station_id", "member_casual")

Data_Combined_2022[Columns_with_strings] <- lapply(Data_Combined_2022[Columns_with_strings],trimws)
```

Identify the columns in the dataframe that have a NA (null) value. Are these expected to occasionally have an NA value?

Total Rows in Data_Combined_2022 = 5,667,717 Total Rows where a column has NA = 1,298,357. **Roughly 23% of the rows for 2022 have an NA in at least one column. Normally would get with someone to figure out why almost a quarter of the data is unusable….**

```
AnyColsNA <- colnames(Data_Combined_2022[colSums(is.na(Data_Combined_2022))>1])

AnyRowsNA <- Data_Combined_2022[rowSums(is.na(Data_Combined_2022))>1,]  #Return data frame with rows containing N
A value.

Total_Rows_2022 <- nrow(Data_Combined_2022)   #Total Rows in the data frame so far
Total_Rows_wNA_2022 <- nrow(AnyRowsNA)        #Total Rows with one or more fields with NA



Cleaned_Data_2022 <- Data_Combined_2022[rowSums(is.na(Data_Combined_2022))==0,]   #Dropping rows from the datafra
me that have NA values and renaming the dataframe
```

Add new columns

Add new columns: ride_time_minutes, started_at_hour, start_day_of_week, start_month, ride_distance_feet

```
# Add a column ride_time_minutes that is the difference in start and end times, convert to minutes
Cleaned_Data_2022 <- Cleaned_Data_2022 %>% mutate(ride_time_minutes = difftime(ended_at,started_at,units="mins"))
Cleaned_Data_2022$ride_time_minutes <- as.numeric(gsub(" mins", "", Cleaned_Data_2022$ride_time_minutes)) #reform
at as number

#Add a column that has the hour the ride started
Cleaned_Data_2022$started_at_hour <- hour(Cleaned_Data_2022$started_at)

Cleaned_Data_2022$hour_am_pm <- ifelse(
  Cleaned_Data_2022$started_at_hour == 0, "12 AM",
  ifelse(
    Cleaned_Data_2022$started_at_hour == 12, "12 PM",
    ifelse(
      Cleaned_Data_2022$started_at_hour < 12,
      paste0(Cleaned_Data_2022$started_at_hour, " AM"),
      paste0(Cleaned_Data_2022$started_at_hour - 12, " PM")
    )
  )
)
# Add a column for the week day the ride started on and a column for the month it was in
Cleaned_Data_2022$start_day_of_week <- weekdays(Cleaned_Data_2022$started_at)
Cleaned_Data_2022$start_month <- month(Cleaned_Data_2022$started_at, label = TRUE, abbr = FALSE)

# Add a ride_distance_feet column that is the distance between starting and ending stations.
Cleaned_Data_2022 <- Cleaned_Data_2022 %>% mutate(ride_distance_feet = distGeo(matrix(c(start_lng, start_lat),
ncol=2),matrix(c(end_lng, end_lat), ncol=2))*3.2808399)
```

Data Validation and More Cleaning

This is where I have my own steps inserted into the process step used by this course. This is where the Process step overlaps with the Analyze step. Normally, the data is validated before further processing. The data is checked against a data dictionary or any notes we have and some testing occurs. If errors are found, then the data may have to be sent back or you may have to go and pull a new query. While this is going on , the rest of the project can be continued while waiting for feedback on the data quality and any questions about the data issues to be answered.

Checking columns that contain unique identifiers - does the column have unique values?

The ride_id column is used to identify individual tirps. Checking if all values are unique by comparing the row count to the distinct count of ride_id's. Both the row count for the data frame and the distinct number of ride id's match

```
Distinct_ride_ids <- n_distinct(Data_Combined_2022$ride_id)
Distinct_ride_ids
```

```
## [1] 5667717
```

```
Total_Rows_2022 <- nrow(Data_Combined_2022)
Total_Rows_2022
```

```
## [1] 5667717
```

Columns with facet data - do these have any unexpected values?

An example of a facet column would be a column like member_casual that has repeated values that stratify the data into categories. Even though we do not have a data dictionary it appears that this column should have only two values - "member" or "casual". The rideable_types column should also have values "electric_bike", "classic_bike" and "docked bike". The station names and ids should be checked as well but that will be mentioned in the next subsection:

Values found in member_casual column are correct:

```
check_member_casual <- unique(Cleaned_Data_2022$member_casual)
check_member_casual
```

```
## [1] "casual" "member"
```

Values found in rideable-types are correct:

```
check_rideable_type <- unique(Cleaned_Data_2022$rideable_type)
check_rideable_type
```

```
## [1] "electric_bike" "classic_bike"  "docked_bike"
```

Columns with facet data continued - why are there several station names found per station id?

Several station id's were found where they had several station names attached to them. Should there not be one station id per station name? Again there is no data dictionary or layout to refer to and there is not any more information provided in the project handout.

An example of multiple station names listed with a station id:

start_station_id	end_station_name
13109	Broadway & Thorndale Ave
13109	Broadway & Thorndale Ave
13109	Broadway & Thorndale Ave
13109	Broadway & Thorndale Ave
13109	Clark St & Winnemac Ave
13109	Clark St & Winnemac Ave
13109	Clark St & Winnemac Ave
13109	Clark St & Winnemac Ave

Find any extreme latitude and longitude data points that do not belong in the data set

The latitude and longitude points in the data should all be located in the Chicago area. Any extreme points that do not belong in the Chicago area may affect the map negatively when trying to load or display it. They may also throw off any distance and travel time calculations.

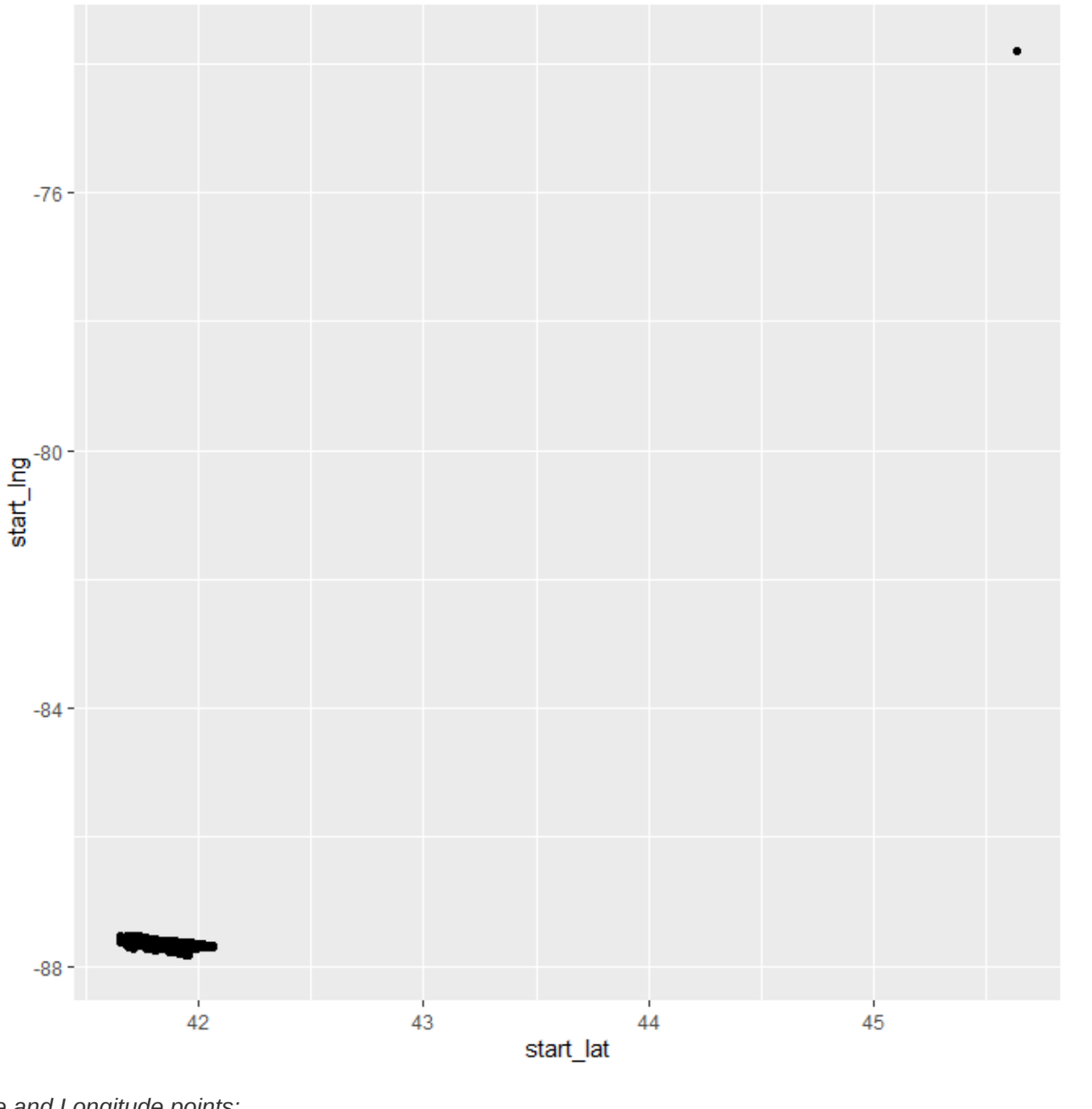
Starting and Ending geolocations should be clustered together when using a scatter plot that has longitutde on the y axis and latitude on the x axis. Extraneous points will appear away from the cluster of Chicago points. In the images below, the clustered points belonging to Chicago are in the -75 to -84 range for longitude points and latitude points are in the 40 to 43 range. The data that does not belong to Chicago is away from these clusters and outside of these ranges:

Starting Station Latitude and Longitude points:

```
#This next line can be used to generate the plot but will have to be made separately and then inserted into the d
ocument as an image. R Studio freezes when trying to execute this while knitting the document

#(checking_start_plot <- ggplot(Cleaned_Data_2022,aes(x = start_lat, y = start_lng)) + geom_point())

knitr::include_graphics("C:/Users/paulw/Desktop/Google Data Analyst Cert/R Documents/Not_In_Chicago_1.png")
```

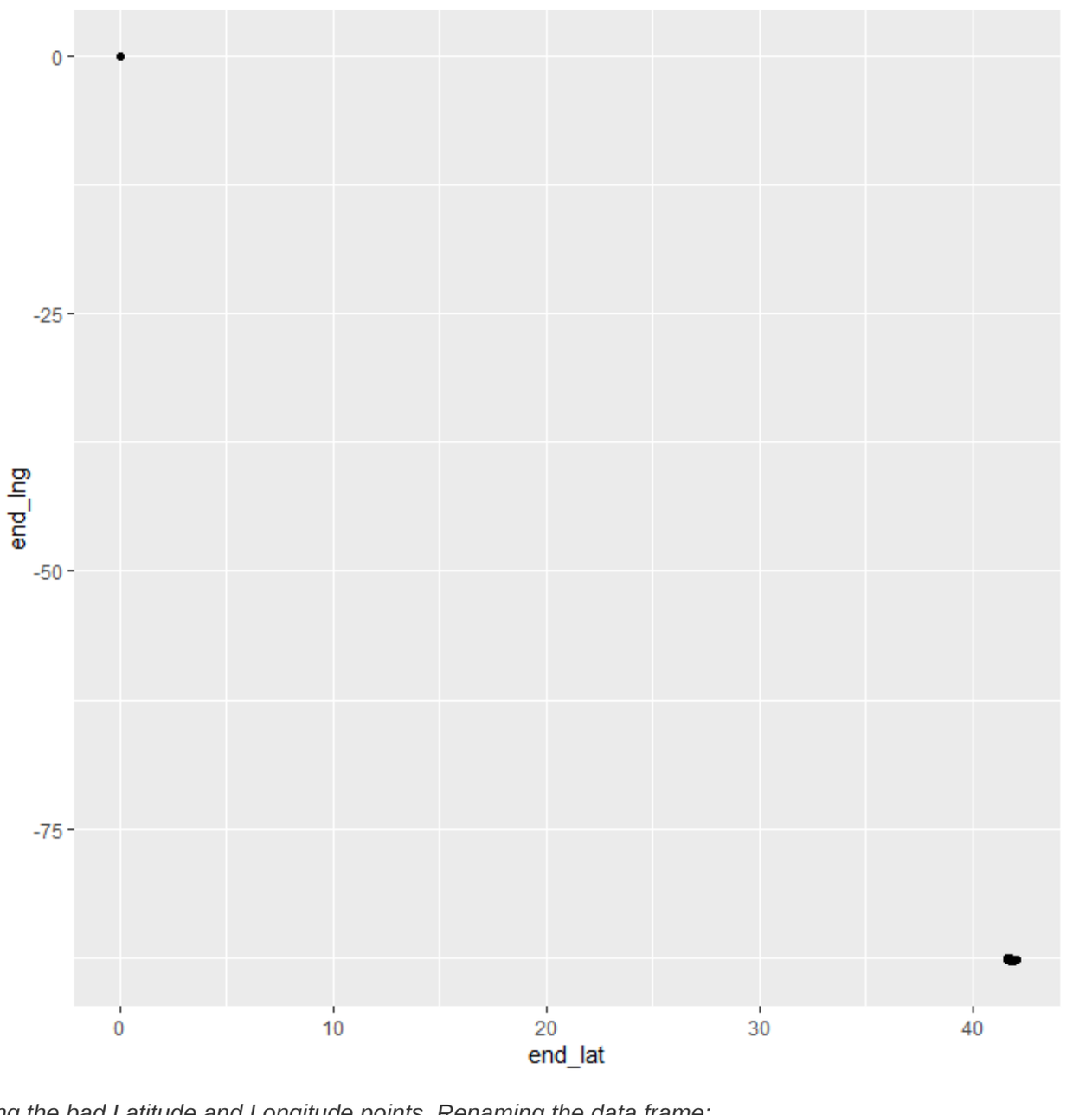


Ending Station Latitude and Longitude points:

```
#This next line can be used to generate the plot but will have to be made separately and then inserted as an imag
e into the document. R Studio freezes when trying to execute this while knitting the document

#(checking_end_plot <- ggplot(Cleaned_Data_2022,aes(x = end_lat, y = end_lng)) + geom_point())

knitr::include_graphics("C:/Users/paulw/Desktop/Google Data Analyst Cert/R Documents/Not_In_Chicago_2.png")
```



Identiyfing and removing the bad Latitude and Longitude points. Renaming the data frame:

```
#Going online and looking up each of these points in this dataframe will show they are not in Chicago
Not_Chicago_Locations <- subset(Cleaned_Data_2022, start_lat < 40 | start_lng > -75 | end_lat < 40 | end_lng > -7
5)


#Removing the rows that have ride data outside the Chicago area
Cleaned_Data_2022 <- subset(Cleaned_Data_2022, (start_lat < 43))
Cleaned_Data_2022 <- subset(Cleaned_Data_2022, (end_lng != 0))
```

Find and remove rows with started_at time greater than ended_at time

The columns started_at and ended_at have timestamps for each bike trip. Are there rows where the start time is greater than the end time? Yes there are 69 rows where this happens and these rows will give a negative value when calculating the ride time. These also have to be removed from the data

```
AnyRows_Invalid_Travel_Time<- subset(Cleaned_Data_2022, started_at > ended_at)
Cleaned_Data_2022 <- subset(Cleaned_Data_2022,started_at <= ended_at)#& started_at = ended_at)
```

Remove any rows where the started_at and ended_at time is incomplete.

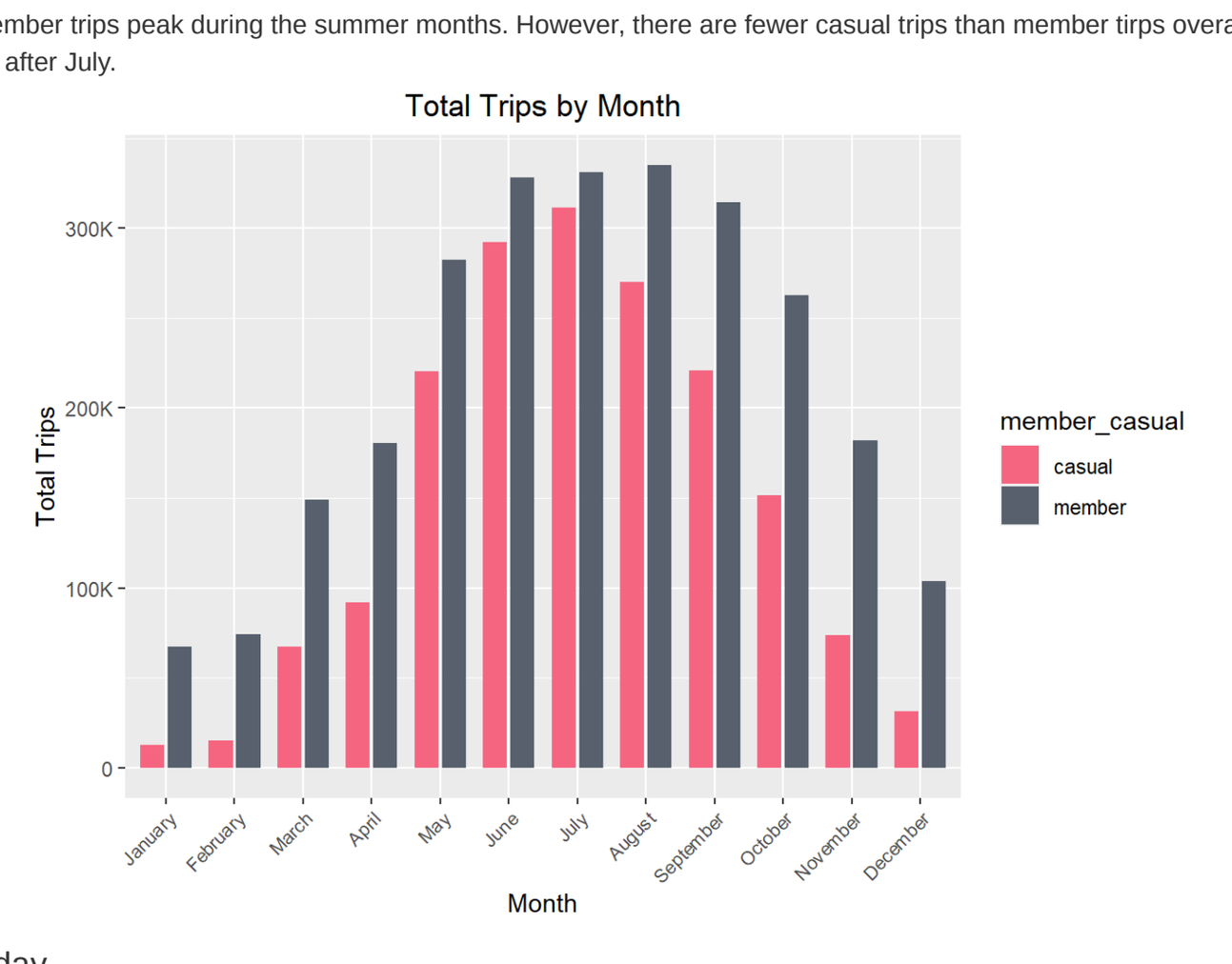
There are rows where there is a date in the column but not a time stamp.These are being removed also. There is a total of 41 rows where this occurs. There is no data dictionary provided by the project but after careful examination, the date time fields should have values with a length of 19. Filtering out lengths less than this will remove incomplete start and times.

```
Incomplete_Start_Time <- Cleaned_Data_2022[nchar(as.character(Cleaned_Data_2022$started_at)) < 19, ]
Cleaned_Data_2022 <- subset(Cleaned_Data_2022, nchar(as.character(started_at)) == 19)
Incomplete_End_Time <-Cleaned_Data_2022[nchar(as.character(Cleaned_Data_2022$ended_at)) < 19, ]
Cleaned_Data_2022 <- subset(Cleaned_Data_2022, nchar(as.character(ended_at)) == 19)
```

Analyze

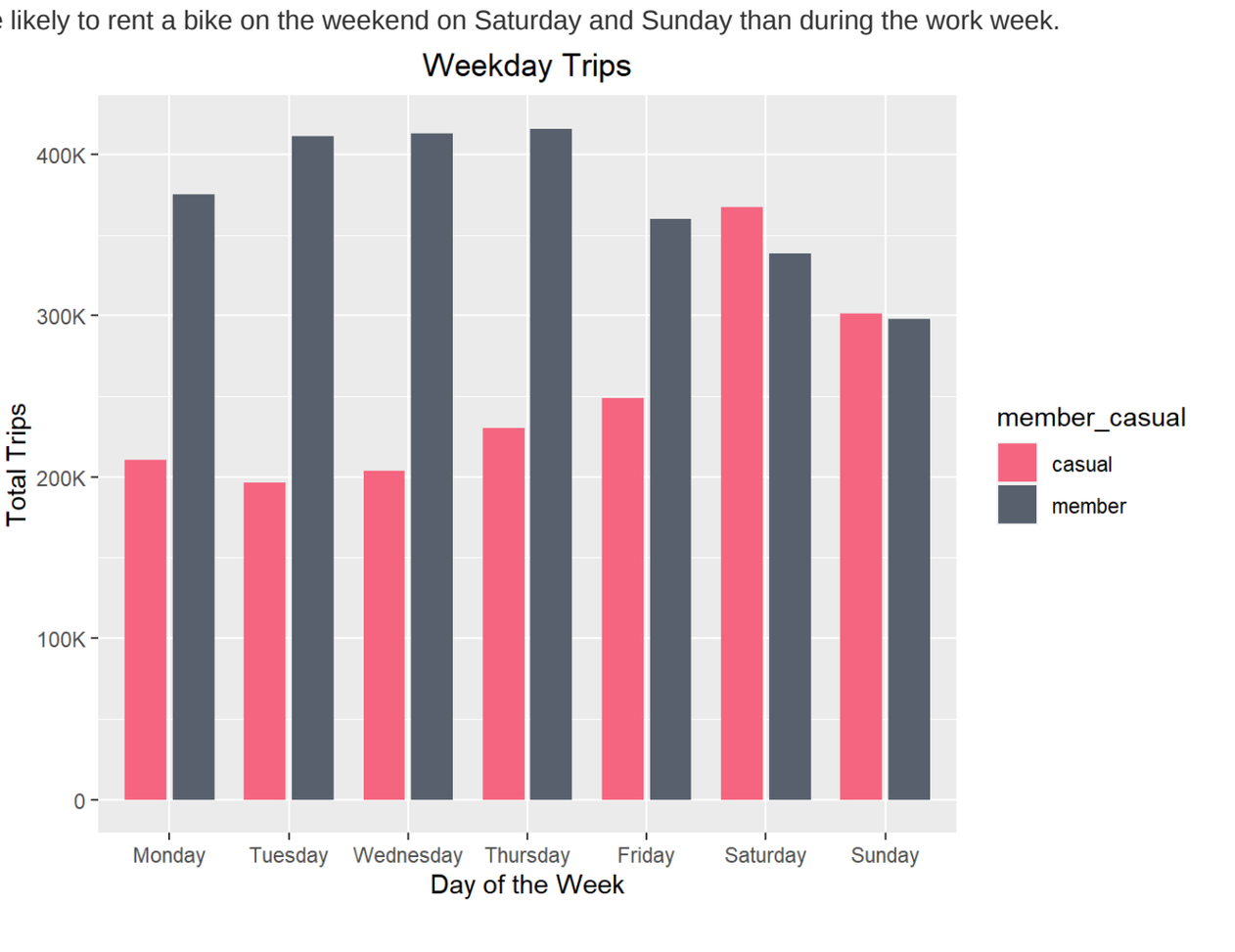
Trips Per Month

Both Casual and Member trips peak during the summer months. However, there are fewer casual trips than member tirps overall and the casual trips decline sharply after July.



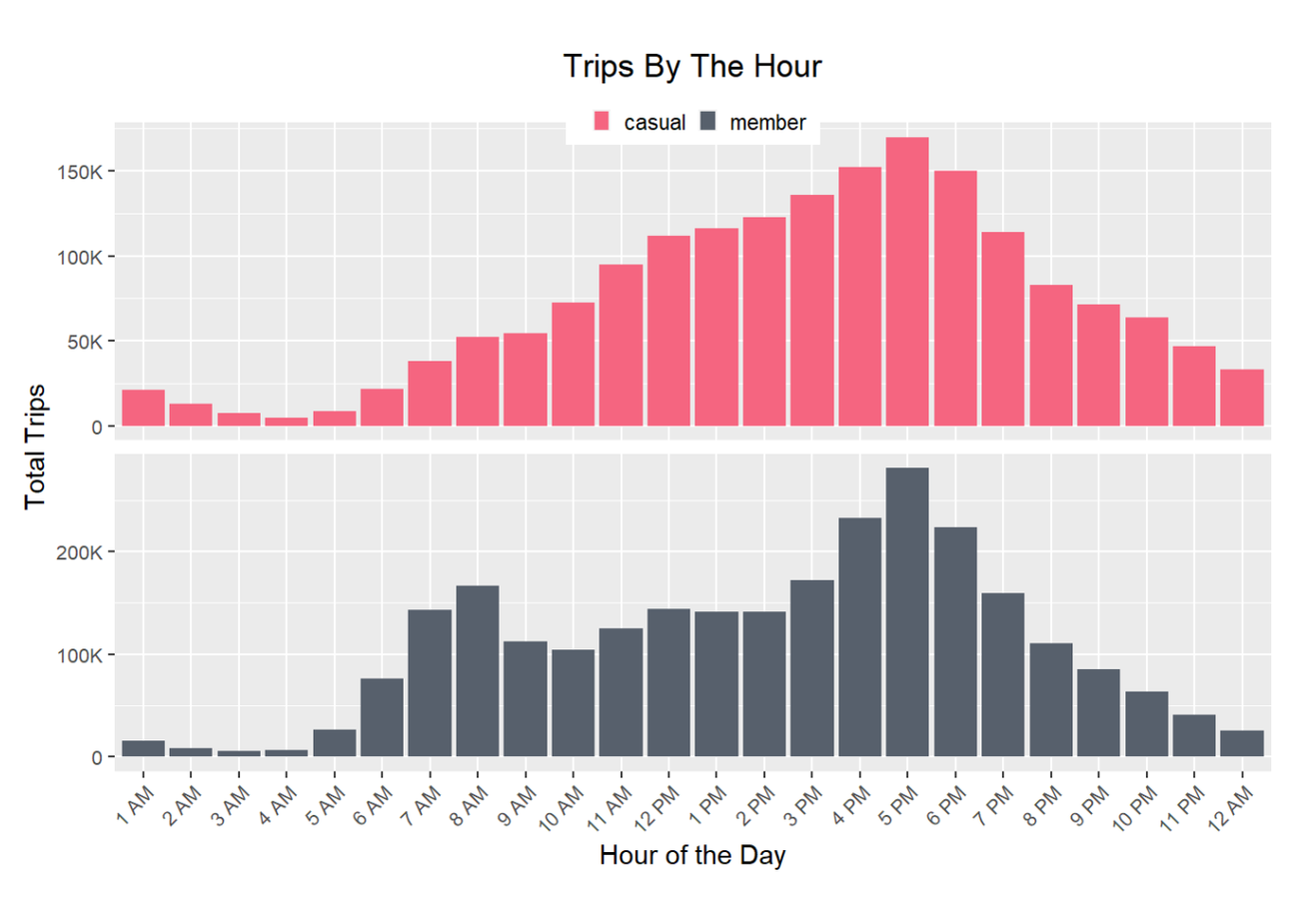
Trips per Weekday

Casual riders are more likely to rent a bike on the weekend on Saturday and Sunday than during the work week.



Trips by the Hour

Member trips peak both in the morning and later in the evening at 5pm. Casual trips peak at 5 pm but do not peak in the morning. Casual trips occur more between 12pm and 5pm.



Time Spent Summary.

Casual riders spend more time on averge and in total riding their bikes than Member riders.

```
##                        casual     member
## Total Minutes Spent  42,182,990 32,512,947
## Max. Minutes Spent   34,354.067  1,493.233
## Avg. Minutes Spent     23.99316   12.45175
## Median Minutes Spent  13.850000   8.983333
## Min. Minutes Spent            0          0
```

Distance Traveled Summary

Casual riders spend more time on average and in total on their bikes than Members, but Members' travel distances are slightly more than what Casual riders travel.

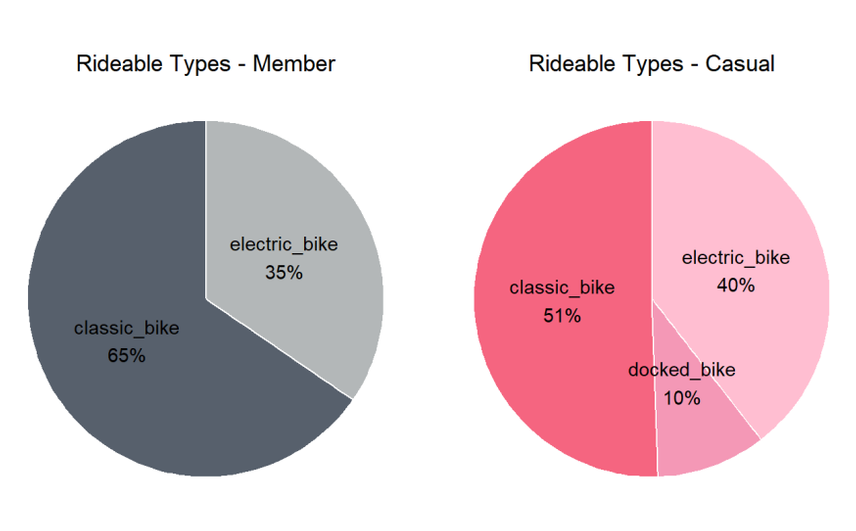
```
##                                casual         member
## Total Distance Traveled   12,397,480,273 17,532,883,606
## Max. Ride Distance Feet        98,941.86      99,375.86
## Avg. Ride Distance Feet        7,051.531      6,714.711
## Median Ride Distance Feet      5,415.157      4,918.963
## Min. Ride Distance Feet                0              0
```

Before creating visuals such as top ten stations by total rides for members and casual riders, will need to split the data into two subsets

```
member_data <- subset(Cleaned_Data_2022, member_casual == "member")
casual_data <- subset(Cleaned_Data_2022, member_casual == "casual")
```

Rideable Types

Members do not show having any trips for docked bikes (again this could be due to the removal of incomplete rows in the data). We do not have any data on bike availability to see whether or not there is a preference for either type of bike. Not enough information has been given to determine whether there is a preference for a type of bike. Also there is this question: can a docked bike also be electric? or a classic bike? The information that came with the case study does not go answer this. More information will be needed from the marketing team.

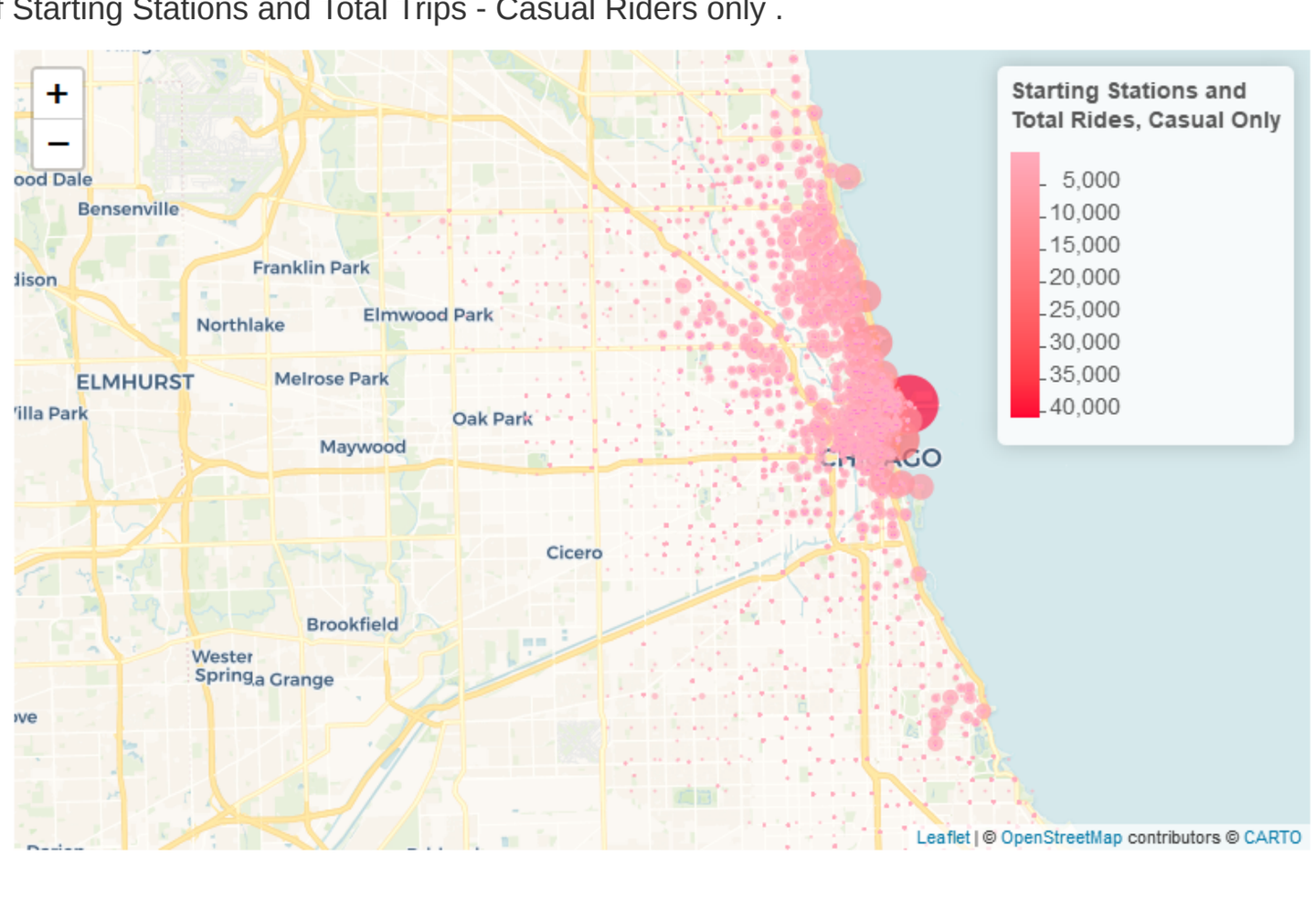


Top Ten Starting Station Names by Total Trip Count - Casual Only

Casual rider tips are more concentrated deep within Chicago's downtown area, along Lake Michigan's shore. The bike station at Streeter Drive and Grand Ave is the most popular and frequented station used by Casual riders. The further away from this bike station, fewer Casual trips are logged. (See the map plot in the next image that goes with this top ten summary)

```
##                   start_station_name total_trips
## 1             Streeter Dr & Grand Ave      55,060
## 2   DuSable Lake Shore Dr & Monroe St      30,262
## 3                    Millennium Park      23,950
## 4              Michigan Ave & Oak St      23,761
## 5  DuSable Lake Shore Dr & North Blvd      22,157
## 6                     Shedd Aquarium      19,421
## 7                Theater on the Lake      17,333
## 8              Wells St & Concord Ln      14,834
## 9                     Dusable Harbor      13,270
## 10           Clark St & Armitage Ave      12,779
```

Map plot of Starting Stations and Total Trips - Casual Riders only .

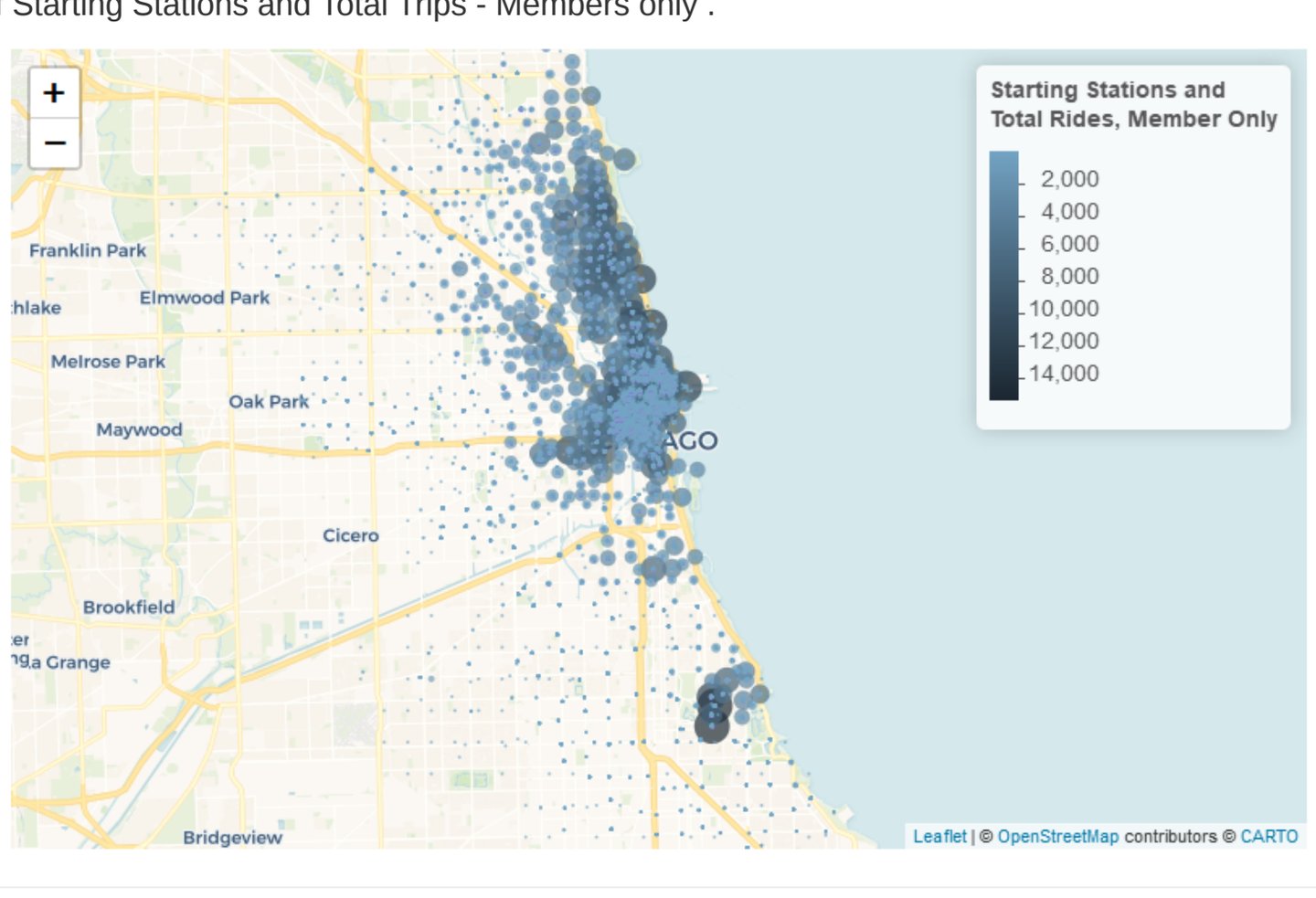


Top Ten Starting Station Names by Total Trip Count - Members Only

Member trips are also more frequently close to Lake Michigan and downtown Chicago. However they are more spread out than the Member trips. The Member trips also appear to be more evenly spread out among the popular bike stations that are near the lake shore and downtown area. (See the map plot in the next image that goes with this top ten summary).

```
##             start_station_name total_trips
## 1      Kingsbury St & Kinzie St      23,523
## 2             Clark St & Elm St      20,581
## 3         Wells St & Concord Ln      19,674
## 4  Clinton St & Washington Blvd      18,828
## 5      Loomis St & Lexington St      18,251
## 6       Clinton St & Madison St      18,007
## 7      University Ave & 57th St      17,581
## 8           Ellis Ave & 60th St      17,504
## 9             Wells St & Elm St      17,496
## 10      Streeter Dr & Grand Ave      16,208
```

Map plot of Starting Stations and Total Trips - Members only .



Share

Besides this R markdown document, below is a link to a report and dashboard made in Tableau Public that can be sent to the marketing department as well. The Tableau Public report is set up as a presentation that goes over the analyis findings and the recommendations. There is also a link to the code documentation listed below the Tableau Public link:

Tableau Public Report
https://public.tableau.com/app/profile/paul.windsor/viz/GoogleDataAnalyticsProject-CyclisticCaseStudy/Story1

Cyclistic Data Analysis Code Documentation
https://github.com/pwindsor37115/Paul-W-s-Google-Data-Analytics-Capstone-Project/blob/main/Cyclistic%20Data%20Analysis.R

More documentation for this project can also be found on this Github repository page
https://github.com/pwindsor37115/Paul-W-s-Google-Data-Analytics-Capstone-Project —

Act

Top three recommendations given to the marketing department:

- More data and information is needed from the rest of the team to properly use this data. The large amount of data that had to be removed because it was incomplete is a big concern. This affects the report negatively and the ability to come to any conclusions. Also there are the issues of multiple station names for each station id, missing start station data on rows but end station data is populated, etc. Examples of data issues such as these that need to be reviewed and also will need a better description of the business processes and how data is recorded. Also another tool other than Tableau Public needs to be used due to the limit on data that can be uploaded - yearly trends could not be examined b/c of this.
- At the stations, offer a discount to casual riders if they fill out a survey to get more information on how and why they purchase day passes. The data suggests that member riders may be using the bikes more for traveling to work or nearby their home, while the casual riders appear to be using the bikes more for recreational purposes. Maybe there are customers who live downtown but are not members that may be better persuaded into purchasing a membership if we can gather more information from them as well.
- Offer seasonal passes, instead of just day passes, for bikes while offering other services. Begin advertising the seasonal passes during the spring months in March and April.