Bike-Share Case Study Report

Milagros N. Cortez

2023-07-05

Abstract

This case study is meant to be an exercise to explore and analyze data for a fictional bike-share company from Chicago called Cyclistic as part of the Google Data Analytics certificate. Using R to perform the analysis, we find differences between casual and annual members ride duration time, location, ride preference, and days of use.

Introduction

Cyclistic is a bike-share company from Chicago that offers casual riders and annual members a bike-share program which features more than 5,800 bicycles and 600 docking stations. The company offers a wide range of bikes including traditional bikes, reclining bikes, hand tricycles, and cargo bikes to make the bike-share experience more inclusive to a wider range of riders. In this report we will use key data analytics concepts on the company's data from June 2022 to May 2023 to find differences between casual riders and annual members, and find factors that influence the number of annual memberships. With the purpose of finding how to best promote the bike-sharing program to maximize the number of annual memberships among users which can be used by Cyclistic's marketing branch.

Data

Cyclistic has data from 2013 to 2023 of the trips and docking stations. Looking at all the data from 2013 to 2023 we found that data from 2013 to 2019 is in a different format from the data from 2020 to 2023. Moreover, we will proceed to use the data available from the last 12 months from June 2022 to May 2023 since it is the most recent data and the format of this data is consistent as it has the same variables. The variables in this data include:

- ride_id: unique value assigned for rides
- rideable_type: shows the type of bike used by a customer
- started_at: date-time stamp of when the ride started
- ended_at: date-time stamp pf when the ride ended
- Start_sation_name: name of identified station where the ride started.
- Start_station_id: ID of start station
- End_sation_name: name of identified station where the ride ended
- End_station_id: ID of end station
- start_lat: latitude coordinate where the customer's ride started
- start_lng: longitude coordinate where the customer's ride started
- end_lat: latitude coordinate where the customer's ride ended
- end_lng: longitude coordinate where the customer's ride ended
- member_casual: variable that mentions whether the customer is an annual member or a causal user

Before using the data we proceed to download it from Cyclistic's historical trip data (https://divvy-tripdata.s3.amazonaws.com/index.html)[1]. This site includes zip files of the company's data from 2013 until May 2023 which was provided by the Google Data Analytics certificate. Given the context that the data comes directly from the company and was collected by the company it is known as internal data. From the desired time frame, the data is separated into months. We proceeded to download the zip files for each month which included a .csv file of data. These .csv files were combined into a single .xlsx file under the name of "202206_202305_tripdata.xlsx" and additional variables were added:

- ride_length: the length of time a customer used a ride which is the difference between started_at and ended_at date-time variables. Variable values were created through the excel command (= D2-C2).
- day_of_week: the day of the week the ride was used extracted from the date-time variable started_at. Values of this variable were created using = WEEKDAY() command.

- start_id: binary variable that indicates whether the start station is identified (0) or not (1). Values were created using the excel command = COUNTBLANK().
- end_id: binary variable that indicates whether the end station is identified (0) or not (1) Values were created using the excel command = COUNTBLANK().
- stations_not_id: categorical variable that indicates if all the stations were identified (2), one was identified (1), or none were identified. Values of this variable were found by adding the start_id and end_id.

The data was cleaned and organized in R as outlined below. With this data we proceeded to use only the variables rideable_type, start_lat, start_lng, end_lat, end_lng, member_casual, ride_length, day_of_week, start_id, end_id, and stations_not_id for the analysis. We 1) use the variables associated with time, stations, and customer status to get an overall picture of how customer status and location preferences have changed in the last 12 months, 2) use the variables associated with ride type and location which are only available from from January 2020 to May 2023 to find trends in location and ride type, and 3) use the variables for station ID to find trends on station locations.

Before proceeding with the analysis, using R, we proceed to upload the packages, and the .xlsx file of the compiled data:

```
# Load packages
library(readxl)
library(dplyr)
library(egg)
library(ggplot2)
library(sf)
library(lubridate)
```

In the .xlsx file, each month is stored its own sheet and has to be uploaded individually in R. We only upload the columns of the variables we are interested in using for this analysis:

```
# Upload the data from June 2022 to May 2023
# The variables of each month we are interested in using are rideable_type,
# ride_length, day_of_week, start_lat, start_lng, end_lat, end_lng,
# member_casual, start_id, end_id, and stations_not_id.

jun22 = read_excel("202206_202305_tripdata.xlsx", sheet = 12, col_types = c("skip", "text","skip", "skip", "gue
ss", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
jul22 = read_excel("202206_202305_tripdata.xlsx", sheet = 11, col_types = c("skip", "text","skip", "skip", "gue
ss", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
aug22 = read_excel("202206_202305_tripdata.xlsx", sheet = 10, col_types = c("skip", "text","skip", "skip", "gue
ss", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
sep22 = read_excel("202206_202305_tripdata.xlsx", sheet = 9, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
oct22 = read_excel("202206_202305_tripdata.xlsx", sheet = 8, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
nov22 = read_excel("202206_202305_tripdata.xlsx", sheet = 7, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
dec22 = read_excel("202206_202305_tripdata.xlsx", sheet = 6, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
jan23 = read_excel("202206_202305_tripdata.xlsx", sheet = 5, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
feb23 = read_excel("202206_202305_tripdata.xlsx", sheet = 4, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
mar23 = read_excel("202206_202305_tripdata.xlsx", sheet = 3, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
apr23 = read_excel("202206_202305_tripdata.xlsx", sheet = 2, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
may23 = read_excel("202206_202305_tripdata.xlsx", sheet = 1, col_types = c("skip", "text","skip", "skip", "gues
s", "numeric", "skip","skip", "skip","skip", "numeric", "numeric", "numeric",  "numeric",  "numeric",  "numeri
c",  "numeric", "text" ))
```

We proceed to clean the data and get rid of rows with NULL values, format the ride_length variable, and complie the data into a single data frame:

```
# deleting rows with NA values

jun22 = na.omit(jun22)
jul22 = na.omit(jul22)
aug22 = na.omit(aug22)
sep22 = na.omit(sep22)
oct22 = na.omit(oct22)
nov22 = na.omit(nov22)
dec22 = na.omit(dec22)
jan23 = na.omit(jan23)
feb23 = na.omit(feb23)
mar23 = na.omit(mar23)
apr23 = na.omit(apr23)
may23 = na.omit(may23)

# Correcting the time variable
jun22$ride_length = format(as.POSIXct(jun22$ride_length), format = "%H:%M:%S")
jul22$ride_length = format(as.POSIXct(jul22$ride_length), format = "%H:%M:%S")
aug22$ride_length = format(as.POSIXct(aug22$ride_length), format = "%H:%M:%S")
sep22$ride_length = format(as.POSIXct(sep22$ride_length), format = "%H:%M:%S")
oct22$ride_length = format(as.POSIXct(oct22$ride_length), format = "%H:%M:%S")
nov22$ride_length = format(as.POSIXct(nov22$ride_length), format = "%H:%M:%S")
dec22$ride_length = format(as.POSIXct(dec22$ride_length), format = "%H:%M:%S")
jan23$ride_length = format(as.POSIXct(jan23$ride_length), format = "%H:%M:%S")
feb23$ride_length = format(as.POSIXct(feb23$ride_length), format = "%H:%M:%S")
mar23$ride_length = format(as.POSIXct(mar23$ride_length), format = "%H:%M:%S")
apr23$ride_length = format(as.POSIXct(apr23$ride_length), format = "%H:%M:%S")
may23$ride_length = format(as.POSIXct(may23$ride_length), format = "%H:%M:%S")

# Creating a df of the data from the last 12 months that makes it easier to analyze the data as a whole
past12mon = bind_rows(jun22 ,jul22 ,aug22 ,sep22, oct22, nov22, dec22, jan23, feb23, mar23, apr23, may23)
```

After this, the data is clean and ready to be used in the analysis.

Analysis

Given the case and the data available we want to explore in this analysis:

- What is the overall distribution of ride time?
- What are the ride time distributions for each customer type?
- what is the customer volume for each day of the week, and what is the volume of each customer type for each day?
- Given the latitude-longitude coordinates for the start and end of each customers ride, are there any differences between customer type and station locations across Chicago?
- Are all start-end locations registered as recognized stations?
- What type of rides have been used by customers, and are there any ride type differences between types of customers?

We first explore the ride length variable to find the overall distribution of time as well as the distribution of time for each type of customer

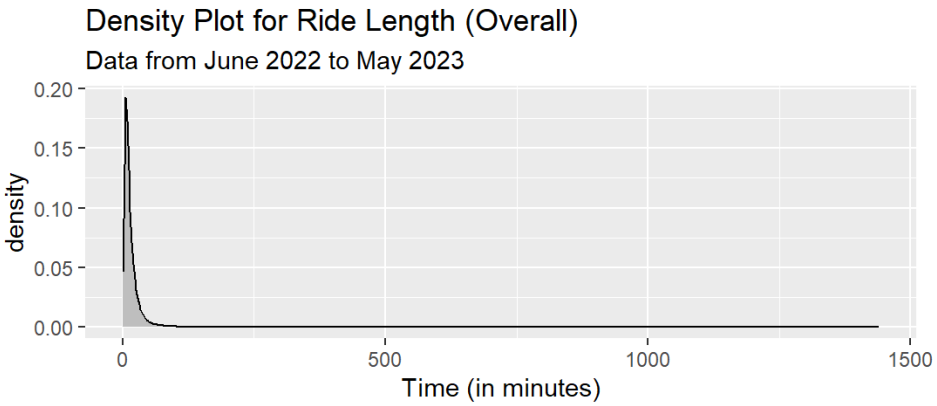
```
# First we analyze time
# obtaining summary statistics for time for all the data:
ridet12 = past12mon %>% summarize(
  min_ride_time = min(period_to_seconds(hms(past12mon$ride_length)))/60,
  med_ride_time = median(period_to_seconds(hms(past12mon$ride_length)))/60,
  mean_ride_time = mean(period_to_seconds(hms(ride_length)))/60,
  max_ride_time = max(period_to_seconds(hms(past12mon$ride_length)))/60,
  sd_ride_time = sd(period_to_seconds(hms(past12mon$ride_length)))/60,.groups = 'drop') %>% as.data.frame()

# Obtaining summary stats for the time spent by casuals in all the data:
ridet12_cm = past12mon %>% group_by(member_casual) %>% summarize(
  min_ride_time = min(period_to_seconds(hms(ride_length)))/60,
  med_ride_time = median(period_to_seconds(hms(ride_length)))/60,
  mean_ride_time = mean(period_to_seconds(hms(ride_length)))/60,
  max_ride_time = max(period_to_seconds(hms(ride_length)))/60,
  sd_ride_time = sd(period_to_seconds(hms(ride_length)))/60, .groups = 'drop') %>% as.data.frame()

# Creating table in R Markdown
ridet12comb = bind_rows(ridet12, ridet12_cm)[1:3, 1:5]
colnames(ridet12comb) = c("Min",  "Med", "Mean", "Max", "Standard Deviation")
rownames(ridet12comb) = c("Overal", "Casual Riders", "Annual Members")
knitr::kable(ridet12comb, "simple")
```

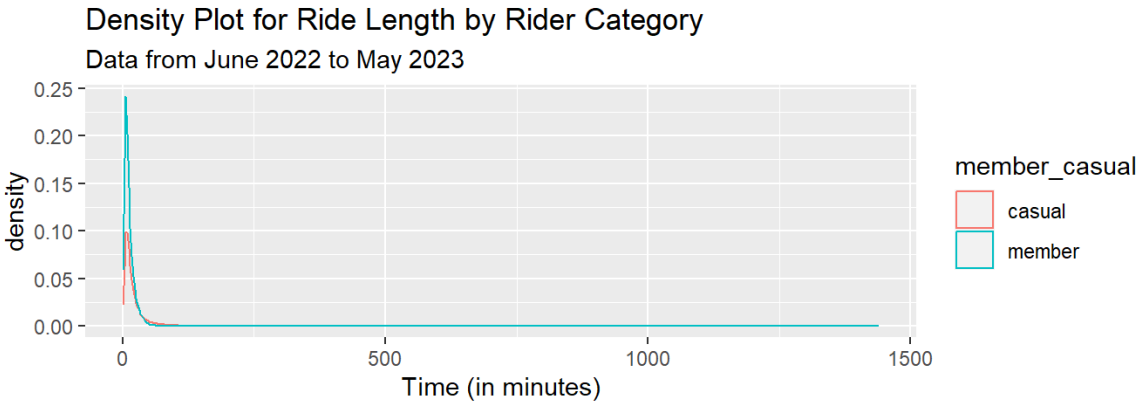
	Min	Med	Mean	Max	Standard Deviation
Overal	0	9.850000	15.55255	1439.983	28.87633
Casual Riders	0	12.216667	20.72511	1439.983	38.30511
Annual Members	0	8.616667	12.15899	1439.983	19.72796

From the table above we can see that annual members have a lower mean and median ride time while casual riders have a higher mean an median ride time. The high standard deviation for casual riders also indicates there is a larger spread of ride times from the mean for casual riders which well us that there is a greater variation between ride times of these type of riders.

From this summary table we can also see that the ride time distribution of overall, casual, and annual members is right skewed. We proceed to plot the denity distributions of each below:

```
# plotting the distributions for all customers, casual members, and annual members
tg1 = ggplot(past12mon, aes(x = period_to_seconds(hms(ride_length))/60))  +
  geom_density( fill = "grey") +
  labs( x = "Time (in minutes)", title = "Density Plot for Ride Length (Overall)", subtitle = "Data from June 2
022 to May 2023")
tg2 = ggplot(past12mon, aes(x = period_to_seconds(hms(ride_length))/60)) +
  geom_density(aes(group = member_casual, color = member_casual)) +
  labs( x = "Time (in minutes)", title = "Density Plot for Ride Length by Rider Category", subtitle = "Data fro
m June 2022 to May 2023")
# combining into a single plot
figure1 = ggarrange(tg1, tg2, ncol = 1, nrow = 2)
```

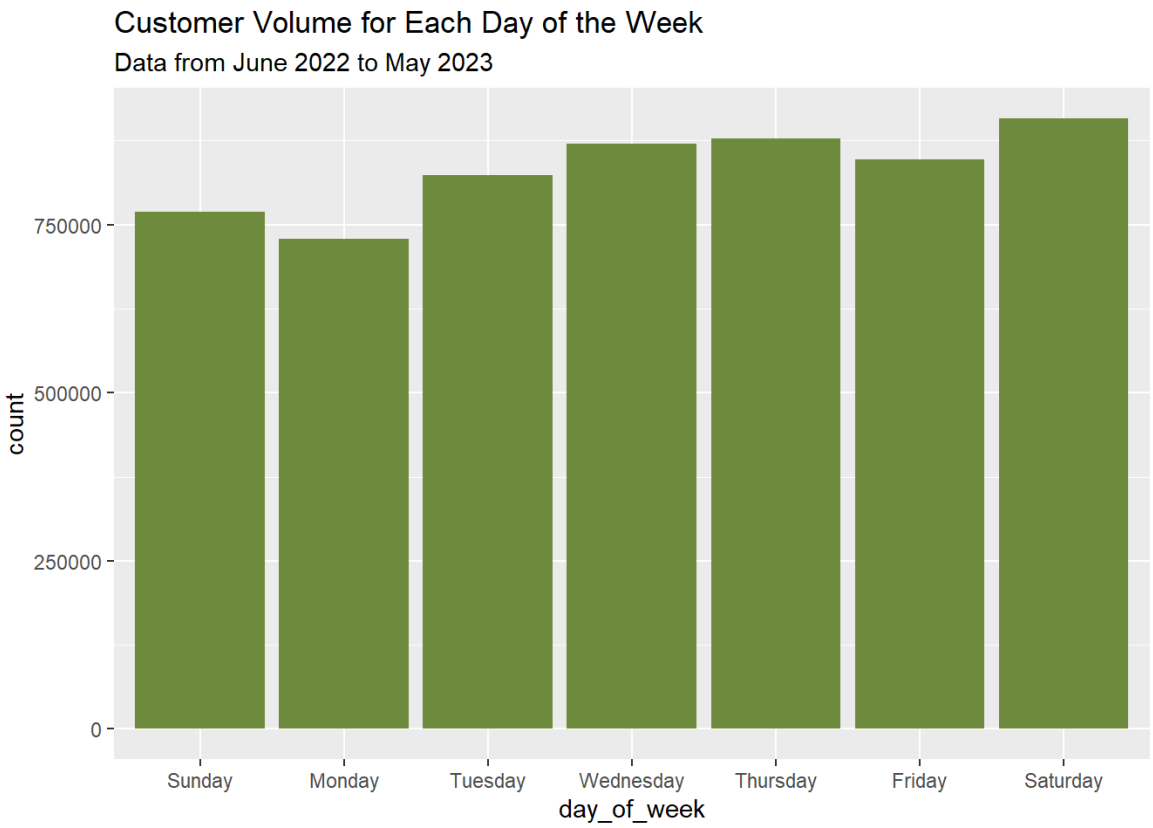




We can see that all density plots are similar in shape (right skewed distribution), and spread from 0 to 1439.983 which is roughly 24 hours. The density plot for casual members appears to have a lower peak in comparison to the density distribution of annual members indicating there are more casual customers what have a longer ride time in comparison to annual members.

We now proceed to see customer volume per week. We create a bar plot between customer volume and day of the week:

```
# We proceed to analyze days of the week

figure2 = ggplot(past12mon, aes(x= day_of_week)) + geom_bar(fill = "darkolivegreen4") +labs(title = "Customer V
olume for Each Day of the Week", subtitle = "Data from June 2022 to May 2023") + scale_x_discrete(limits = c
("1", "2", "3", "4", "5", "6", "7"), labels = c( "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Frida
y", "Saturday"))
figure2
```



From this plot we can see that Saturday is the day with the greatest customer volume followed by Thursday and Wednesday respectively. Days with the lowest customer volume are Monday, Sunday and Tuesday, with Monday having the lowest customer volume.

Below we can find a summary of ride time per week:

```
# finding the average ride length for days of the week as well as min and max times
ridet12_days = past12mon %>% group_by(day_of_week) %>% summarize(
  min_ride_time = min(period_to_seconds(hms(ride_length)))/60,
  med_ride_time = median(period_to_seconds(hms(ride_length)))/60,
  mean_ride_time = mean(period_to_seconds(hms(ride_length)))/60,
  max_ride_time = max(period_to_seconds(hms(ride_length)))/60,
  sd_ride_time = sd(period_to_seconds(hms(ride_length)))/60, .groups = 'drop') %>% as.data.frame()


# Creating table in R Markdown
dayplot = ridet12_days[1:7, 2:6]
colnames(dayplot) = c( "Min", "Med", "Mean", "Max", "Standard Deviation")
rownames(dayplot) = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
knitr::kable(dayplot, "simple")
```

	Min	Med	Mean	Max	Standard Deviation
Sunday	0	11.400000	18.64659	1439.950	34.33729
Monday	0	9.150000	14.72329	1439.983	27.71463
Tuesday	0	9.066667	13.95502	1439.950	27.21102
Wednesday	0	9.183333	13.75239	1439.933	24.16568
Thursday	0	9.333333	14.16098	1439.917	25.74119
Friday	0	9.750000	15.24053	1439.983	27.97143

	Min	Med	Mean	Max	Standard Deviation
Saturday	0	11.550000	18.40879	1439.900	33.28479

From this summary we can see that the summary values from Monday to Friday are fairly consistent with each other, while Saturday and Sunday have the highest values indicating longer tide times during these days.

Now we proceed to beak this up more to create a summary of ride times for each day of the week for each customer type:

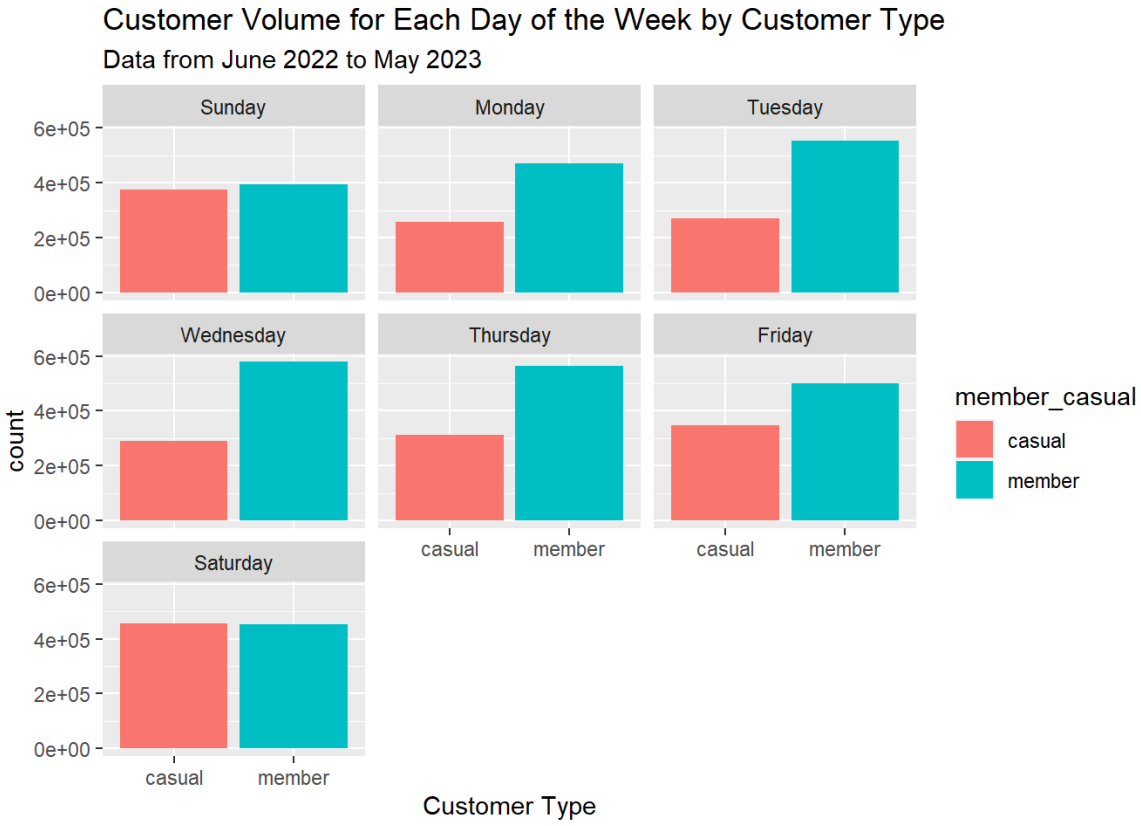
```
# Analyzing ride time differences between customer times for each day

ridet12_days_cm = past12mon %>% group_by(member_casual, day_of_week) %>% summarize(min_ride_time = min(period_t
o_seconds(hms(ride_length)))/60,
                                                                                   med_ride_time = median(period_to_seconds(hms(r
ide_length)))/60,
                                                                                   mean_ride_time = mean(period_to_seconds(hms(ri
de_length)))/60,
                                                                                   max_ride_time = max(period_to_seconds(hms(ride
_length)))/60,
                                                                                   sd_ride_time = sd(period_to_seconds(hms(ride_l
ength)))/60, .groups = 'drop') %>% as.data.frame()
ridet12_days_cm[,2] = c( "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunda
y", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")

daycatime = ridet12_days_cm[1:14, 1:7]
colnames(daycatime) = c("Customer Type", "Day", "Min", "Med", "Mean", "Max", "Standard Deviation")
knitr::kable(daycatime, "simple")
```

Customer Type	Day	Min	Med	Mean	Max	Standard Deviation
casual	Sunday	0	14.233333	24.03045	1439.950	42.96662
casual	Monday	0	11.633333	20.55708	1439.983	38.25256
casual	Tuesday	0	10.866667	18.62818	1439.950	37.14418
casual	Wednesday	0	10.816667	17.95418	1437.583	33.72268
casual	Thursday	0	11.100000	18.49135	1439.917	35.58091
casual	Friday	0	11.966667	19.91925	1439.983	36.59336
casual	Saturday	0	14.133333	23.26822	1439.817	40.37293
member	Sunday	0	9.316667	13.53310	1439.867	22.17642
member	Monday	0	8.133333	11.53911	1439.983	18.97260
member	Tuesday	0	8.333333	11.66756	1439.950	20.28828
member	Wednesday	0	8.466667	11.63909	1439.933	17.09922
member	Thursday	0	8.500000	11.76313	1439.900	17.67376
member	Friday	0	8.516667	12.01362	1439.933	19.31207
member	Saturday	0	9.500000	13.52696	1439.900	23.14781

We can see from this table that casual members have higher median and mean ride times for each day of the week compared to annual members. With Saturday, Sunday, and Monday having the highest mean and median values. In particular, Sunday has the highest mean and median values. Looking at the summary values of each day for annual members, we can see that Friday, Saturday, and Sunday are days that have a high mean and median ride_time. Saturday has the highest median ride time, while Sunday has the highest mean ride_time.

We porceed to plot the volume of each type of customer for each day of the week:

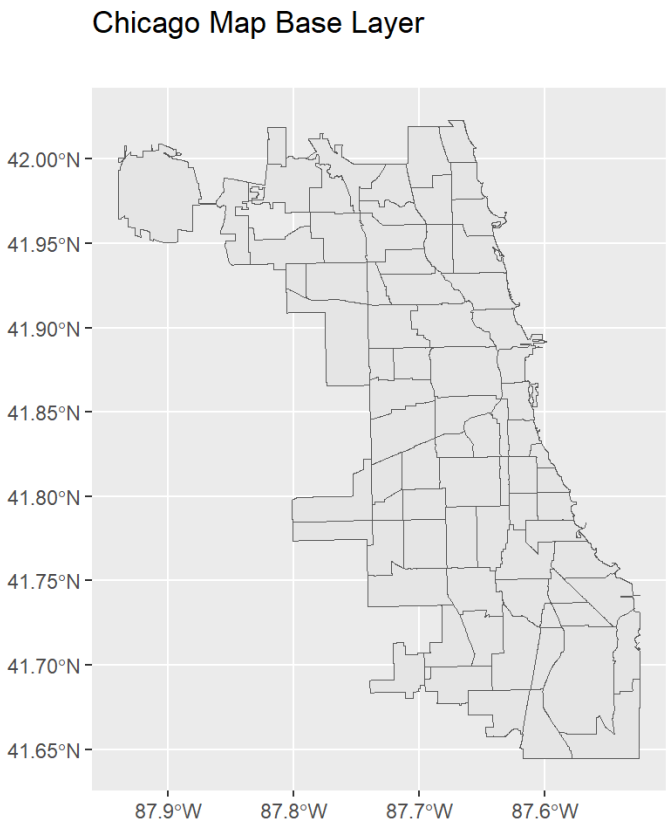
```
# breaking customer volume to see of there are differences between days and types of customers
day_names = c( "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
names(day_names) = c( "1", "2", "3", "4", "5", "6", "7")
figure3 = ggplot(past12mon, aes(x= member_casual)) +
  geom_bar(aes(fill = member_casual)) +
  labs(x = "Customer Type", title = "Customer Volume for Each Day of the Week by Customer Type", subtitle = 'Da
ta from June 2022 to May 2023') + facet_grid(~ day_of_week, labeller = labeller(day_of_week = day_names)) + fac
et_wrap(~ day_of_week, ncol = 3, labeller = labeller(day_of_week = day_names))

figure3
```



From the plots above we can see that Sunday and Saturday have a roughly equal volume of casual and annual members, but there is a higher volume of casual members compared to annual members from Monday to Friday. What could lead to a higher volume of casual members from Monday to Friday? It could be causals who commute to work. It would be nice to look further into why casual customer volumes are higher from Monday to Friday.

Now given the latitude and longitude coordinates for start and end locations for each ride, we proceed to plot the start and end ride locations for each customer type in the city of Chicago. To do this, we first get a base layer of the city. The base layer used is in a GeoJSON format which was sourced from the Chicago Data Portal (data.cityofchicago.org) [2] and was found in Daryn Ramsden's Intro to Making maps with ggplot2 (thisisdaryn.netlify.app/post/intro-to-making-maps-with-ggplot2/)[3]. The base map can be seen below:

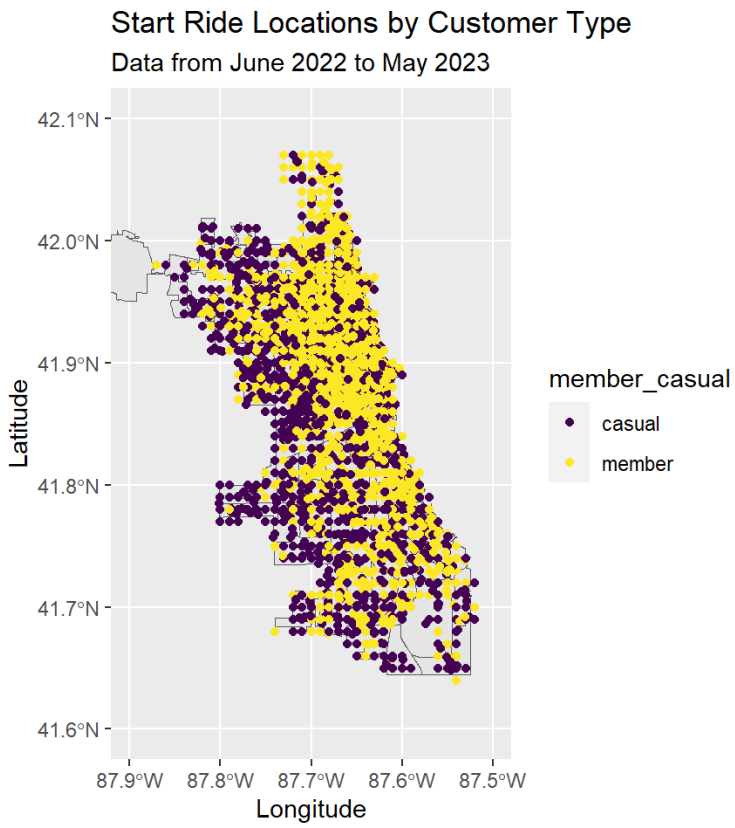
```
# Create the base layer of the map
city_map = read_sf("https://raw.githubusercontent.com/thisisdaryn/data/master/geo/chicago/Comm_Areas.geojson")
ggplot(data = city_map, fill = "darkseagreen2", color = "darkseagreen2") + geom_sf() + labs(title = "Chicago Ma
p Base Layer", subtitle = "")
```



This map includes the 77 identified community areas in Chicago.

Using the base map, we plot the starting locations of casual and annual members:

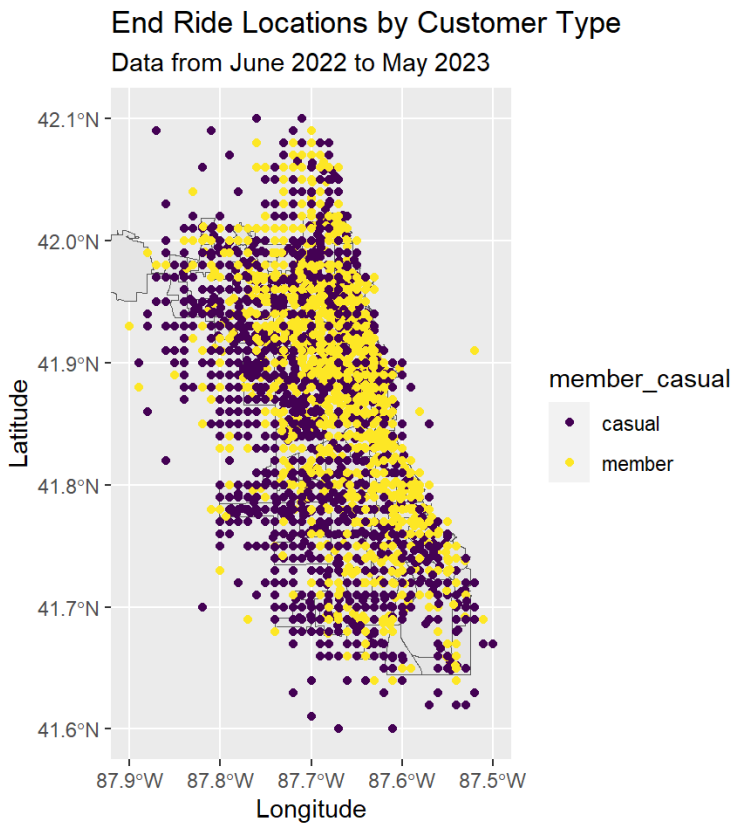
```
# map plotting start locations
ggplot(data = city_map, fill = "darkseagreen2", color = "darkseagreen2") + geom_sf() + scale_x_continuous(name=
"Longitude", limits = c(-87.9, -87.5)) + scale_y_continuous(name = "Latitude", limits = c(41.6, 42.1)) + geom_p
oint(data = past12mon, aes(x = start_lng, y = start_lat, col = member_casual)) + labs(title = "Start Ride Locat
ions by Customer Type", subtitle = 'Data from June 2022 to May 2023') + scale_color_viridis_d()
```



From the map we can see that there is a greater concentration of casual members starting their rides at the western-most locations of the city. We proceed to plot the end location of each customer type below:

```
# map plotting end locations
ggplot(data = city_map, fill = "darkseagreen2", color = "darkseagreen2") + geom_sf()+ scale_x_continuous(name=
"Longitude", limits = c(-87.9, -87.5)) + scale_y_continuous(name = "Latitude", limits = c(41.6, 42.1)) + geom_p
oint(data = past12mon, aes(x = end_lng, y = end_lat, col = member_casual)) + labs(title = "End Ride Locations b
y Customer Type", subtitle = 'Data from June 2022 to May 2023') + scale_color_viridis_d()
```

```
## Warning: Removed 34 rows containing missing values (`geom_point()`).
```



In this map we can see that the some of the end locations are outside of Chicago, and a great amount of these points are from casual customers. Additionally, some points were outside of the state of Illinois, which have been included in the map. As the data is of a fictional company there, isn't much to do other than omit these points outside of the state, but it would be interesting on investigating these points further if the data was from a real case.

Since the name of start and end locations was sometimes included in the information fo customer's rides, we also looked a bit into seeing how many rides included recognized start and end locations. Strart or end locations that are recognized would have a name of a station in the start_sation_name or end_station_name variables. We transformed this data into two binary variables and a categorical variable (start_id, end_id, and stations_not_id).

```
# confusion matrix between start stations identified and customer type
tab = table(past12mon$start_id, past12mon$member_casual)
perctab = round(tab/sum(tab), 3)
perctab
```

```
##
##     casual member
##   0  0.328  0.520
##   1  0.068  0.084
```

From the confusion matrix above, we can see that 52% of customers are members and their starting location for their ride is recognized as a station in the system, 32.8% of customers are casual a their starting location is registered as a station. About 15.2% of rides have a starting station that is not recognized.

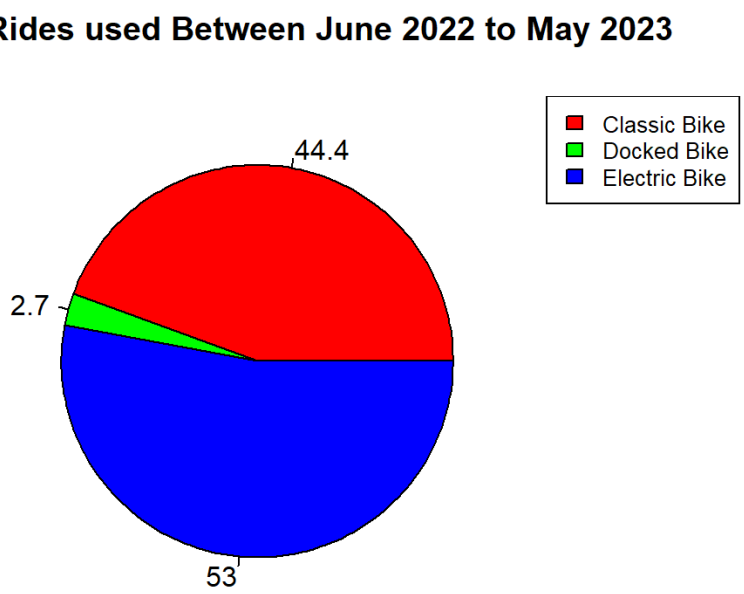
```
# confusion matrix between end stations identified and customer type
tab2 = table(past12mon$end_id, past12mon$member_casual)
perctab2 = round(tab2/sum(tab2), 3)
perctab2
```

```
##
##     casual member
##   0  0.328  0.520
##   1  0.069  0.084
```

The confusion matrix between customer type and end station recognized is similar to the previous confusion table, but we can see that the percent of casual members with a non recognized end station is higher at 6.9%.

Lastly, we look at types of rides used by customers. We create a pie chart to show the types of rides used by riders throughout the past 12 months.

```
# We proceed to analyze bike type
vtab = table(past12mon$rideable_type)
values = c(vtab[1], vtab[2], vtab[3])
perc<- round(100*values/sum(values), 1)
pie(values, labels = perc, main = "Types of Rides used Between June 2022 to May 2023" ,col = rainbow(length(val
ues)))
legend("topright", c("Classic Bike", "Docked Bike", "Electric Bike"), cex = 0.8, fill = rainbow(length(value
s)))
```



From the pie chart above, we can see that 53% of customers, use electric bikes, 44.4% of users use classic bikes, and 2.7% of users use docked bikes. Given that Cyclistic offers different types of bikes including reclining bikes, hand trycicles, and cargo bikes, it would be interesting to see if these are also registered under the variable rideable_type or if they could be registered in future data sets.

We further analyze ride type with type of customer:

```
# We proceed to analyze bike type with customer type
vtab = table(past12mon$rideable_type, past12mon$member_casual)
vperc<- round(100*vtab/sum(vtab), 1)
vperc
```

```
##
##                 casual member
##   classic_bike    14.7   29.7
##   docked_bike      2.7    0.0
##   electric_bike   22.3   30.7
```

From the confusion matrix above, we can see that 2.7% of customers are casual customers that prefer dock_bikes, which matches the percentage of docked bikes from the pie chart. We can also see that annual members prefer classic and electric bikes. there are higher percentages of casual and annual members that prefer electric bikes compared to classic bikes.

Summary of Findings and Conclusion

From the analysis above, we can see trends between casual and annual members. Casual members have longer ride times and higher variation. Even though the day of the week with the highest volume of customers is Saturday, and Saturday and Sunday are days where customers have the longest ride times, we find that there is a higher volume of casual customers from Monday to Friday and these type of customers also have the longest ride times throughout the week in comparison to annual customers, specially in Sunday and Saturday. When plotting the location of start and end stations for casual and annual members we find that there is a concentration of casual customers on the west side of the city. Start locations for both casual and annual members is within the 77 identified community areas in Chicago, but the location of some end locations is outside the 77 identified community areas in Chicago with most points belonging to casual members. Lastly, we found that the 2.7% of overall customers who used docked bikes in the past 12 months from June 2022 to May 2023 were casual customers. Causal customers used the three types of bikes in comparison to members who either used classic bikes or electric bikes.

We can see that the factors we found differences in consumers membership status are days of the week, location, ride time, and ride type. These factors should be considered if we want to maximize the number of annual memberships. Based on this findings it is suggested that we look into finding ways to improve the annual membership (promote longer rides or weekday perks under the membership) by getting customer data, particularly casual customers, about how the bike-share service could improve, and use this analysis within the Cyclistic's marketing branch to consider new marketing strategies to promote the annual membership program to customers. Perhaps advertising the annual membership program to casual customers during weekdays and stations on the west of the city could be considered within strategies to maximize annual memberships and converting casual members to the membership program.

References

[1] Cyclistic's historical trip data. (n.d). Case Study 1: How does a bike-share navigate speedy success?. Coursera. https://divvy-tripdata.s3.amazonaws.com/index.html (https://divvy-tripdata.s3.amazonaws.com/index.html)

- Note that the .zip files downloaded are: 202305-divvy-tripdata.zip, 202304-divvy-tripdata.zip, 02303-divvy-tripdata.zip, 202302-divvy-tripdata.zip, 202301-divvy-tripdata.zip, 202212-divvy-tripdata.zip, 202211-divvy-tripdata.zip, 202210-divvy-tripdata.zip, 202209-divvy-tripdata.zip, 202208-divvy-tripdata.zip, 202207-divvy-tripdata.zip, 202206-divvy-tripdata.zip

[2] City of Chicago. (n.d.). City of chicago: Data Portal: City of chicago: Data Portal. City of Chicago. https://data.cityofchicago.org/ (https://data.cityofchicago.org/)

[3] Ramsden, D. (2020, March 28). Intro to making maps with GGPLOT2. Intro to Making maps with ggplot2. https://thisisdaryn.netlify.app/post/intro-to-making-maps-with-ggplot2/ (https://thisisdaryn.netlify.app/post/intro-to-making-maps-with-ggplot2/)

[4] Case Study 1: How does a bike-share navigate speedy success?. (n.d). Google Data Analytics Capstone: Complete a Case Study. Coursera. https://d3c33hcgiwev3.cloudfront.net/aacF81H_TsWnBfNR_x7FIg_36299b28fa0c4a5aba836111daad12f1_DAC8-Case-Study-1.pdf?Expires=1689120000&Signature=Js-MRPXHfBlfHFkV5J9dG3~LanLPccZd0uC4hdoigTLtf~f3~YXUigUZuUGQ6mQdOr2qqi-naqxH-oD125Cdk~C7VTeG3PasE~iYmXzIBTgVNeSSTVm6WVwz9Sz8J3iPGtjCcvmmH0mfylMcBCmiJO8e5G0m0anSociswwEKMUo_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A (https://d3c33hcgiwev3.cloudfront.net/aacF81H_TsWnBfNR_x7FIg_36299b28fa0c4a5aba836111daad12f1_DAC8-Case-Study-1.pdf?Expires=1689120000&Signature=Js-MRPXHfBlfHFkV5J9dG3~LanLPccZd0uC4hdoigTLtf~f3~YXUigUZuUGQ6mQdOr2qqi-

naqxH-
oD125Cdk~C7VTeG3PasE~iYmXzIBTgVNeSSTVm6WVwz9Sz8J3iPGtjCcvmmH0mfylMcBCmiJO8e5G0m0anSociswwEKMUo_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)