Cyclistic 2022

This R markdown Github document is published in fulfillment of the capstone project for the Google Data Analytics Professional Certification program through Coursera and the American Dream Academy.

This analysis is based on the Divvy case study written by Kevin Hartman "Sophisticated, Clear, and Polished: Divvy and Data Visualization".

The purpose of this case study is to perform real-world analysis of a fictional bike share company, Cyclistic, by following the data analysis process learned in the course - ask, prepare, process, analyze, share, and act.

I. ASK

As a junior data analyst and part of the marketing analytics team of Cyclistic, I am assigned to find out and gain insights on the problem:

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

The main business objective of this analysis is to be able to help the marketing team design a strategy to convert casual riders into annual members. Well-founded insights and recommendations, and compelling visualizations must be presented to my manager and director of marketing, Lily Moreno, and to the Cyclistic executives.

II. PREPARE

The data source I am working with is the cyclistic historical trip data made available under a license by Motivate International Inc.

The data set that I decided to work on is historical data from January 2022 to December 2022. Each individual month of data were downloaded as csv files into Microsoft Excel then sorted and filtered for the processing phase.

III. PROCESS

I have chosen to process the data for analysis using R Studio Programming. Prior to uploading into R Studio, the following steps were taken to ensure that data set is clean:

1. **Check for and unhide rows and columns.**

2. **Check for and remove duplicates.**

3. **Find and replace blanks.**

4. **Find and remove unnecessary spaces.**

A. Setting up my environment by loading the ‘tidyverse’ package and my working directory:

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library (tidyverse)
```

```
## ── Attaching packages
## ───────────────────────────────────────
## tidyverse 1.3.2 ──

## ✓ ggplot2 3.4.1     ✓ purrr   1.0.1
## ✓ tibble  3.1.8     ✓ dplyr   1.1.0
## ✓ tidyr   1.2.1     ✓ stringr 1.5.0
## ✓ readr   2.1.4     ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
getwd()
```

```
## [1] "/cloud/project/divvy_data_tsdelaney"
```

```
setwd("/cloud/project/divvy_data_tsdelaney")
```

B. Load twelve months of data by installing readr:

```
install.packages("readr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(readr)

q1_202201 <- read_csv ("202201.csv")
```

```
## Rows: 103770 Columns: 13

## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
q1_202202 <- read_csv ("202202.csv")
```

```
## Rows: 115609 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
q1_202203 <- read_csv ("202203.csv")
```

```
## Rows: 284042 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
q2_202204 <- read_csv ("202204.csv")
```

```
## Rows: 371249 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
q2_202205 <- read_csv ("202205.csv")
```

```
## Rows: 634858 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
q2_202206 <- read_csv ("202206.csv")
```

```
## Rows: 769204 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
```

```
## 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.
```

```r
q3_202207 <- read_csv ("202207.csv")
```

```
## Rows: 823488 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```r
q3_202208 <- read_csv ("202208.csv")
```

```
## Rows: 785932 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```r
q3_202209 <- read_csv ("202209.csv")
```

```
## Rows: 701339 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
q4_202210 <- read_csv ("202210.csv")
```

```
## Rows: 558685 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
q4_202211 <- read_csv ("202211.csv")
```

```
## Rows: 337735 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

```
q4_202212 <- read_csv ("202212.csv")
```

```
## Rows: 181806 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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.
```

C. Inspect the data frames and look for inconsistencies:

```
str(q1_202201)
```

```
## spc_tbl_ [103,770 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:103770] "98D355D9A9852BE9" "42178E850B92597A" "04706
##  $ rideable_type     : chr [1:103770] "classic_bike" "electric_bike" "electric_bik
##  $ started_at        : chr [1:103770] "1/1/2022 0:00" "1/1/2022 0:01" "1/1/2022 0:
##  $ ended_at          : chr [1:103770] "1/1/2022 0:01" "1/1/2022 0:32" "1/1/2022 0:
##  $ start_station_name: chr [1:103770] "Michigan Ave & 8th St" "Clark St & Ida B We
##  $ start_station_id  : chr [1:103770] "623" "TA1305000009" "13325" "623" ...
##  $ end_station_name  : chr [1:103770] "Michigan Ave & 8th St" "Clark St & Ida B We
##  $ end_station_id    : chr [1:103770] "623" "TA1305000009" "13137" "623" ...
##  $ start_lat         : num [1:103770] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:103770] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:103770] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:103770] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:103770] "casual" "casual" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q1_202202)
```

```
## spc_tbl_ [115,609 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:115609] "6325229942E058A1" "E7F30D46ACF9071D" "C9981
##  $ rideable_type     : chr [1:115609] "classic_bike" "electric_bike" "classic_bike
##  $ started_at        : chr [1:115609] "2/1/2022 0:03" "2/1/2022 0:04" "2/1/2022 0:
##  $ ended_at          : chr [1:115609] "2/1/2022 0:09" "2/1/2022 0:17" "2/1/2022 0:
##  $ start_station_name: chr [1:115609] "DuSable Lake Shore Dr & Diversey Pkwy" "Bro
##  $ start_station_id  : chr [1:115609] "TA1309000039" "13109" "13008" "13008" ...
##  $ end_station_name  : chr [1:115609] "Clark St & Wellington Ave" "Western Ave & L
##  $ end_station_id    : chr [1:115609] "TA1307000136" "TA1307000140" "623" "623" ..
##  $ start_lat         : num [1:115609] 41.9 42 41.9 41.9 41.8 ...
##  $ start_lng         : num [1:115609] -87.6 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:115609] 41.9 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:115609] -87.6 -87.7 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:115609] "casual" "casual" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q1_202203)
```

```
## spc_tbl_ [284,042 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:284042] "41557457145715FC" "2CF34B94DEDAF6D1" "ED3DD
##  $ rideable_type     : chr [1:284042] "classic_bike" "electric_bike" "classic_bike
##  $ started_at        : chr [1:284042] "3/1/2022 0:00" "3/1/2022 0:02" "3/1/2022 0:
##  $ ended_at          : chr [1:284042] "3/1/2022 0:04" "3/1/2022 0:08" "3/1/2022 0:
##  $ start_station_name: chr [1:284042] "Wentworth Ave & Cermak Rd" "State St & Pear
##  $ start_station_id  : chr [1:284042] "13075" "TA1307000061" "KA1504000143" NA ...
##  $ end_station_name  : chr [1:284042] "Normal Ave & Archer Ave" "Ogden Ave & Chica
##  $ end_station_id    : chr [1:284042] "TA1308000014" "TA1305000020" "TA1307000006"
##  $ start_lat         : num [1:284042] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:284042] -87.6 -87.6 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:284042] 41.8 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:284042] -87.6 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:284042] "member" "member" "casual" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q2_202204)
```

```
## spc_tbl_ [371,249 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:371249] "6DFBE82AEF4187C0" "AFDC60E4BD0755EB" "B0AD9
##  $ rideable_type     : chr [1:371249] "electric_bike" "electric_bike" "electric_bi
##  $ started_at        : chr [1:371249] "4/1/2022 0:01" "4/1/2022 0:01" "4/1/2022 0:
##  $ ended_at          : chr [1:371249] "4/1/2022 0:02" "4/1/2022 0:07" "4/1/2022 0:
##  $ start_station_name: chr [1:371249] "Kedzie Ave & 48th Pl" "Base - 2132 W Hubbar
##  $ start_station_id  : chr [1:371249] "382" "HubbardBike-checking(LBS-WH-TEST)" "T
##  $ end_station_name  : chr [1:371249] "Kedzie Ave & 48th Pl" NA "Lakeview Ave & Fu
##  $ end_station_id    : chr [1:371249] "382" NA "TA1309000019" NA ...
##  $ start_lat         : num [1:371249] 41.8 41.9 41.9 41.8 42 ...
##  $ start_lng         : num [1:371249] -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num [1:371249] 41.8 41.9 41.9 41.8 42 ...
##  $ end_lng           : num [1:371249] -87.7 -87.6 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr [1:371249] "casual" "member" "casual" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q2_202205)
```

```
## spc_tbl_ [634,858 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:634858] "2ADEBDB639C80F16" "F6A29D3852FE4B05" "E7EB5
##  $ rideable_type     : chr [1:634858] "classic_bike" "classic_bike" "classic_bike"
##  $ started_at        : chr [1:634858] "5/1/2022 0:00" "5/1/2022 0:00" "5/1/2022 0:
##  $ ended_at          : chr [1:634858] "5/1/2022 0:12" "5/1/2022 0:15" "5/1/2022 0:
##  $ start_station_name: chr [1:634858] "Sheffield Ave & Webster Ave" "Federal St &
##  $ start_station_id  : chr [1:634858] "TA1309000033" "SL-008" "TA1309000033" "TA13
##  $ end_station_name  : chr [1:634858] "Southport Ave & Belmont Ave" "Desplaines St
##  $ end_station_id    : chr [1:634858] "13229" "15535" "13229" "13247" ...
##  $ start_lat         : num [1:634858] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:634858] -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:634858] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:634858] -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:634858] "casual" "casual" "casual" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q2_202206)
```

```
## spc_tbl_ [769,204 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:769204] "C10099D850C7FC17" "75FA47BE49CA9CE1" "D6750
##  $ rideable_type     : chr [1:769204] "classic_bike" "electric_bike" "electric_bik
##  $ started_at        : chr [1:769204] "6/1/2022 0:00" "6/1/2022 0:00" "6/1/2022 0:
##  $ ended_at          : chr [1:769204] "6/1/2022 0:02" "6/1/2022 0:19" "6/1/2022 0:
##  $ start_station_name: chr [1:769204] "Broadway & Argyle St" "Elston Ave & Wabansi
##  $ start_station_id  : chr [1:769204] "13108" "TA1309000032" "KA1504000135" NA ...
##  $ end_station_name  : chr [1:769204] "Broadway & Berwyn Ave" NA "Wells St & Everg
##  $ end_station_id    : chr [1:769204] "13109" NA "TA1308000049" NA ...
##  $ start_lat         : num [1:769204] 42 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:769204] -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num [1:769204] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:769204] -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr [1:769204] "member" "member" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q3_202207)
```

```
## spc_tbl_ [823,488 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:823488] "56C6CCD4EE89184D" "7EA7A3AEAAB5F621" "FDBFB
##  $ rideable_type     : chr [1:823488] "classic_bike" "classic_bike" "electric_bike
##  $ started_at        : chr [1:823488] "7/1/2022 0:00" "7/1/2022 0:00" "7/1/2022 0:
##  $ ended_at          : chr [1:823488] "7/1/2022 0:20" "7/1/2022 0:11" "7/1/2022 0:
##  $ start_station_name: chr [1:823488] "Southport Ave & Roscoe St" "Sheffield Ave &
##  $ start_station_id  : chr [1:823488] "13071" "TA1307000052" "TA1306000032" "TA130
##  $ end_station_name  : chr [1:823488] "Ravenswood Ave & Lawrence Ave" "Sheffield A
##  $ end_station_id    : chr [1:823488] "TA1309000066" "TA1307000126" "TA1307000143"
##  $ start_lat         : num [1:823488] 41.9 41.9 41.9 41.9 42 ...
##  $ start_lng         : num [1:823488] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:823488] 42 41.9 41.9 41.9 42 ...
##  $ end_lng           : num [1:823488] -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr [1:823488] "casual" "casual" "casual" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q3_202208)
```

```
## spc_tbl_ [785,932 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:785932] "8EFB69C0CAB8779D" "CD6A225BE301AB8F" "FB9EC
##  $ rideable_type     : chr [1:785932] "electric_bike" "electric_bike" "classic_bik
##  $ started_at        : chr [1:785932] "8/1/2022 0:00" "8/1/2022 0:00" "8/1/2022 0:
##  $ ended_at          : chr [1:785932] "8/1/2022 0:08" "8/1/2022 0:08" "8/1/2022 0:
##  $ start_station_name: chr [1:785932] "Michigan Ave & Washington St" "Streeter Dr
##  $ start_station_id  : chr [1:785932] "13001" "13022" "KA1503000015" NA ...
##  $ end_station_name  : chr [1:785932] "Wabash Ave & 9th St" "Wabash Ave & Adams St
##  $ end_station_id    : chr [1:785932] "TA1309000010" "KA1503000015" "13409" NA ...
##  $ start_lat         : num [1:785932] 41.9 41.9 41.9 42 42 ...
##  $ start_lng         : num [1:785932] -87.6 -87.6 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num [1:785932] 41.9 41.9 41.9 42 42 ...
##  $ end_lng           : num [1:785932] -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:785932] "member" "casual" "member" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q3_202209)
```

```
## spc_tbl_ [701,339 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:701339] "A4BBE90F834C0422" "51DC98F92D41C0CD" "CD46A
##  $ rideable_type     : chr [1:701339] "classic_bike" "electric_bike" "classic_bike
##  $ started_at        : chr [1:701339] "9/1/2022 0:00" "9/1/2022 0:00" "9/1/2022 0:
##  $ ended_at          : chr [1:701339] "9/1/2022 1:07" "9/1/2022 0:18" "9/1/2022 0:
##  $ start_station_name: chr [1:701339] "Lincoln Ave & Sunnyside Ave" "Clark St & Li
##  $ start_station_id  : chr [1:701339] "TA1307000156" "13179" "13332" "13135" ...
##  $ end_station_name  : chr [1:701339] "Manor Ave & Leland Ave" "Lincoln Ave & Bell
##  $ end_station_id    : chr [1:701339] "KA1504000127" "TA1309000026" "TA1307000130"
##  $ start_lat         : num [1:701339] 42 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:701339] -87.7 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:701339] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:701339] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:701339] "member" "casual" "member" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q4_202210)
```

```
## spc_tbl_ [558,685 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:558685] "8BD7CD28EB78C3F0" "D6A773C844DE6EEE" "DCBCC
##  $ rideable_type     : chr [1:558685] "electric_bike" "electric_bike" "electric_bi
##  $ started_at        : chr [1:558685] "10/1/2022 0:00" "10/1/2022 0:00" "10/1/2022
##  $ ended_at          : chr [1:558685] "10/1/2022 0:06" "10/1/2022 0:10" "10/1/2022
##  $ start_station_name: chr [1:558685] "Racine Ave & Congress Pkwy" NA NA "Sheffiel
##  $ start_station_id  : chr [1:558685] "TA1306000025" NA NA "TA1309000023" ...
##  $ end_station_name  : chr [1:558685] "Wolcott Ave & Polk St" "Southport Ave & Ros
##  $ end_station_id    : chr [1:558685] "TA1309000064" "13071" NA "RN-" ...
##  $ start_lat         : num [1:558685] 41.9 41.9 42 41.9 42 ...
##  $ start_lng         : num [1:558685] -87.7 -87.7 -87.8 -87.7 -87.7 ...
##  $ end_lat           : num [1:558685] 41.9 41.9 42 41.9 42 ...
##  $ end_lng           : num [1:558685] -87.7 -87.7 -87.8 -87.6 -87.7 ...
##  $ member_casual     : chr [1:558685] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q4_202211)
```

```
## spc_tbl_ [337,735 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:337735] "5446FCEB95D32460" "88005CF56808334A" "AF7B0
##  $ rideable_type     : chr [1:337735] "classic_bike" "electric_bike" "electric_bik
##  $ started_at        : chr [1:337735] "11/1/2022 0:00" "11/1/2022 0:00" "11/1/2022
##  $ ended_at          : chr [1:337735] "11/1/2022 0:13" "11/1/2022 0:18" "11/1/2022
##  $ start_station_name: chr [1:337735] "Halsted St & Wrightwood Ave" NA "California
##  $ start_station_id  : chr [1:337735] "TA1309000061" NA "17660" "TA1307000064" ...
##  $ end_station_name  : chr [1:337735] "Clark St & North Ave" NA NA "Clifton Ave &
##  $ end_station_id    : chr [1:337735] "13128" NA NA "TA1307000163" ...
##  $ start_lat         : num [1:337735] 41.9 42 41.9 41.9 42 ...
##  $ start_lng         : num [1:337735] -87.6 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:337735] 41.9 42 41.9 41.9 42 ...
##  $ end_lng           : num [1:337735] -87.6 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:337735] "casual" "member" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

```
str(q4_202212)
```

```
## spc_tbl_ [181,806 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:181806] "C2728784FCDB3735" "8FE02CD119032644" "918C1
##  $ rideable_type     : chr [1:181806] "electric_bike" "electric_bike" "electric_bi
##  $ started_at        : chr [1:181806] "12/1/2022 0:01" "12/1/2022 0:01" "12/1/2022
##  $ ended_at          : chr [1:181806] "12/1/2022 0:03" "12/1/2022 0:14" "12/1/2022
##  $ start_station_name: chr [1:181806] "Greenview Ave & Fullerton Ave" "Wood St & T
##  $ start_station_id  : chr [1:181806] "TA1307000001" "13285" NA NA ...
##  $ end_station_name  : chr [1:181806] "Racine Ave & Fullerton Ave (Temp)" "Orleans
##  $ end_station_id    : chr [1:181806] "TA1306000026" "TA1305000022" "SL-008" NA ..
##  $ start_lat         : num [1:181806] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:181806] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:181806] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:181806] -87.7 -87.6 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr [1:181806] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

D. Create one big data frame by stacking individual data frames:

```
library(dplyr)

divvy_trips <- bind_rows (q1_202201, q1_202202, q1_202203, q2_202204, q2_202205, q2_2
```

E. Remove unnecessary columns (latitude columns, longitude columns):

```
divvy_trips <- divvy_trips %>%
select (-c (start_lat, start_lng, end_lat, end_lng))
```

F. Reassign the member_column to the the desired values:

```
divvy_trips <- divvy_trips %>%
mutate (member_casual = recode (member_casual, "member" = "annual_members"
,"casual" = "casual_riders"))
```

G. Check to make sure the proper number of observations were reassigned:

```
num_riders <- table(divvy_trips$member_casual)
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

H. Before adding columns that list the date, month, day, and year of each ride, I have to convert date and time to POSIXlt class:

```
divvy_trips$started_at <- strptime (divvy_trips$started_at, format = '%m/%d/%Y %H:%M'
divvy_trips$ended_at <- strptime (divvy_trips$ended_at, format = '%m/%d/%Y %H:%M')
```

I. Add columns that list the date, month, day, and year of each ride:

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

J. Add a “ride_length” calculation to divvy_trips (in seconds) which involves ‘difftime’ calculation:

```
divvy_trips$ride_length <- difftime(divvy_trips$ended_at,divvy_trips$started_at)
```

K. Convert “ride_length” from factor to numeric so we can run calculations on the data:

```
is.factor(divvy_trips$ride_length)
```

```
## [1] FALSE
```

```
divvy_trips$ride_length <- as.numeric(as.character(divvy_trips$ride_length))
is.numeric(divvy_trips$ride_length)
```

```
## [1] TRUE
```

L. Remove “bad” data (e.g. ride_length was negative):

```
divvy_trips_v2 <- divvy_trips [!(divvy_trips$ride_length<0),]
```

IV. ANALYZE

A. Descriptive analysis on ride_length (all figures in seconds):

```
mean(divvy_trips_v2$ride_length)
```

```
## [1] 1166.774
```

```
median(divvy_trips_v2$ride_length)
```

```
## [1] 600
```

```
max(divvy_trips_v2$ride_length)
```

```
## [1] 2483220
```

```
min(divvy_trips_v2$ride_length)
```

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

The four lines above may be condensed to one line using summary() on the specific attribute:

```
summary(divvy_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0     360     600    1167    1080 2483220
```

B. The data is aggregated to compare members and casual users:

```
aggregate(divvy_trips_v2$ride_length ~ divvy_trips_v2$member_casual, FUN = mean)
```

```
##   divvy_trips_v2$member_casual divvy_trips_v2$ride_length
## 1               annual_members                   762.8505
## 2                casual_riders                  1748.7691
```

```
aggregate(divvy_trips_v2$ride_length ~ divvy_trips_v2$member_casual, FUN = median)
```

```
##   divvy_trips_v2$member_casual divvy_trips_v2$ride_length
## 1               annual_members                        540
## 2                casual_riders                        780
```

```
aggregate(divvy_trips_v2$ride_length ~ divvy_trips_v2$member_casual, FUN = max)
```

```
##   divvy_trips_v2$member_casual divvy_trips_v2$ride_length
## 1               annual_members                      93600
## 2                casual_riders                    2483220
```

```
aggregate(divvy_trips_v2$ride_length ~ divvy_trips_v2$member_casual, FUN = min)
```

```
##   divvy_trips_v2$member_casual divvy_trips_v2$ride_length
## 1               annual_members                          0
## 2                casual_riders                          0
```

C. To see the average ride time by each day for annual members and casual riders:

```
aggregate(divvy_trips_v2$ride_length ~ divvy_trips_v2$member_casual +
divvy_trips_v2$day_of_week, FUN = mean)
```

```
##    divvy_trips_v2$member_casual divvy_trips_v2$day_of_week
## 1                annual_members                     Friday
## 2                 casual_riders                     Friday
## 3                annual_members                     Monday
## 4                 casual_riders                     Monday
## 5                annual_members                   Saturday
## 6                 casual_riders                   Saturday
## 7                annual_members                     Sunday
## 8                 casual_riders                     Sunday
## 9                annual_members                   Thursday
## 10                casual_riders                   Thursday
## 11               annual_members                    Tuesday
## 12                casual_riders                    Tuesday
## 13               annual_members                  Wednesday
## 14                casual_riders                  Wednesday
##    divvy_trips_v2$ride_length
## 1                    751.9322
## 2                   1682.6268
## 3                    736.1577
## 4                   1751.2321
## 5                    848.3786
## 6                   1956.8904
## 7                    841.8464
## 8                   2043.4805
## 9                    737.5799
## 10                  1532.9079
## 11                   727.7927
## 12                  1549.4268
## 13                   726.3266
## 14                  1485.1390
```

D. To fix the order of the days of the week:

```
divvy_trips_v2$day_of_week <- ordered(divvy_trips_v2$day_of_week,
levels=c("Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"))
```

E. To find out the average ride time by each day for annual members vs casual users:

```
aggregate(divvy_trips_v2$ride_length ~ divvy_trips_v2$member_casual+
divvy_trips_v2$day_of_week, FUN = mean)
```

```
##    divvy_trips_v2$member_casual divvy_trips_v2$day_of_week
## 1                annual_members                     Sunday
## 2                 casual_riders                     Sunday
## 3                annual_members                     Monday
## 4                 casual_riders                     Monday
## 5                annual_members                    Tuesday
## 6                 casual_riders                    Tuesday
## 7                annual_members                  Wednesday
## 8                 casual_riders                  Wednesday
## 9                annual_members                   Thursday
## 10               annual_members                   Thursday
## 11               annual_members                     Friday
## 12                casual_riders                     Friday
## 13               annual_members                   Saturday
## 14                casual_riders                   Saturday
##    divvy_trips_v2$ride_length
## 1                    841.8464
## 2                   2043.4805
## 3                    736.1577
## 4                   1751.2321
## 5                    727.7927
## 6                   1549.4268
## 7                    726.3266
## 8                   1485.1390
## 9                    737.5799
## 10                  1532.9079
## 11                   751.9322
## 12                  1682.6268
## 13                   848.3786
## 14                  1956.8904
```

V. SHARE

My preference is to disable scientific notation in plots so prior to installing ggplot2, I run this code:

```
options(scipen = 999)
```

```
install.packages("ggplot2")
```

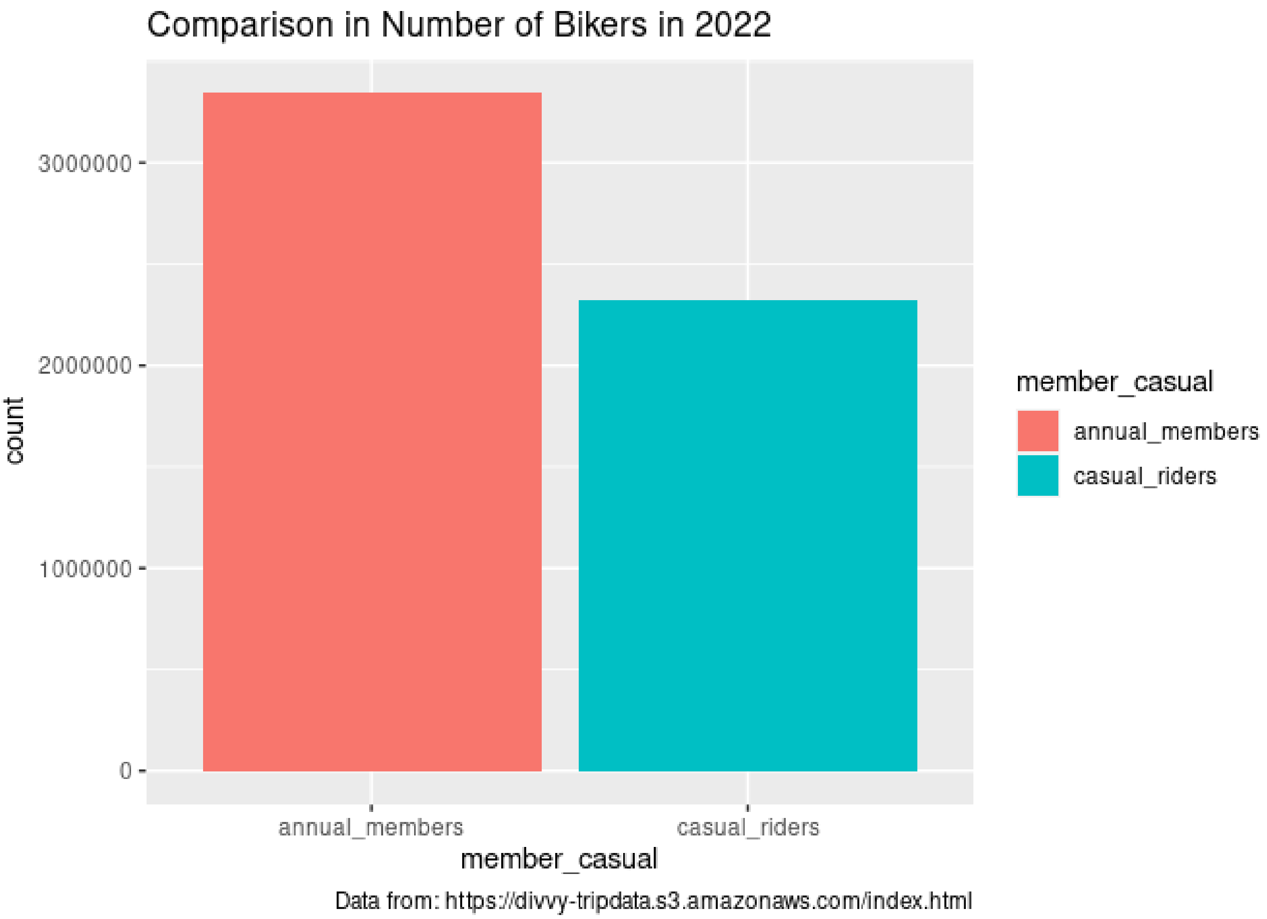
```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(ggplot2)
```

A. To visualize the number of annual members and the number of casual riders in 2022:

```
riders_2022 <- divvy_trips_v2 %>%
  select (member_casual)

ggplot (data = riders_2022)+
  geom_bar (mapping = aes (x = member_casual, fill = member_casual))+
  labs (title = "Comparison in Number of Bikers in 2022",caption = paste0("Data from:
```



Data from: https://divvy-tripdata.s3.amazonaws.com/index.html

B. Next, I thought of making a comparison in the number of riders per month and create a Tableau visualization called Monthly Rider

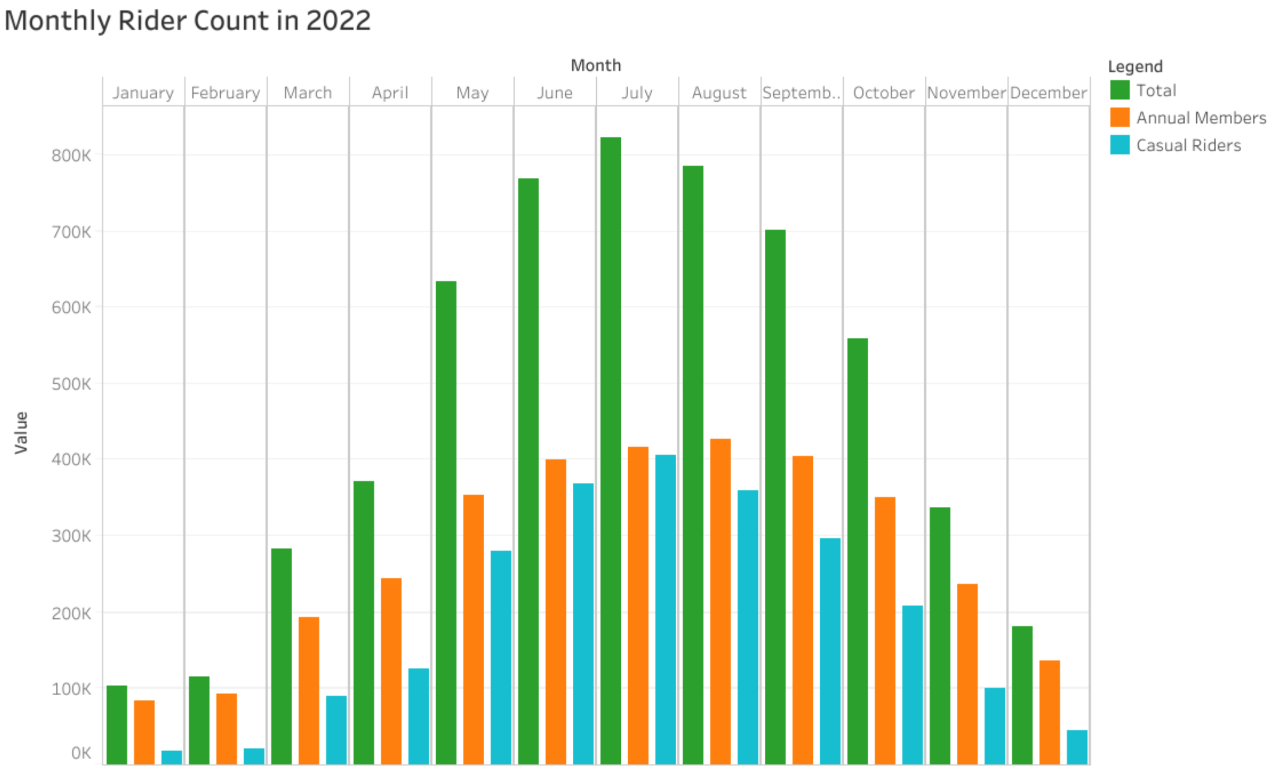
Count in 2022. In order to create the visualization in Tableau, I exported the divvy_trips_v2 as csv file form R Studio to Microsoft Excel and then Tableau:

```
write.csv(divvy_trips_v2, file = "C:\\Users\\Thessa Delaney\\Desktop\\divvy_trips_Rfi
```

```
monthly_2022_rider_count <- read_csv("monthly_rider_count_2022csv.csv")
```

```
## Rows: 12 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): month
## dbl (3): annual_members, casual_riders, total
##
## 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.
```

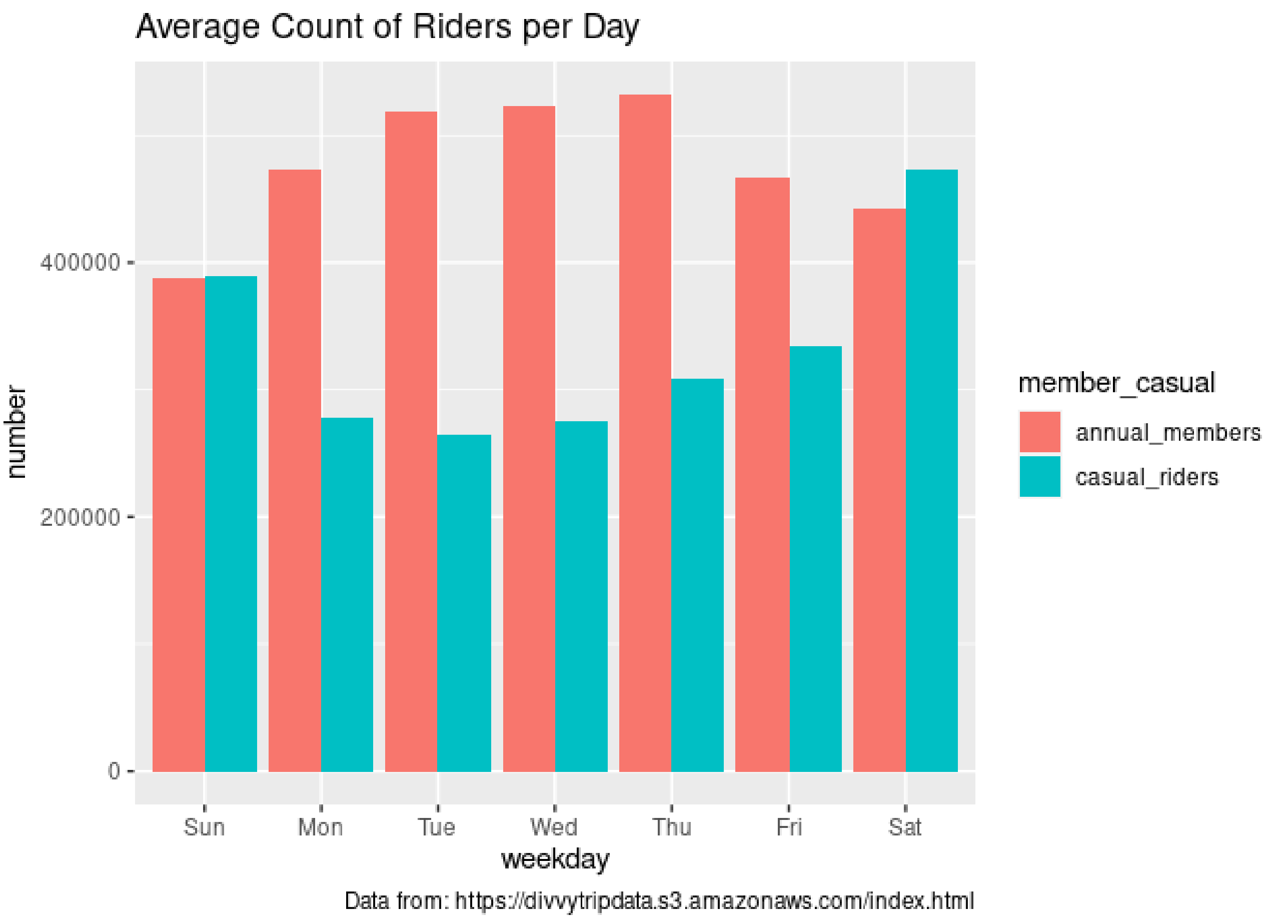
```
knitr::include_graphics("monthly_rider_count_2022tbl.png")
```



C. To visualize the number of rides by rider type:

```
divvy_trips_v2 %>%
mutate(weekday = wday(started_at, label = TRUE)) %>%
group_by(member_casual, weekday) %>%
summarise(number = n()
,average_duration = mean(ride_length)) %>%
arrange(member_casual,weekday) %>%
ggplot(aes(x = weekday, y = number, fill = member_casual)) +
geom_col(position = "dodge")+
labs (title = "Average Count of Riders per Day",caption = paste0("Data from: https://
```

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



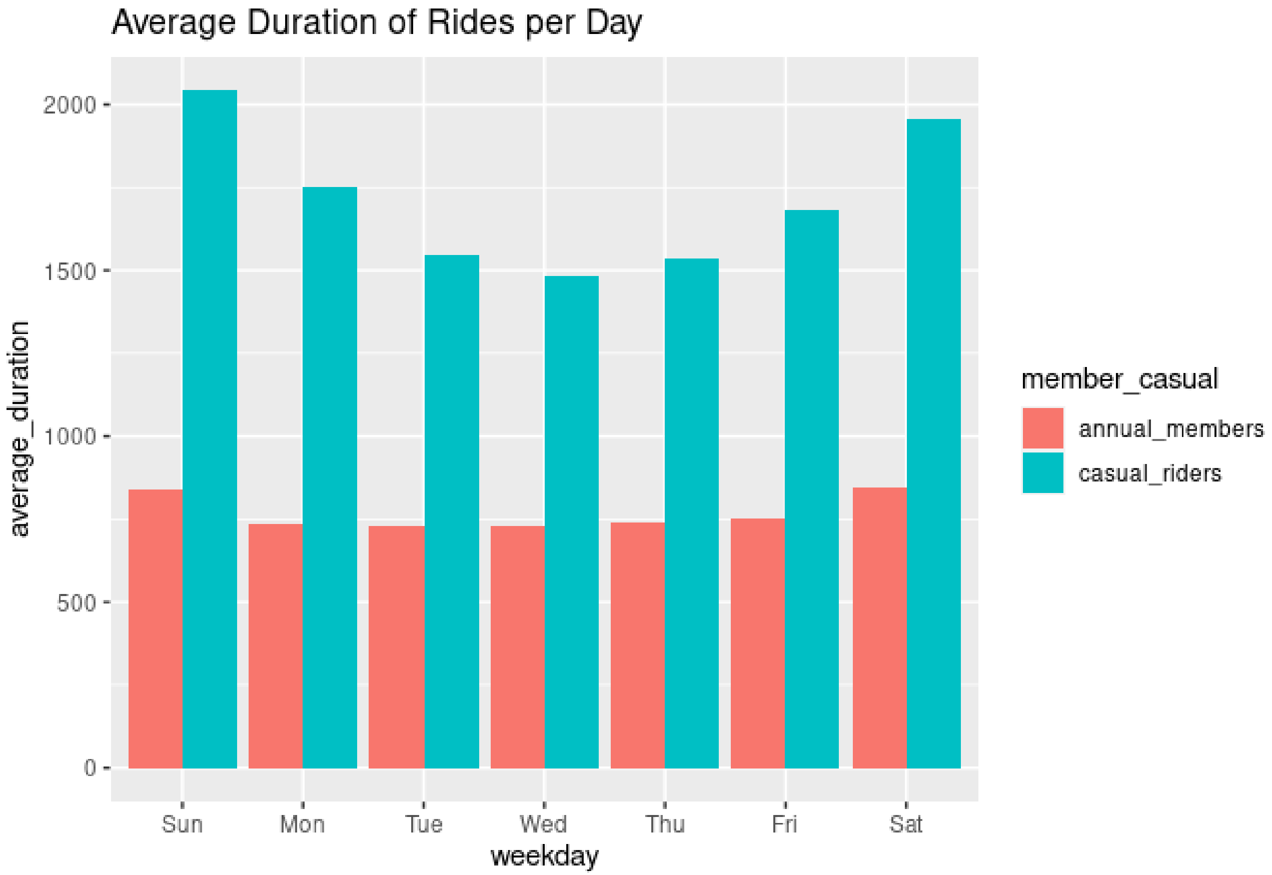
Average Count of Riders per Day

Data from: https://divvytripdata.s3.amazonaws.com/index.html

D. To visualize the average duration per day:

```
divvy_trips_v2 %>%
mutate(weekday = wday(started_at, label = TRUE)) %>%
group_by(member_casual, weekday) %>%
summarise(number_of_rides = n()
,average_duration = mean(ride_length)) %>%
arrange(member_casual,weekday) %>%
ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
geom_col(position = "dodge")+
labs (title = "Average Duration of Rides per Day",caption = paste0("Data from: https:
```

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

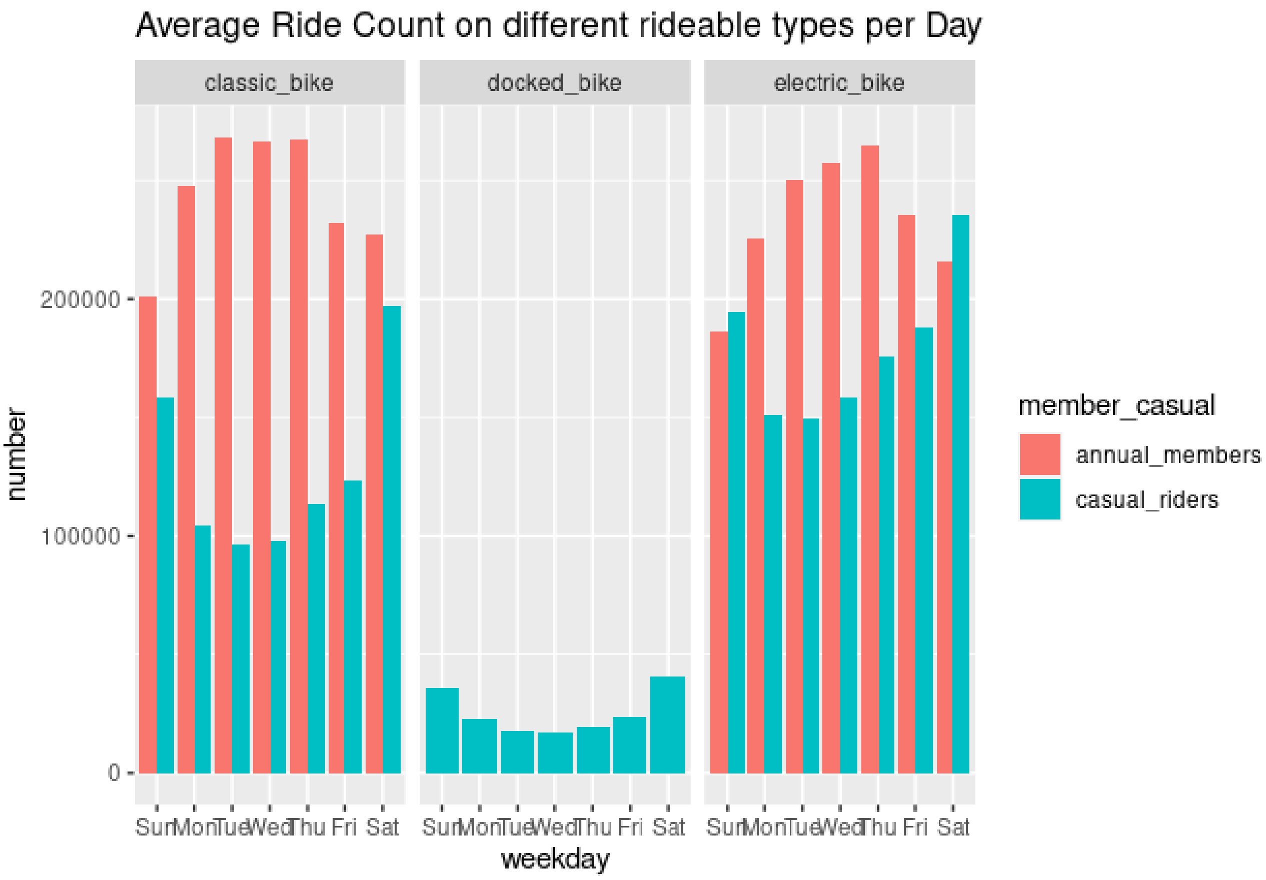

Average Duration of Rides per Day

Data from: https://divvytripdata.s3.amazonaws.com/index.html

E. To visualize average ride count by on different rideable types per day:

```
divvy_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(rideable_type, weekday, member_casual) %>%
  summarise(number = n()
          ,average_duration = mean(ride_length)) %>%
  arrange(rideable_type,weekday, rideable_type) %>%
  ggplot(aes(x = weekday, y = number, fill = member_casual)) +
  facet_wrap (~ rideable_type)+
  geom_col(position = "dodge")+
  labs (title = "Average Ride Count on different rideable types per Day",caption = pa
```

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


Average Ride Count on different rideable types per Day

Data from: https://divvy-tripdata.s3.amazonaws.com/index.html

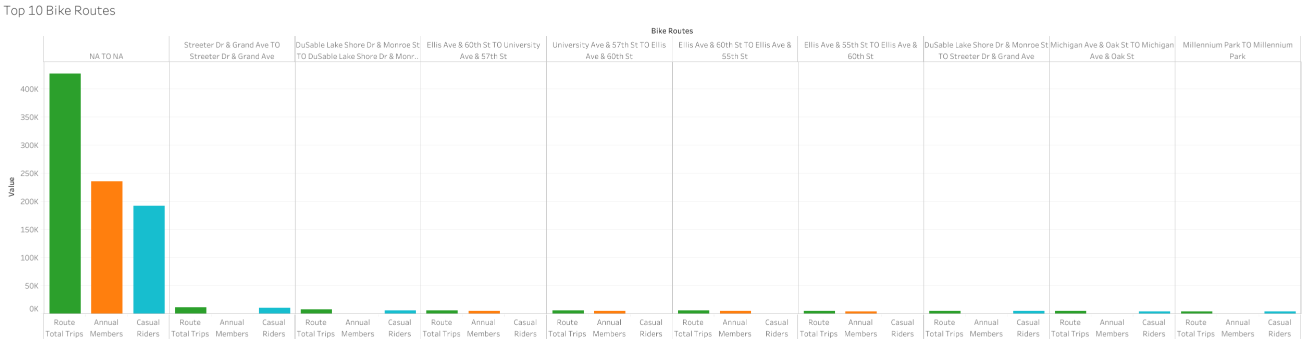
F. To visualize the top 10 routes using Tableau:

```
top10_bike_routes <- read_csv("tbl_top10_routes.csv")
```

```
## Rows: 10 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): Bike Routes
## dbl (3): Annual Members, Casual Riders, Route Total Trips
##
```

```
## 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.
```

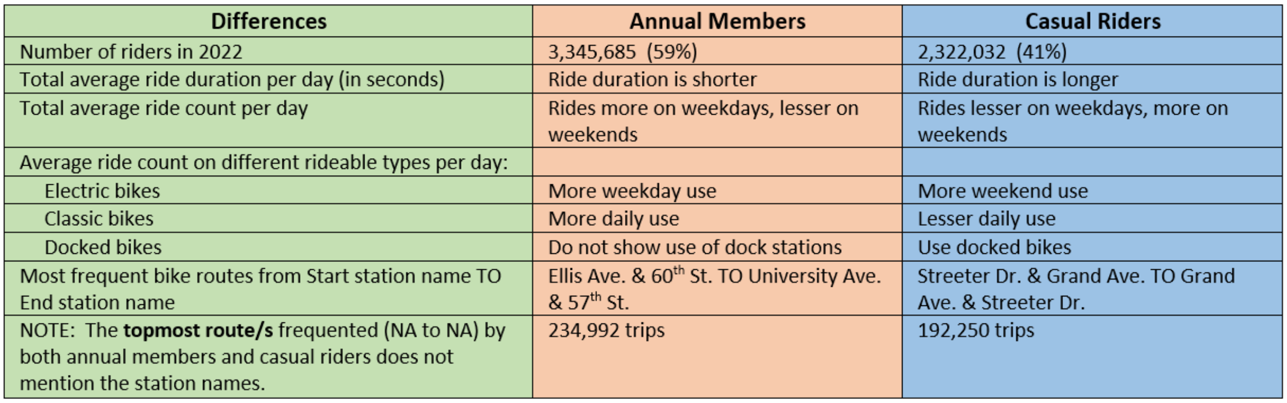
```
knitr::include_graphics("top10_routes.png")
```



VI. ACT

A. Differences between Annual Members and Casual Riders:

```
knitr::include_graphics("differences_bw_member_casual.png")
```



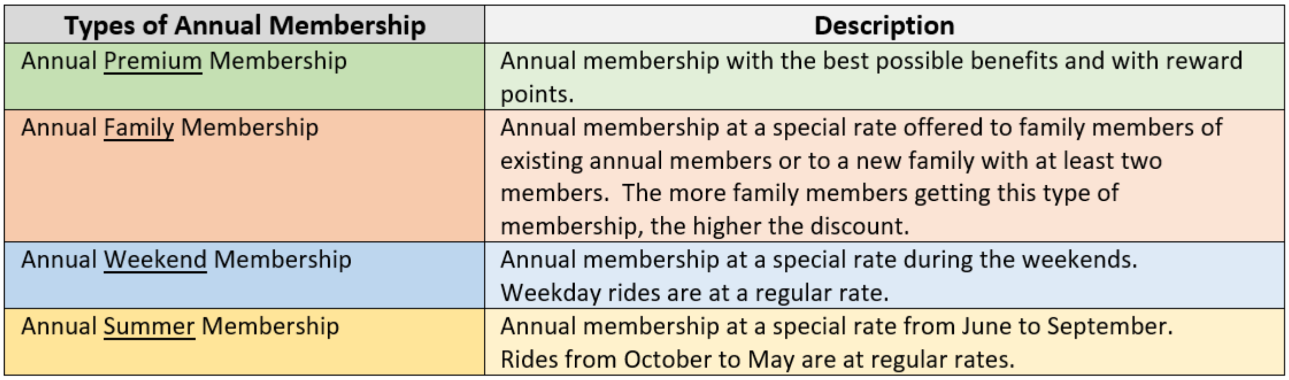
Differences	Annual Members	Casual Riders
Number of riders in 2022	3,345,685 (59%)	2,322,032 (41%)
Total average ride duration per day (in seconds)	Ride duration is shorter	Ride duration is longer
Total average ride count per day	Rides more on weekdays, lesser on weekends	Rides lesser on weekdays, more on weekends
Average ride count on different rideable types per day:		
Electric bikes	More weekday use	More weekend use
Classic bikes	More daily use	Lesser daily use
Docked bikes	Do not show use of dock stations	Use docked bikes
Most frequent bike routes from Start station name TO End station name	Ellis Ave. & 60th St. TO University Ave. & 57th St.	Streeter Dr. & Grand Ave. TO Grand Ave. & Streeter Dr.
NOTE: The **topmost route/s** frequented (NA to NA) by both annual members and casual riders does not mention the station names.	234,992 trips	192,250 trips

Data also shows a huge increase in the number of riders during the summer months and a decline from October to May.

B. Based on my analysis, here are my recommendations:

1. Develop marketing programs to offer different types of annual memberships:

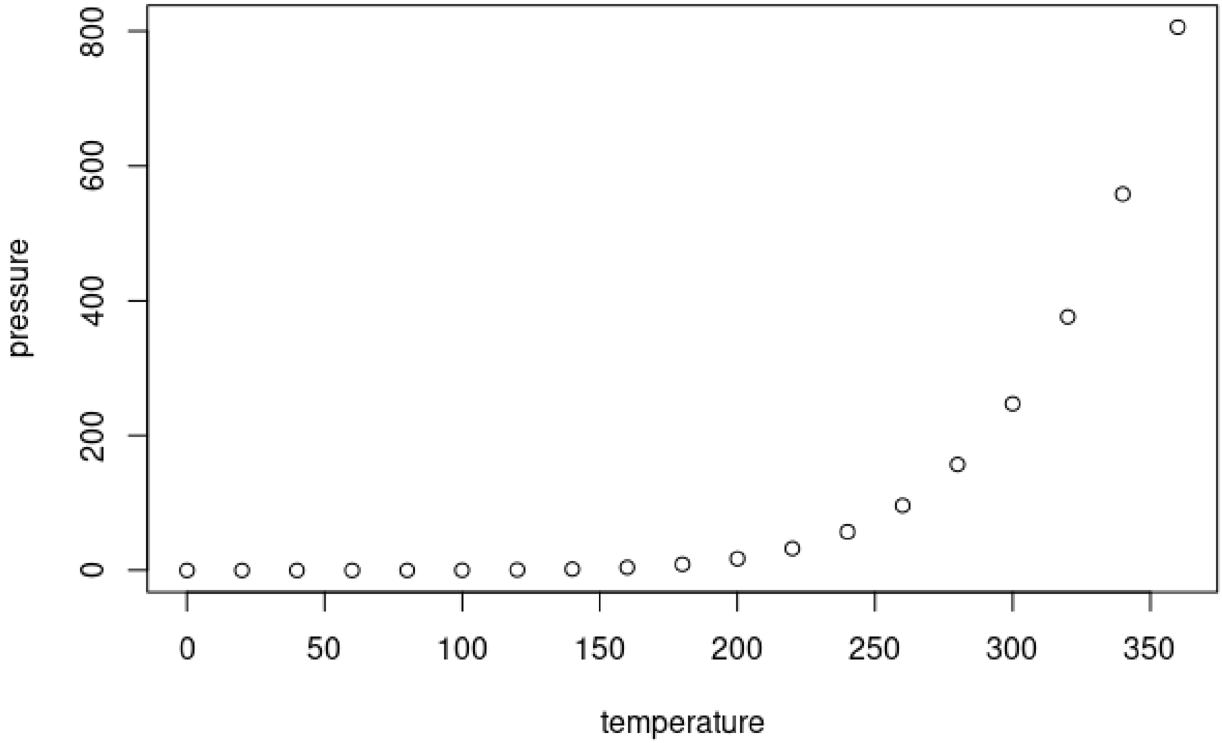
```
knitr::include_graphics("annual_membership_types.png")
```



Types of Annual Membership	Description
Annual Premium Membership	Annual membership with the best possible benefits and with reward points.
Annual Family Membership	Annual membership at a special rate offered to family members of existing annual members or to a new family with at least two members. The more family members getting this type of membership, the higher the discount.
Annual Weekend Membership	Annual membership at a special rate during the weekends. Weekday rides are at a regular rate.
Annual Summer Membership	Annual membership at a special rate from June to September. Rides from October to May are at regular rates.

2. As there is a noticeable increase in the number of riders during the summer months for both annual members and casual riders, establish a program to offer hotel establishments annual memberships at a special discount. These memberships will be owned and paid for by the hotel for use by hotel guests who would like to use *Cyclistic* bikes but are not willing to subscribe annually as they are only in the area during summer months for a brief stay.

3. Get data on the individual annual member riders and casual riders frequency, duration of rides and spend for comparison purposes. Comparing the annual members to casual riders will provide insight into the higher spend or more profitable riders. Understanding how frequent they ride, the time and duration of their ride, and the amount they spend over a set period will allow identification of casual riders to convert to annual members.



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.