Cyclist project

Scott Schisler

2025-05-14

Project explanation:

This is a project for the Google data analytics certificate. I am using R studio to explore and analyze the data from Motivate International Inc under their data license agreement link. In this project I have been tasked with answering the following question: How do annual members and casual riders use Cyclistic bikes differently? (Cyclistic is a fictional company being used as a stand in for this analysis)

Loading packages

```
options(repos = c(CRAN = "https://cloud.r-project.org"))
install.packages("dplyr")

## Installing package into
'C:/Users/scott_kglyvae/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'dplyr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'dplyr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
##
C:\Users\scott_kglyvae\AppData\Local\R\win-library\4.5\00LOCK\dplyr\libs\x64\
dplyr.dll
## to
##
C:\Users\scott_kglyvae\AppData\Local\R\win-library\4.5\dplyr\libs\x64\dplyr.d
ll:
## Permission denied

## Warning: restored 'dplyr'

##
## The downloaded binary packages are in
##  C:\Users\scott_kglyvae\AppData\Local\Temp\Rtmp8ShA0O\downloaded_packages

install.packages("tidyverse")

## Installing package into
'C:/Users/scott_kglyvae/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked
##
```

```
## The downloaded binary packages are in
##  C:\Users\scott_kglyvae\AppData\Local\Temp\Rtmp8ShA0O\downloaded_packages
```

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

```
## Installing package into
'C:/Users/scott_kglyvae/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\scott_kglyvae\AppData\Local\Temp\Rtmp8ShA0O\downloaded_packages
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.5
## ✓ ggplot2   3.5.2     ✓ stringr   1.5.1
## ✓ lubridate 1.9.4     ✓ tibble    3.2.1
## ✓ purrr     1.0.4     ✓ tidyr     1.3.1
```

```
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(ggplot2)
```

Loading and Merging datasets:

This is the code I used to load the datasets and group them together so I can make a new dataset with the full years worth of data:

```
data_4.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202404-divvy-tripdata.csv")
data_5.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202405-divvy-tripdata.csv")
data_6.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202406-divvy-tripdata.csv")
data_7.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202407-divvy-tripdata.csv")
data_8.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202408-divvy-tripdata.csv")
data_9.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202409-divvy-tripdata.csv")
data_10.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202410-divvy-tripdata.csv")
data_11.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202411-divvy-tripdata.csv")
data_12.2024 <- read.csv("~/Data analytics Coursera/Biking
data/202412-divvy-tripdata.csv")
data_1.2025 <- read.csv("~/Data analytics Coursera/Biking
data/202501-divvy-tripdata.csv")
data_2.2025 <- read.csv("~/Data analytics Coursera/Biking
data/202502-divvy-tripdata.csv")
data_3.2025 <- read.csv("~/Data analytics Coursera/Biking
data/202503-divvy-tripdata.csv")
data_4.2025 <- read.csv("~/Data analytics Coursera/Biking
data/202504-divvy-tripdata.csv")
bike_data_year <- list(data_4.2024, data_5.2024, data_6.2024, data_7.2024,
data_8.2024, data_9.2024, data_10.2024, data_11.2024, data_12.2024,
data_1.2025, data_2.2025, data_3.2025)
year_data <- Reduce(function(x, y) merge(x, y, all=TRUE), bike_data_year)
```

Making a new dataset and adding a column for ride length:

I want to create a new dataset after merging so that I have a baseline data set of year_data. This will be year_data_v2. I will also be adding a new column that calculates the ride_length by taking the ended_at column and subtracting the started_at column. I am using mutate and difftime syntax to create the new column.

```
year_data_v2 <- year_data %>%
  mutate(ride_length = difftime(ended_at, started_at, units = "hours"))
```

Adding another column to year_data_v2:

I am adding a column for which day of the week the bike was rented on, using the lubridate package with the started_at column to determine the weekday. Adding this column in V2 of the dataset does not require a new version because it doesn't change any of the data in a substantive way and could be removed without changing anything.

```
year_data_v2$weekday <- wday(year_data_v2$started_at, label=TRUE)
```

Creating a new data set while removing negative ride length:

I am creating a new version of the dataset because I am filtering out negative ride-length data. The ride_length should not be a negative value, so I have decided to remove it from my analysis. Anytime I take out data or materially change something I will create a new version of the data set so if I find mistakes it won't require redoing coding. In this case there were 177 rows of data filtered out.

```
year_data_v3 <- year_data_v2 %>%
  filter(ride_length >= 0)
```

Calculating mean and max for ride length:

This code is calculating the mean ride length and max ride length for all users.

```
max_rl <- max(year_data_v3$ride_length, na.rm= TRUE)
print(max_rl)

## Time difference of 25.01438 hours

mean_rl <- mean(year_data_v3$ride_length, na.rm= TRUE)
print(mean_rl)

## Time difference of 0.2840451 hours
```

Getting the counts for each day of the week:

```
weekday_count <- year_data_v3 %>%
  count(weekday)
print(weekday_count)

##   weekday      n
## 1     Sun 770431
## 2     Mon 777928
## 3     Tue 781775
## 4     Wed 852647
## 5     Thu 817052
## 6     Fri 855023
## 7     Sat 924535
```

Getting the counts for if the user is a member:

I wanted to know what the difference in member count vs. casual user count would be. I ran this code to find the number of members vs. casual renters. There is a difference of around 1.5 million more members than casual renters.

```
member_casual_ct <- year_data_v3 %>%
  count(member_casual)
print(member_casual_ct)

##   member_casual       n
## 1        casual 2135187
## 2        member 3644204
```

Getting the counts for if the user is a member:

This code is showing how many riders exceed 1.0 hour rentals split by members and casual riders. True is the number of each category that had the rental last an hour or longer and False is less than an hour of rental time.

```r
ride_length_ct <- year_data_v3 %>%
  count(ride_length >= 1.0, member_casual)
print(ride_length_ct)
```

```
##   ride_length >= 1 member_casual       n
## 1            FALSE        casual 2010531
## 2            FALSE        member 3619134
## 3             TRUE        casual  124656
## 4             TRUE        member   25070
```

Getting the average ride length:

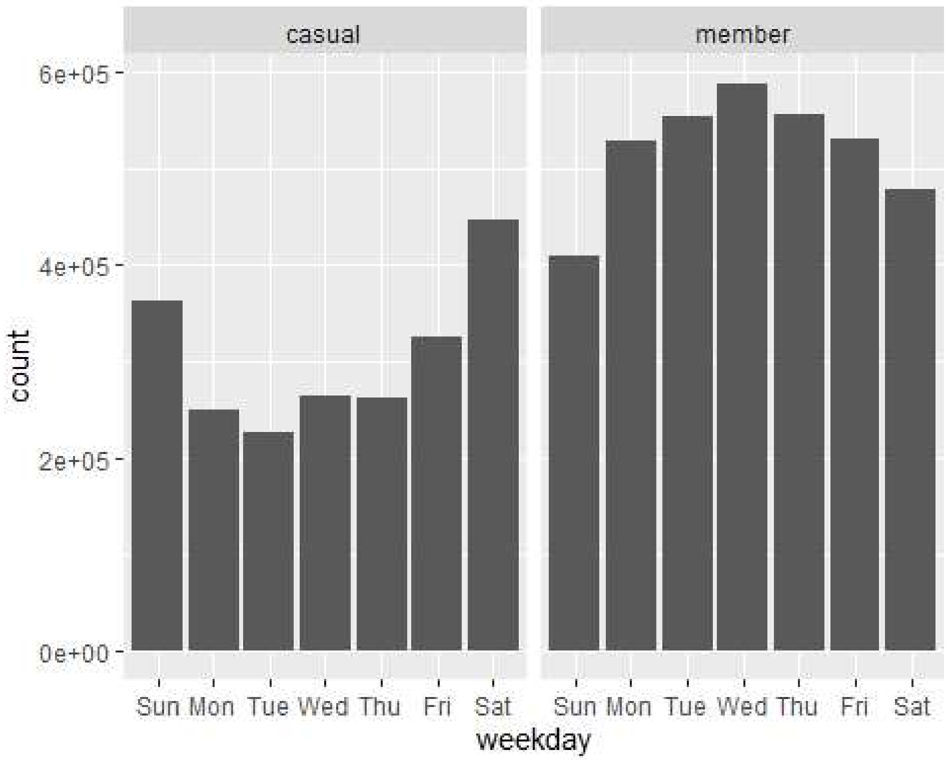
```r
ride_length_avg <- year_data_v3 %>%
  group_by(member_casual) %>%
  summarise(avg_ride_length = mean(ride_length, na.rm = TRUE))
print(ride_length_avg)
```

```
## # A tibble: 2 × 2
##   member_casual avg_ride_length
##   <chr>         <drtn>
## 1 casual        0.4128853 hours
## 2 member        0.2085559 hours
```

Plotting weekday counts for casual and members:

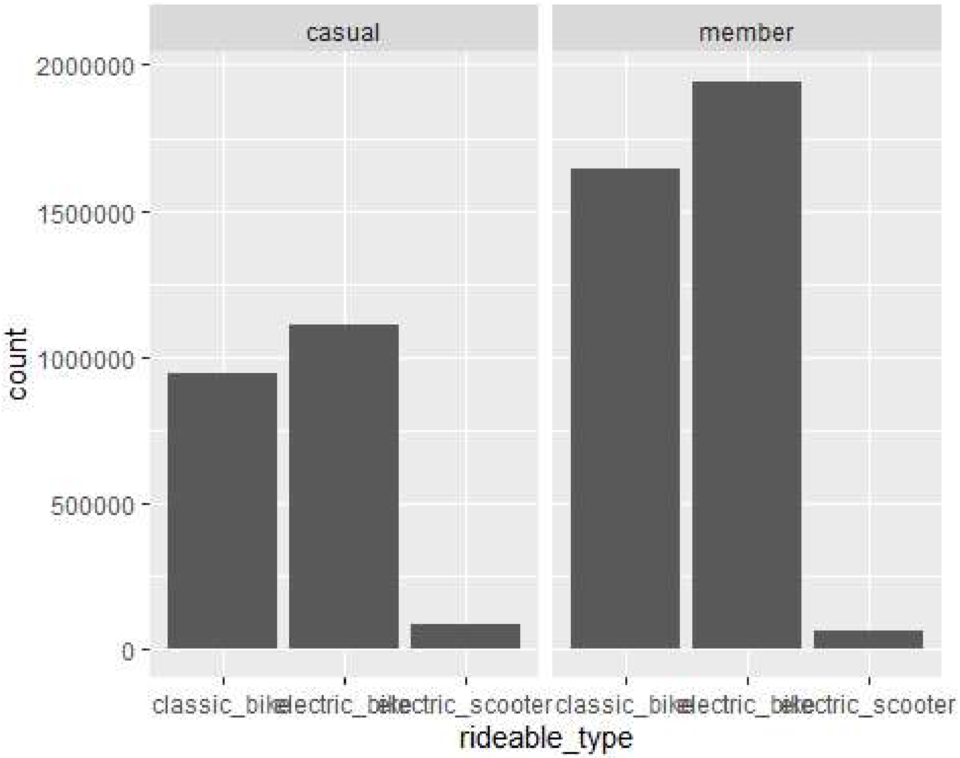
This bar graph shows the counts for weekday rides by whether the user is a casual renter or a member of the service. I am using this graph to show what days the casual members prefer to use the service. I want this information so we can compare to the members, and it might be useful to marketing to help

```r
ggplot(data= year_data_v3)+
  geom_bar(mapping= aes(weekday))+
  facet_grid(.~member_casual)
```



Plotting counts of each type of rideable rental for causal and members:

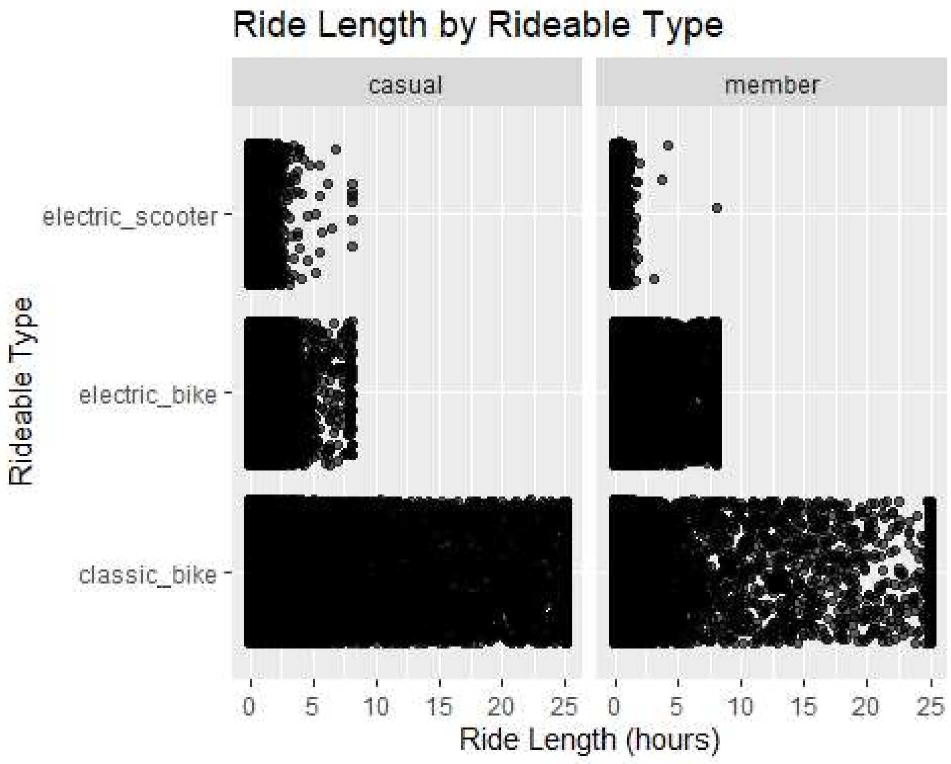
```
ggplot(data=year_data_v3)+
  geom_bar(mapping= aes(rideable_type))+
  facet_grid(.~member_casual)
```



Plotting ride lenght by ride type counts for casual and members:

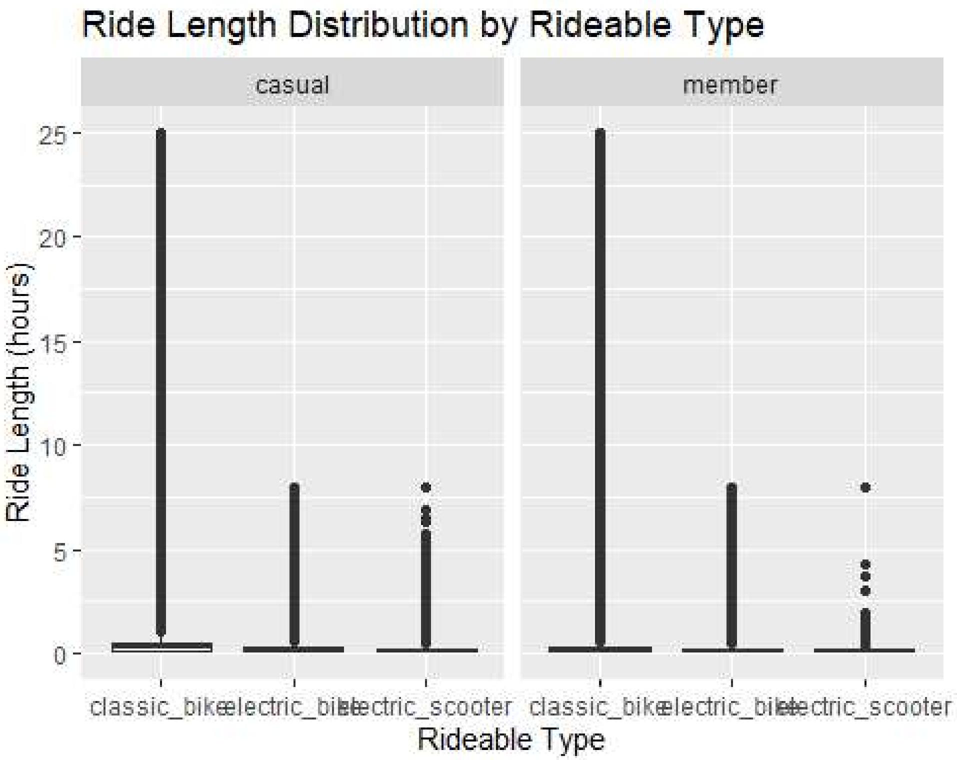
I wanted to plot the ride lengths by what type of bike the user rents. This is a jitter graph of this data.

```
ggplot(data = year_data_v3) +
  geom_jitter(mapping = aes(x = ride_length, y = rideable_type), width = 0.1,
alpha = 0.6) +
  facet_grid(. ~ member_casual) +
  labs(title = "Ride Length by Rideable Type",
       x = "Ride Length (hours)",
       y = "Rideable Type")

## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



I also made a box graph of the same data:

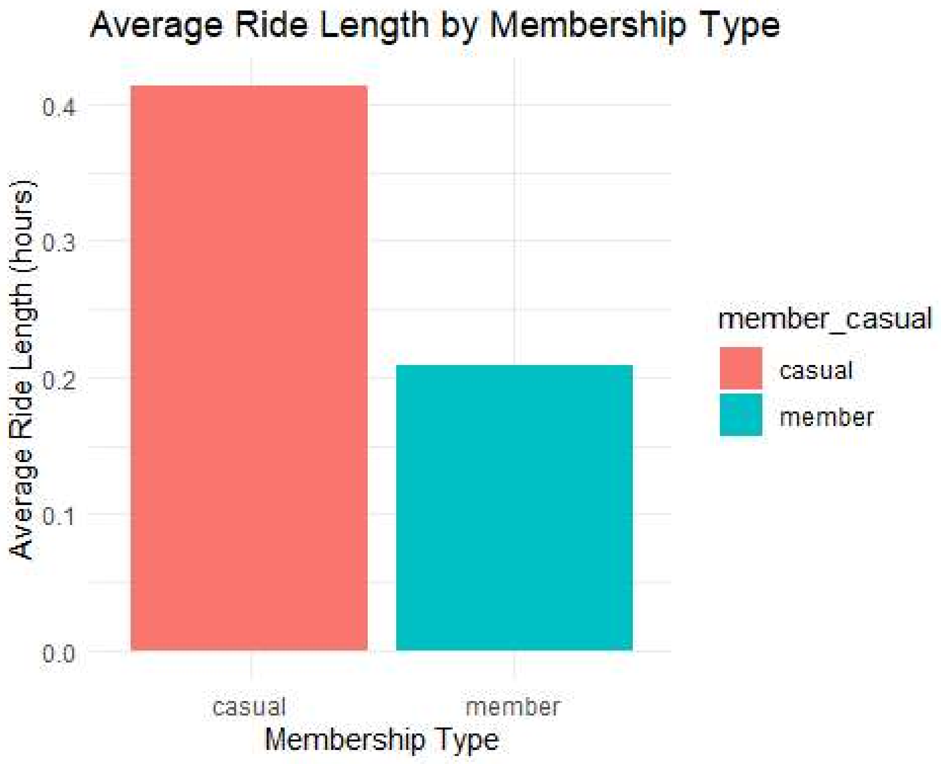
```
ggplot(data = year_data_v3) +
  geom_boxplot(mapping = aes(x = rideable_type, y = ride_length)) +
  facet_grid(. ~ member_casual) +
  labs(title = "Ride Length Distribution by Rideable Type",
       x = "Rideable Type",
       y = "Ride Length (hours)")

## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



Plotting a bar graph for ride length average by user type:

```
ggplot(ride_length_avg, aes(x = member_casual, y = avg_ride_length, fill =
member_casual)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Ride Length by Membership Type",
       x = "Membership Type",
       y = "Average Ride Length (hours)") +
  theme_minimal()
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



Conclusion:

After performing my data analysis as shown by the above code I have determined that there are some ways a casual biker use the bike rental service differently than members. The data shows that casual bikers tend to use the service more on the weekdays than weekends while members tend to use the service on weekends. There are approximately 1.5 million more members than casual bikers.Casual users are more likely to rent for over an hour than members.

They tend to use the same ratio of e-bikes, scooters, and classic bikes. I would point out that this might have to do with availability and would recommend a seperate data analysis based on where the e-bikes, and scooters are .