Transforming Riders: Cyclistic Trip Data Analysis

Katherine Tetlak, Bradford Analytics

Mar 07, 2023

Contents

NOTE: This entire document was written in R Markdown

Disclaimer: This is my version of the Google Data Analytics case study. I wrote this account "as if" I had actually been hired as a data analyst for Cyclistic, but the actual work—preparing, analyzing, sharing the data, and making final recommendations—is my own. I wanted to immerse myself in the job role in an effort to make it more "real," and I wanted to set myself apart from the millions of other students who publish articles entitled "Google Data Analytics Capstone" or something similar. I do not actually work for Cyclistic.

I was hired about six months ago as a junior data analyst with the marketing team for Cyclistic (fictitious company), a bike-share company in Chicago, and I've been busy learning about their business goals and mission, as well as about how I can help them achieve those goals. My manager, Lily Moreno, just gave me my first "real" assignment, and I am beyond excited!

I heard about Cyclistic through the Google Analytics course I was taking. Although it was somewhat outside my comfort zone, I decided to apply as I strongly believe in accessibility, and I want to do everything I can to help such companies grow. Also, if I can help a city reduce its dependence on motor vehicles, thus reducing congestion and pollution, I'm all for it. Bike-sharing is also a great way to get people outside and exercising, meeting new people and making friends, and that too is a good thing.

I learned that Cyclistic has more than 5,800 bikes (docked, classic, electric) and 600 docking stations. They also offer reclining bikes, hand tricycles and cargo bikes, making bike-sharing more inclusive to people with disabilities and riders who can't use a standard two-wheeled bike. This is another reason I wanted to work for them: their commitment to accommodating all types of riders, even those who wouldn't normally consider riding a bike, is impressive. Although the majority of riders use traditional bikes, approximately 8% opt for

the assistive options. Also, Cyclistic users are more likely to ride for leisure, but approximately 30% use them to commute to work every day.

The company offers flexible payment plans through single-ride passes, full-day passes, and annual memberships. Riders who purchase single-ride or full-day passes are considered "casual" riders. Riders who purchase annual memberships are Cyclistic members.

The company believes that its future growth depends on transforming casual riders into annual members. To this end, they want use historical data to understand how casual riders and annual members use Cyclistic bikes to identify any differences between the two and see if there are any trends. That is my current task. The ultimate goal is to create marketing strategies aimed at converting casual riders into annual members.

I began by downloading trip data for 2022, from https://divvy-tripdata.s3.amazonaws.com/index.html. These files were labeled "202201_divvy_tripdata.csv," etc.

The column names were

```
"ride_id"            "rideable_type"      "started_at"         "ended_at"           "start_station_name"
[6] "start_station_id"   "end_station_name"   "end_station_id"     "start_lat"          "start_lng"
[11] "end_lat"            "end_lng"            "member_casual"
```

There were also files denoted as "Divvy_Trips_XXXX_QX.zip" These files contain additional information on gender and year of birth. This data would be useful for analyzing differences by age and gender.

Shall we begin??

Preparing the data: Trip data

I begin by downloading trip data for 2022 and saving the files to my dropbox folder to ensure their integrity.

I then installed and loaded the tidyverse, dplyr, janitor, here, and skimr packages. The "here" package makes it easier to find your files. The "janitor" package has simple tools for examining and cleaning dirty data. The "skimr" package provides compact and flexible summaries of data, and you know the rest.

I discovered a faster way to install packages using the librarian package:

```
library(librarian)

librarian::shelf(here, tidyverse, dplyr, janitor, skimr,ggplot2, magrittr, tinytex)
```

```
##
##   The 'cran_repo' argument in shelf() was not set, so it will use
##   cran_repo = 'https://cran.r-project.org' by default.
##
##   To avoid this message, set the 'cran_repo' argument to a CRAN
##   mirror URL (see https://cran.r-project.org/mirrors.html) or set
##   'quiet = TRUE'.
```

I included this in the "Project" master file, so every time I open the project file, the packages automatically load and I don't have to worry about finding out that a package is missing in the middle of what I'm doing.

I imported the .csv files for each month of 2022 into R using the following command as an example:

```
Jan2022 <- read.csv("cyclistic/data/raw/202201-divvy-tripdata.csv")
```

To create a single dataset, I used the following command (found here) to bind the data frames together into one:

```
bike_data_2020 <- rbind(Jan2020, Feb2020, etc.)
```

I then saved the file to the "combined_raw" data folder.

```
write_csv(bike_data_2020, "cyclistic/data/combined_raw/bike_data_2020.csv")
```

I also downloaded the Divvy_Trips_XXXX_QX.zip files for Q1-Q4 for 2018, Q1-Q4 for 2019. I extracted the corresponding .csv files and saved them to dropbox. I only want the last three columns in each one: user type, gender, birth date. I noticed that there is no consistency between the column headings, even though the values are the same, so that will have to be addressed before I can combine them by year.

To load the files when I reopened the project, I used

```
  bike_data_2022 <- read.csv("cyclistic/data/combined_raw/bike_data_2022.csv")
```

Once I had the combined files for 2022, I viewed the data using

```
head(bike_data_2020)
glimpse(bike_data_2020)
skim_without_charts(bike_data_2020)
```

The results for 2022 indicated that in the "ride_id" column, there were 5,667,717 unique rows out of 5,667,717 total rows, so there were no duplicates.

For data cleaning and manipulation, I carried out the following tasks:

- Eliminated rows with missing data
- Split started/ended_at columns
- Create a new column, "ride_length," representing the difference between "started_" and "ended_at".
- Change the column heading "member_casual" to "member_type"

To remove the duplicate entries in ride_id, I used

```
bike_clean_2022 <- bike_data_2022 %>% distinct(ride_id, .keep_all = TRUE)
```

The columns "end_lat," "end_lng," "end_station_id," and "start_station_id" had many NA values. To remove those values from the datasets I used

```
clean_2022_ver2 <- clean_2022[!(is.na(clean_2022$start_station_id)) &
          !(is.na(clean_2022$start_station_name)) &
          !(is.na(clean_2022$end_station_id)) &
          !(is.na(clean_2022$end_station_name)) &
          !(is.na(clean_2022$start_lat)) &
          !(is.na(clean_2022$start_lng)) &
          !(is.na(clean_2022$end_lat)) &
          !(is.na(clean_2022$end_lng)), ]
```

For the 2022 dataset, 5858 missing values were removed.

To better manage the data, I decided to split the started/ended_at columns into separate "date" and "time" columns. This would also make it easier to format the values as numeric rather than character.

```
clean_2022_ver2 <- separate(clean_2022_ver2, started_at, into = c("start_date", "start_time"), sep = " "
clean_2022_ver2 <- separate(clean_2022_ver2, ended_at, into = c("end_date", "end_time"), sep = " ")
```

These files were also saved.

NOTE: It turns out that I needed to retain the "data-time" format, so I had to reverse this change to keep date and time together. The resulting type is "character," so I'll have to reformat the values to date-time.

I also wanted to round up the started/ended_at times to the nearest minute to avoid having to deal with seconds. I accomplished that using

```
clean_2022_ver2$started_at <- round_date(clean_2022_ver2$started_at, "minute")
clean_2022_ver2$ended_at <- round_date(clean_2022_ver2$ended_at, "minute")
```

As a result, all started_at and ended_at value show :00 for seconds. I confirmed the results by selecting a few "before" and "after" examples and they were all correct.

I then created a new column title "ride_length_min". When I first did this, the time difference was appearing in seconds, even though the value was in minutes. I found the solution here. The code sample was

```
  df <- df %>%
  mutate(trip_duration = difftime(as.POSIXct(‘end time‘), as.POSIXct(‘start time‘), units = "mins"))
```

I replaced the sample code with my own

```
clean_2022_ver2 <- clean_2022_ver2 %>%
    mutate(ride_length_min = difftime(as.POSIXct(‘ended_at‘),
         as.POSIXct(‘started_at‘), units = "mins"))
```

However, the values in ride_length_min were in "difftime" format. I needed them to be numeric, which I accomplished using

```
clean_2022_ver2$ride_length_min = as.numeric(as.difftime(clean_2022_ver2$ride_length_min))
```

I discovered there were negative ride_length_min values. I checked one particular example and found

```
  started_at = 2022-09-28 11:04:32
  ended_at = 2022-09-21 06:31:11
```

Obviously the ride can't have ended (9/21) before it started (9/28), so this is an erroneous value that should be deleted. I checked a few others and got the same results; someone entered start and end times backwards.

I first counted how many rows had negative values in the started_at/ended_at columns:

```
clean_2022_ver2 %>%
  arrange(ride_length_min)

clean_2022_ver2 %>% count(ride_length_min <= 0)
```

I then deleted those rows by filtering using

```
clean_2022_ver2 <- clean_2022_ver2 %>%
 filter(ride_length_min > 0)
```

Are casual riders more likely to ride on particular days of the week or particular months of the year? To find out, I needed to be able to sort riders by weekday and month.

```
clean_2022_ver2$weekday_start <- weekdays(clean_2022_ver2$started_at, abbreviate=FALSE)
clean_2022_ver2$month_start <- format(clean_2022_ver2$started_at, "%m")
View(clean_2022_ver2)
```

I then sorted the dataset by ride_length in descending order.

```
clean_2022_ver2 <- clean_2022_ver2 %>% arrange(-ride_length_min)
```

I then changed the name of the column "member_casual" to "member_type and deleted the original column.

```
clean_2020_ver2 <- clean_2020_ver2 %>% mutate(member_type = member_casual)
clean_2020_ver2 <- select(clean_2020_ver2, -11)
```

Once the data were clean, I saved them as a .csv files to continue my analysis.

```
write_csv(final_2022, "cyclistic/data/cleaned/final_2022.csv")
```

Preparing the data: TBQ files

I imported the "trip data by quarter" (TBQ) .csv files for each quarter of 2019 (the latest full year available) into R using the following command as an example:

```
Divvy_Trips_2019_Q1 <- read.csv("cyclistic/data/TBQ//TBQ_2019/Divvy_Trips_2019_Q1.csv")
```

I only wanted the last three columns, so I deleted the first nine:

```
Divvy_Trips_2018_Q1 <- Divvy_Trips_2018_Q1[, 10:12]
```

NOTE: This was actually a mistake. I needed the **ride_id** column to check for duplicates. I therefore backtracked to retain that column.

The formats for column names in Q1 did not match those of the others, so that needed to be fixed:

```
Divvy_Trips_2019_Q1 <- Divvy_Trips_2019_Q1 %>%
    mutate(usertype = User.Type) %>%
    mutate(gender = Member.Gender) %>%
    mutate(birthyear = X05...Member.Details.Member.Birthday.Year)
```

I then deleted the original columns

```
Divvy_Trips_2018_Q1 = select(Divvy_Trips_2018_Q1, 4:6)
```

and bound the four quarters into one file:

```
TBQ_2019 <- rbind(Divvy_Trips_2019_Q1, Divvy_Trips_2019_Q2, Divvy_Trips_2019_Q3, Divvy_Trips_2019_Q4)
```

Once I had the combined files for 2019, I viewed the data using the format

```
head(bike_data_2029)
glimpse(bike_data_2029)
skim_without_charts(bike_data_2029)
```

There are were missing values for birthyear, usertype, trip_id, and gender.

I checked the columns for missing/NA values using

```
  sapply(TBQ_2018,function(x) table(as.character(x) ==="")["TRUE"])
```

(from https://stackoverflow.com/questions/33848312/count-empty-rows-in-a-data-frame-containing-character-date-and-numeric)

There were NA values in usertype, birthyear, and trip_id; there were 562,505 missing values in gender.

I removed missing values using

```
  TBQ_2019_clean <- TBQ_2019[!(is.na(TBQ_2019$usertype)) &
      !(is.na(TBQ_2019$birthyear)) &
      !(is.na(TBQ_2019$trip_id)) &
      !(is.na(TBQ_2019_clean$gender))]
```

Nothing more needed to be done, and so I saved the files to my dropbox:

```
write_csv(TBQ_2019_clean, "cyclistic/data/TBQ/TBQ_2019_clean.csv")
```

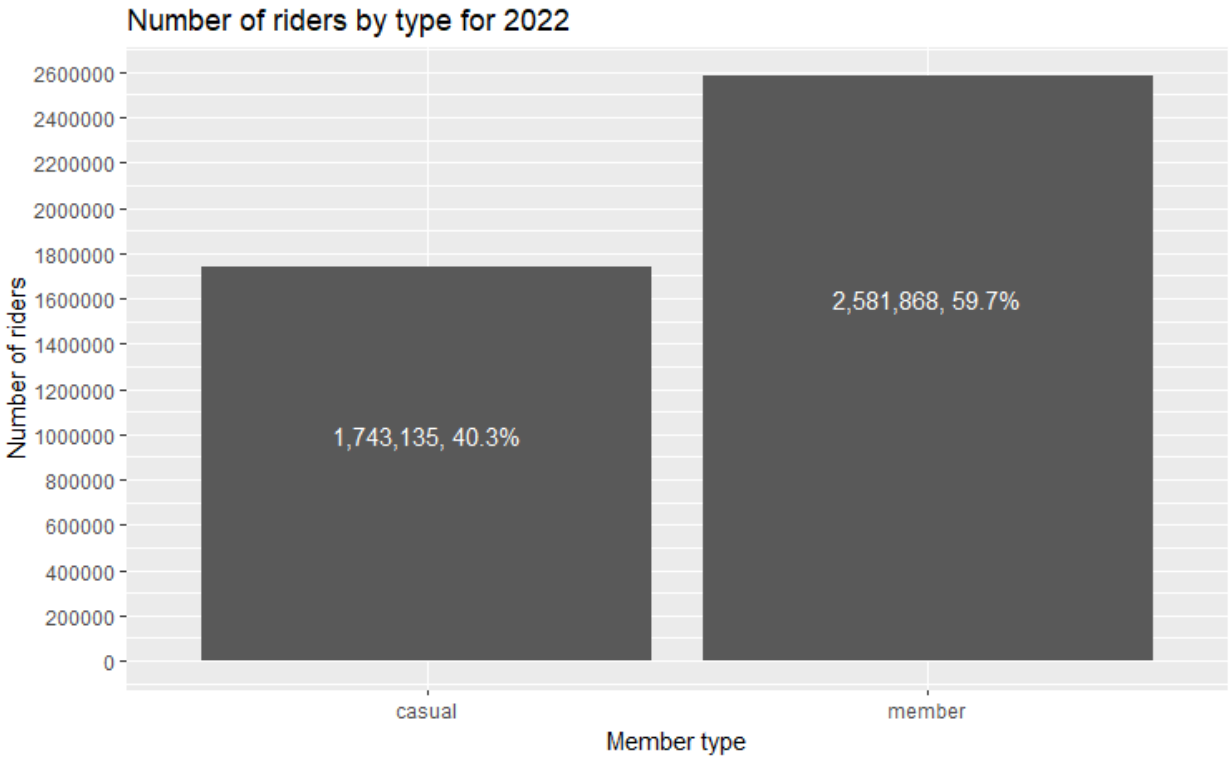
Analyzing the data

After installing the required packages, I imported the cleaned dataset that I saved after the prep stage.

```
  final_2022 <- read.csv("cyclistic/data/cleaned/final_2022.csv")
```

Initial plot of number of members vs. casual riders

To get an overview of ridership, I plotted member type by frequency (number of riders). There are currently 4,325,003 riders overall. Casual riders comprise 40.3% of the total number of riders, whereas members comprise 59.7%.



Casual riders make up approximately 40% of total riders, which gives us a significant source of potential members.

Statistical analyses

To answer the question, "How do casual riders differ from members," I considered the following questions:

- Is there a difference in the mean (average) ride length between the two types of users?
- What is the max/min ride length for each type of user?
- Is there a difference in the day of week, month, or time of day?
- Is there a difference in "rideable_type"? For example, do casual riders favor one type of bike over another?
- Are there differences in terms of gender or age group? (The data with this information is from 2019, and thus it cannot be compared with the data from 2022. Still, some conclusions can be drawn.)

Ride length: max, min, mean grouped by rider_type

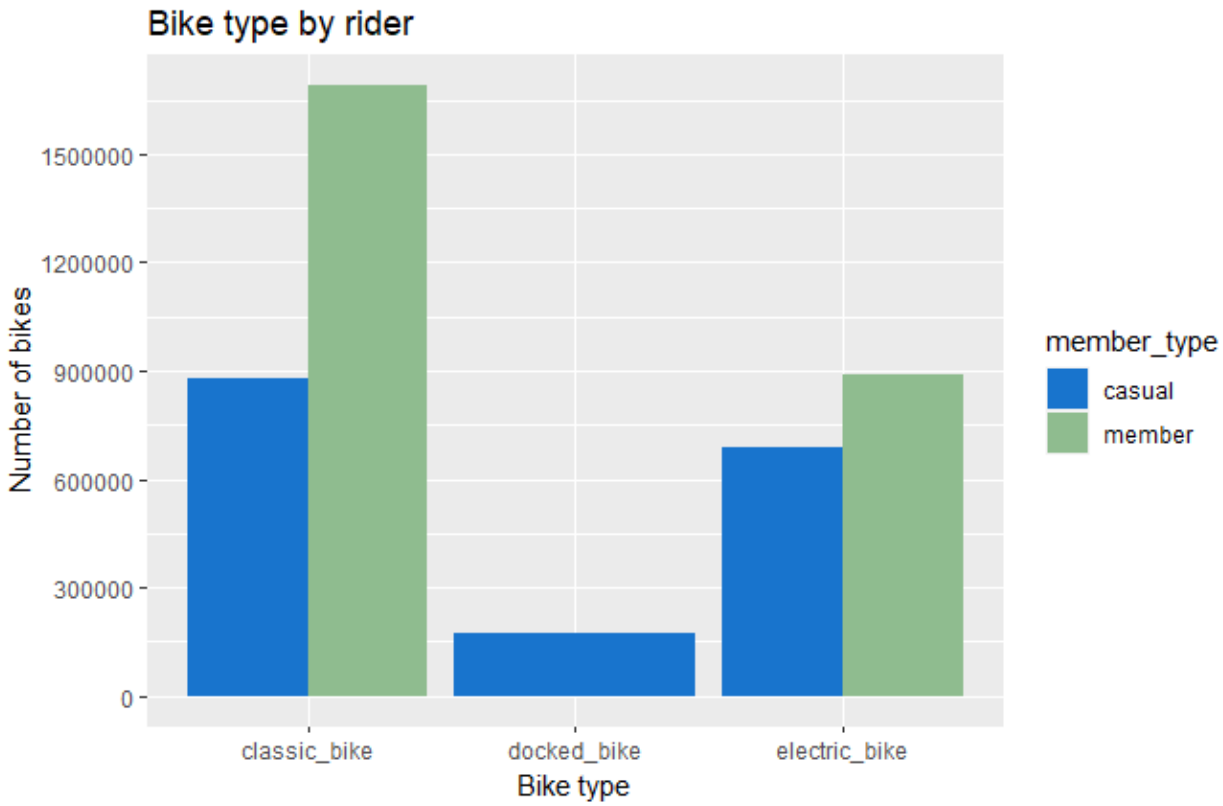
I calculated the mean ride length, as well as the max and min, using

```
stats_full <- final_2022 %>%
    group_by(member_type) %>%
    summarise(meanRL_min = round(mean(ride_length_min), digits = 1), maxRL_min = max(ride_length_min), r
```

Based on the results, the maximum ride length for casual riders in 2022 was approximately 24 **days**; the minimum ride length was 1 minute, and the mean was 24 minutes. For members, the maximum was approximately 25 **hours**, the min was 1 minute, and the mean was approximately 13 min.

Bike type

Is there a difference in "rideable_type"? For example, do casual riders favor one type of bike over another? If so, what do they prefer?

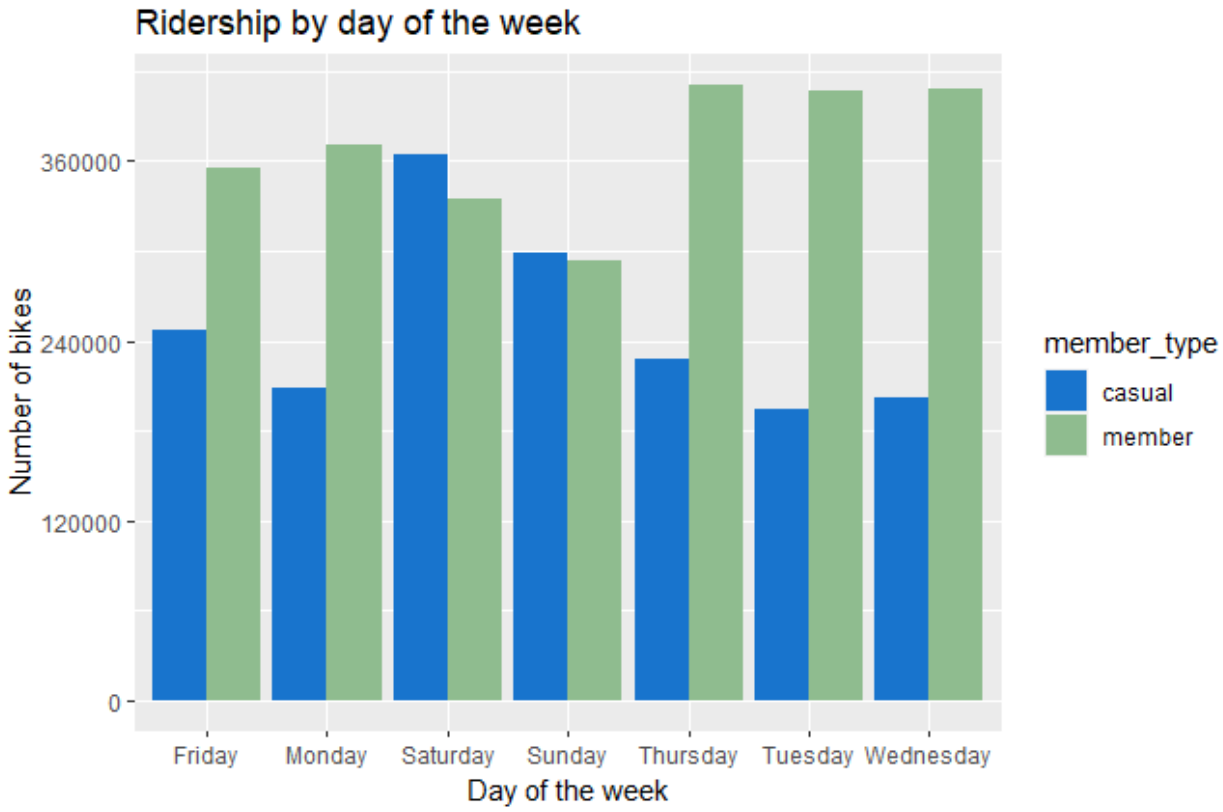


It is apparent that docked bikes are the least used among the three types of bikes and that annual members do not use them at all. Members overwhelmingly choose classic bikes and they do so approximately twice as often as they choose electric bikes. Casual riders predominantly choose classic bikes, with electric bikes chosen somewhat less frequently.

Classic bikes are definitely better for one's health than electric bikes, but I'm not sure how bike type can be used to encourage casual riders to become members.

Ridership by day of week

Is there a difference in the day of the week for each rider type?



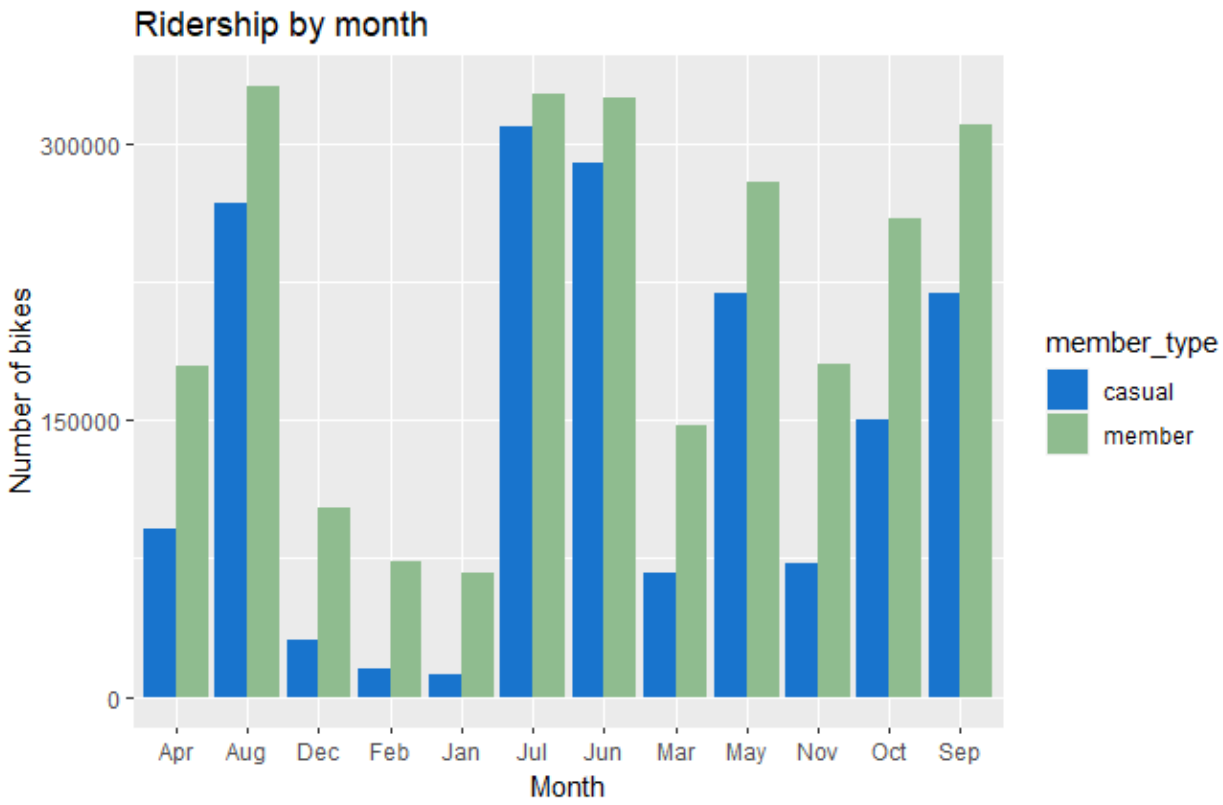
I was unable to get the days of the week in order, even though I tried dozens of solutions, so I'm leaving the chart as is. It shows, however, that bike use among casual riders is highest on the weekend, Saturday and Sunday, with Friday seeing somewhat less use. The other weekdays see fairly consistent use among casual riders. The most prominent days for bike use among members are Tuesday, Wednesday, and Thursday, with Monday and Friday seeing somewhat less frequent use. Thus, bike use is highest among members during the week and highest among casual riders on the weekend. This implies that members use bikes for commuting, whereas casual riders use them for pleasure. This requires further analysis on time of day and mapping locations.

Possible action items to persuade more casual members to purchase annual memberships

- Develop a marketing campaign that highlights the benefits of commuting to work
- Offer casual members discounts for weekday use
- Offer casual members a discounted rate for membership for the first year

Ridership by month

Is there a difference in the month for each rider type?



Again, I was unable to order the months correctly, so I left the chart as is. It shows, however, that the summer months of June, July, and August have the highest number of casual riders, followed by May and September. However, bike use by casual riders never exceeds that by members. The winter months of December, January, and February, have very low ridership among casual riders.

Ridership for members follows the same trend, but bike use for members is still significantly higher in the winter months compared that for to casual riders, which is probably a result of the assumption that members use bikes for commuting to work.

Casual riders may think, "Well, I don't ride in the winter months, so why should I purchase a membership?" This is a good question that requires a response.

Ridership by gender

To determine user type by gender, I switched to SQL and used

```
SELECT usertype, gender, COUNT(usertype) AS CountUsertype
FROM Cyclistic.dbo.TBQ_2019_cleaned
WHERE LEN(age) = 2
GROUP BY usertype, gender
ORDER BY CountUsertype
```

The following results were returned:

```
Female riders:

  131,432 casual
```

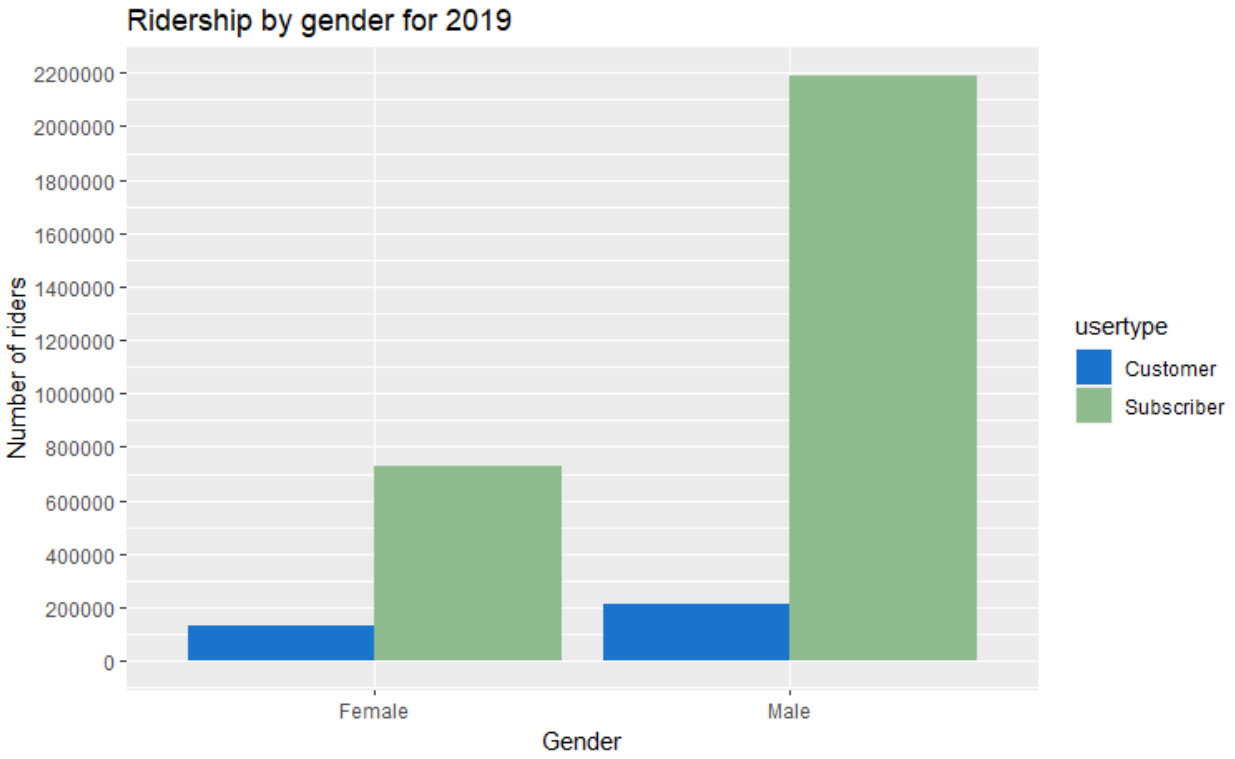
```
  726,523 members

Male riders:

  212,709 casual

  2,187,131 members
```

I then plotted the TBQ 2019 data. Although I had no data more recent than that, I assumed that the trends would be the same for 2022.



Female riders are vastly underrepresented in terms of membership. There isn't a huge difference in the number of casual riders by gender, but there is in the number of members by gender. Perhaps fewer women use bikes to commute to work? Is this because they have children to manage? Is there a way to address this?

Riders by age group

In using SQL with the TBQ data, I got an error message when I imported the "cleaned" TBQ file because some values in the "age" column didn't fit the default "tinyint" data type; it was hung up on a three-digit age. I changed the data type to "int" and it worked. But then I wrote a query to find the obviously erroneous values:

```
SELECT trip_id, age
FROM Cyclistic.dbo.TBQ_2019_cleaned
WHERE LEN(age) > 2
```

Lo and behold I got 986 rows with three-digit ages and they were values like 102, 105, 119, 123, etc., so they are obviously incorrect. Also, out of the millions of rows, there are (supposedly) five riders age 9. No other single-digit age, just 9.

```
SELECT gender, max(age) AS MaxAge, min(age) AS MinAge, avg(age) AS AvgAge
FROM Cyclistic.dbo.TBQ_2019_cleaned
WHERE LEN(age) = 2
GROUP BY (gender)
```

Setting aside the five single-digit ages, the maximum age for men is 98; the maximum age for women is 92. The minimum for both is 20, and the average is 39 for men and 37 for women. So, not much difference between genders in terms of age.

Going back to R, I deleted the rows with ages greater than 90 and less than 18, assuming that one has to be 18 to be a casual rider or member:

```
TBQ_2019_cleaned_v2 <- TBQ_2019_cleaned[TBQ_2019_cleaned$age < 90, ]
nrow(TBQ_2019_cleaned_v2[TBQ_2019_cleaned_v2$age > 90, ])
```

```
TBQ_2019_cleaned_v2 <- TBQ_2019_cleaned_v2[TBQ_2019_cleaned_v2$age >= 18,]
nrow(TBQ_2019_cleaned_v2[TBQ_2019_cleaned_v2$age < 18, ])
```

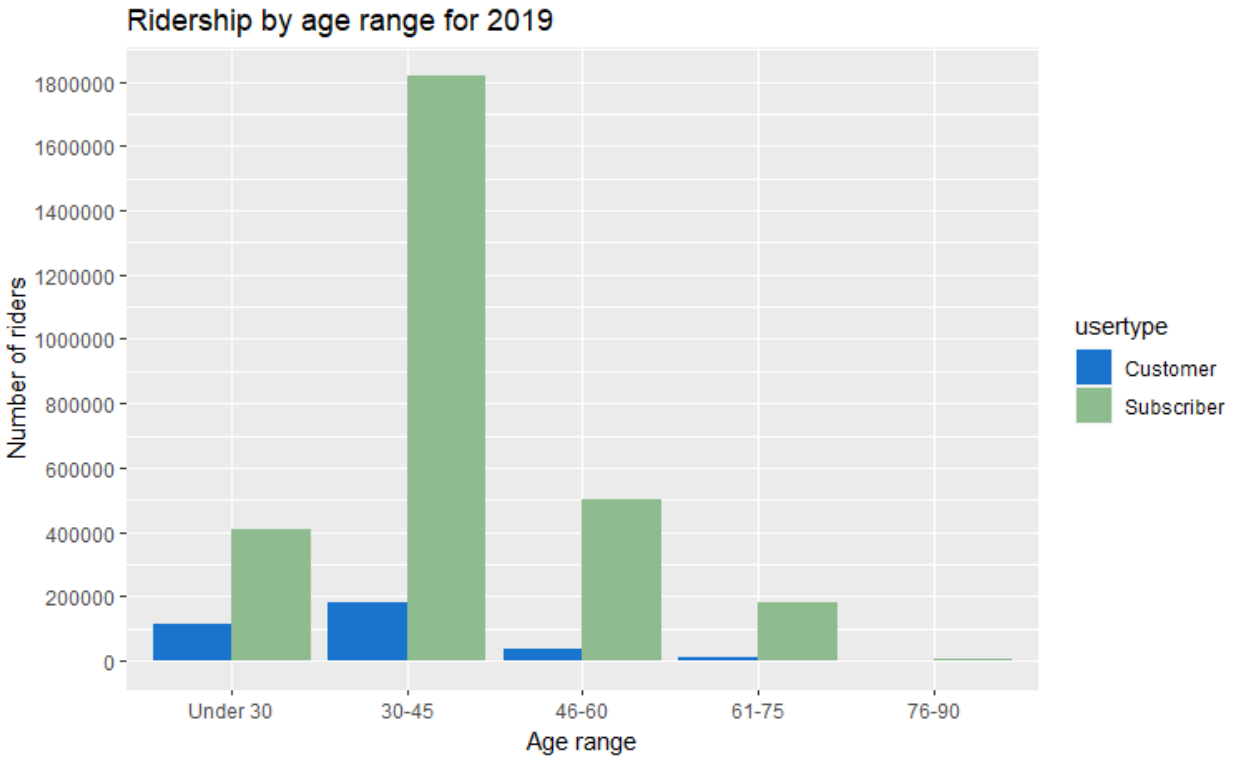
There was probably a way to combine these two statements, but I couldn't figure it out, so I just went with two. I also decided not to plot these values because the data is from 2019. I can use the numbers, perhaps, but not the plot.

I used the following code in R to create age categories:

```
TBQ_v2 <- TBQ_v2 %>%
  mutate(Age_cat = case_when(age >= 18 & age <= 29 ~ "Under 30",
                   age >= 30 & age <= 45 ~ "30-45",
                   age >= 46 & age <= 60 ~ "46-60",
                   age >= 61 & age <= 75 ~ "61-75",
                   age >= 76 & age <= 90 ~ "76-90",
                   TRUE ~ NA_character_))
```

This created a new "Age_cat" column. I then plotted the Age_cat column:

```
ggplot(TBQ_v2, aes(x = vec1)) +
  geom_bar() +
  scale_y_continuous(breaks = seq(0, 3257731, by = 200000)) +
  labs(x = "Age range", y = "Number of riders", title = "Number of riders by age range for 2019") +
  theme(legend.position="none")
```



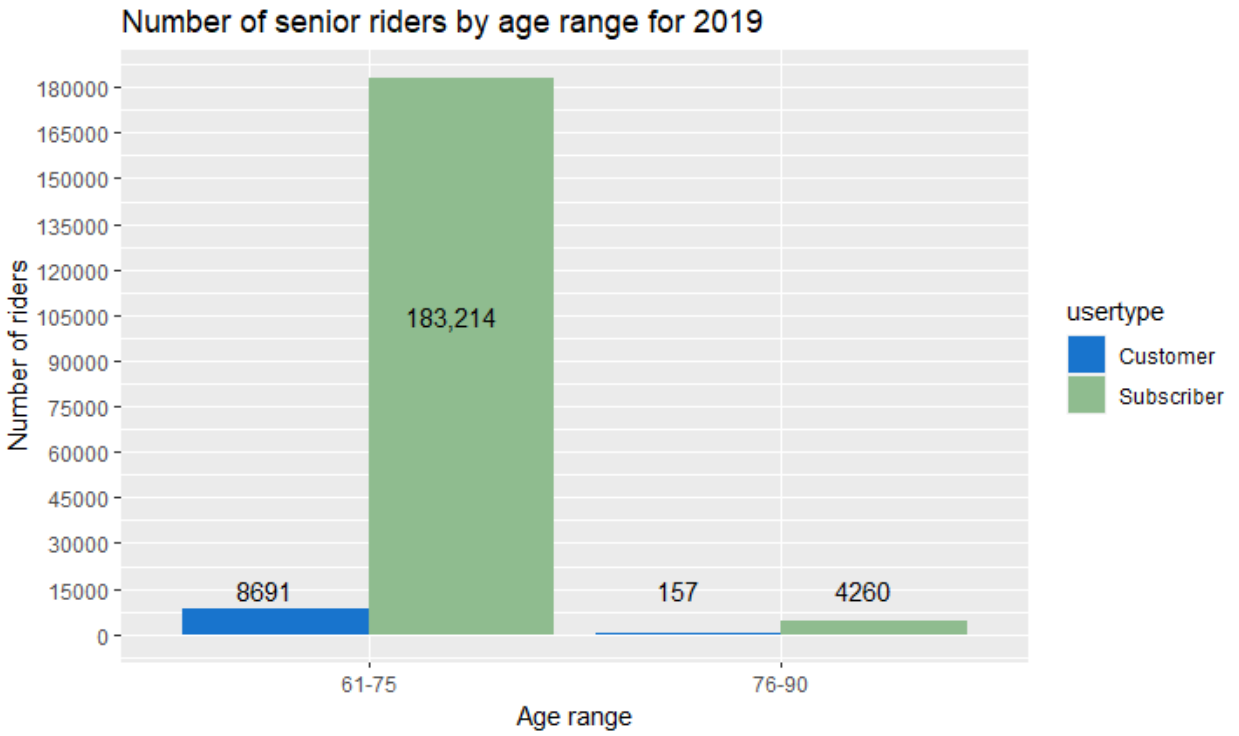
I got "NA" for the 76-90 category, I think because there were only 4417 riders in that age range. I therefore created a new dataframe with only those age 61-90.

```
TBQ_senior <- TBQ_v2 %>%
  filter(age >= 61)

level_order_v3 <- c("61-75", "76-90")

level_order_v3 <- as_factor(level_order_v3)


ggplot(data = TBQ_senior, aes(x = factor(Age_cat, levels = level_order), fill = usertype)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(breaks = seq(0, 200000, by = 15000)) +
  labs(x = "Age range", y = "Number of riders", title = "Number of senior riders by age range for
  scale_fill_manual(values=c("dodgerblue3", "darkseagreen")) +
  annotate("text", x = 0.75, y = 15000, label = "8691", color = "black") +
  annotate("text", x = 1.20, y = 105000, label = "183,214", color = "black") +
  annotate("text", x = 1.75, y = 15000, label = "157", color = "black") +
  annotate("text", x = 2.20, y = 15000, label = "4260", color = "black")
```



The values for the 76-90 age range are very low, but they are there. However, because these values don't really add anything to the analysis, i.e., they don't answer the question "What are the differences between casual riders and members?" I therefore decided to leave them out of the analysis.

It is clear from the analysis that members are primarily those in the 30-45 age range. This also applies to casual riders, although the numbers are far lower than those of members.

Ridership by time of day

Are casual riders more inclined to ride at certain times of the day compared to members? The weekday plot showed that members ride more on weekdays; is this because they commute to work, and is this borne out by an analysis of time of day?

To focus exclusively on time of day, I created a new dataframe with starting/ending_time and member_type.

```
time_of_day <- select(final_2022, 3, 4, 14)
```

I then rounded up the start/end times to the nearest hour

```
TOD$start_time <- round_date(TOD$start_time, "hour")
  TOD$end_time <- round_date(TOD$end_time, "hour")
```

and separated start time and start date

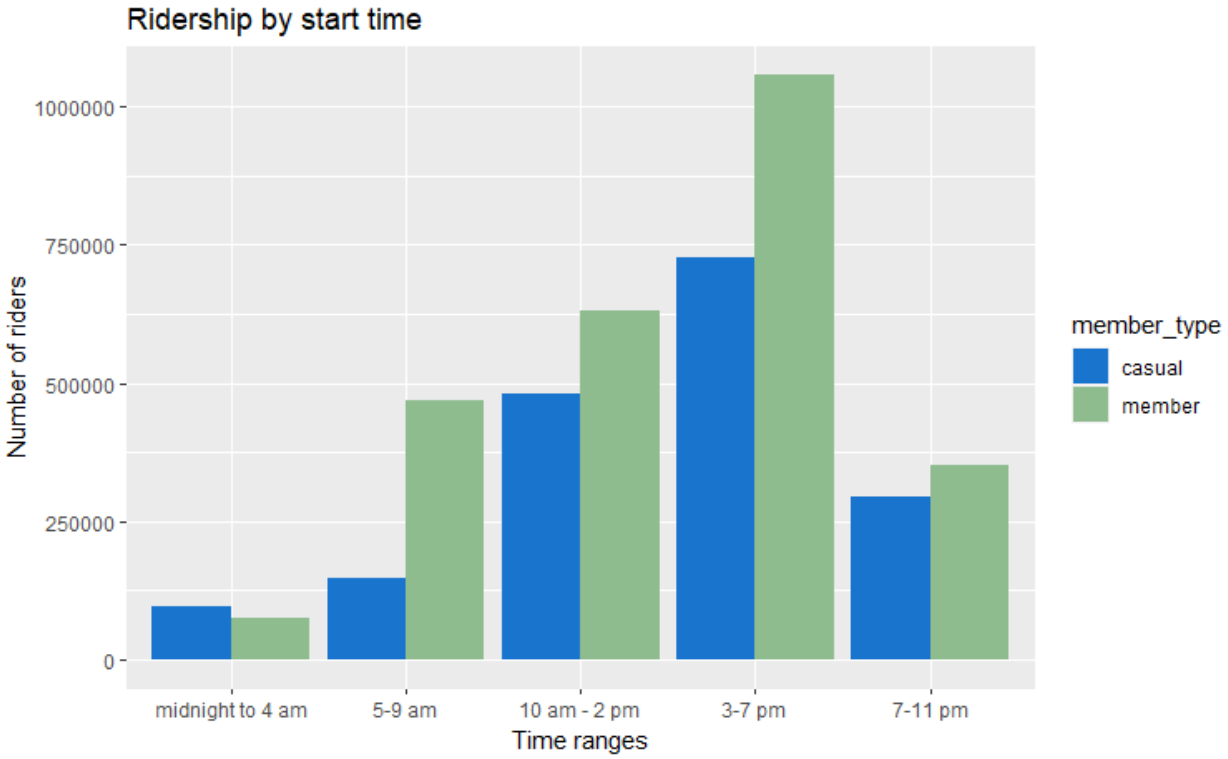
extracting time

```
  TOD$start_time <- format(as.POSIXct(
    TOD$started_at),format = "%H:%M:%S")
```

extracting date

```
TOD$start_date <- as.Date(TOD$started_at)
```

I then created time ranges and plotted the results:



The data show that members far outweigh casual riders between the hours from 5 am to 9 am and from 3 to 7 pm, which seems to corroborate the assumption that members tend to use bikes to commute to and from work. The time of greatest ridership for casual riders is 3 to 7 pm, followed by the 10 am to 2 pm time range. It's interesting to note that there are riders between midnight and 4 am; this is the only case in which casual riders outnumber members.

Final step, mapping

https://pro.arcgis.com/en/pro-app/latest/get-started/create-a-project.htm

I had created maps in ArcGIS with a small sample of data points, but I was unable to recreate them with an expanded sample. I would get either the base map with no points or points with no base map. So, I was unable to provide a definitive answer to the question regarding starting/ending locations for casual riders vs. members.

Summary of analysis

Possible action items

The analysis reveals three possible paths toward converting casual riders to annual members:

- Commuting to work
- Winter riding
- Female ridership

I therefore suggest the following actions:

Immediate implementation

1. Offer casual members discounts for weekday use
2. Offer casual members a discounted rate for membership for the first year

Marketing campaigns

1. **Commuting to work:** Develop a marketing campaign that encourages casual riders to use bikes to commute to work. Promote health benefits, benefits to environment (reduced pollution), benefits to city (reduced congestion). Emphasize accommodations the city has made (or will commit to making) for such use.
2. **Winter riding:** Develop a marketing campaign that shows different ways bikes can be used in winter, i.e., getting to local winter events, sights to see in winter, etc. As part of the campaign, provide suggestions on how to prepare for winter riding and address concerns regarding safety, i.e., icy roads, etc. Again, if the city has made or will commit to making winter riding safer, that would be a good selling point.

Other possible actions The number of female riders lags far behind that of male riders. This might be because they have additional responsibilities, i.e., children, food shopping, appointment, etc. I'm not sure how to address this, but a random survey would be the place to start. Based on the results, perhaps we could offer bike cart wagon trailers/carriers for cargo/kids. This is a vast, untapped resource that could offer additional opportunities. Tying in bike riding to reduced health insurance premiums may also be something to explore.

Mapping

I'm trying to map the ending latitude/longitude in Tableau by way of SQL but it's not working. I created new columns called "latitude" and "longitude" based on end_lat/lng, but without actual addresses, I can't seem to map them. I get latitude and longitude in Tableau but they are not "generated" latitude and longitude, and so I can't use them.