Cyclistic - Sample Data Analytics

Olivia Brubaker

10/05/2024

Introduction

Cyclistic, a bike rental company located in Chicago, Illinois, has noticed an increased profit margin for individuals who are subscribed to the annual membership. In an effort to craft a campaign to convert the everyday consumer into a member, this report seeks to capture recent trends of the "casual" customer through data analysis.

The Data

The data analyzed in this project covers Cyclistic's rental data from the first quarter of 2020. The data has been made available through the following license.

In the repository, the data is divided by quarter, and includes the following categories: * Unique trip identifier

- Bicycle type: Electric or Classic
- Starting Location: Includes name of the cross streets, latitude and longitude
- Trip End Location: Includes name of cross streets, latitude, and longitude
- The trip's date and when the trip began and ended
- Member Classification: If the member is an annual subscriber, the data assigns "member". To the other classification, it is labeled "casual", and that represents consumers who have rented a bicycle for their one-time, daily, or monthly passes.

Cleaning the Data

The data was saved in a zip file to my local drive under the label "quarterly data". In the dataframe conversion, there was a conversion of data type for the start and end of the trip. The code reflects the column conversion to DateTime, which required a standardization.

```
install.packages("readr", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/odbgi/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'readr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\odbgi\AppData\Local\Temp\Rtmpe4J0F7\downloaded_packages
```

```
install.packages("tidyr", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/odbgi/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'tidyr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\odbgi\AppData\Local\Temp\Rtmpe4J0F7\downloaded_packages
```

```
install.packages("dplyr", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/odbgi/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'dplyr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\odbgi\AppData\Local\Temp\Rtmpe4J0F7\downloaded_packages
```

```
install.packages("lubridate", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/odbgi/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'lubridate' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\odbgi\AppData\Local\Temp\Rtmpe4J0F7\downloaded_packages
```

```
install.packages("ggplot2", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/odbgi/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\odbgi\AppData\Local\Temp\Rtmpe4J0F7\downloaded_packages
```

```
library(readr)
library(tidyr)
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(lubridate)
```

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

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

```
library(ggplot2)
```

```
twen1 <- read_csv("quarterly data/Divvy_Trips_2020_Q1.csv",
    col_types = cols(started_at = col_datetime(format = "%Y-%m-%d %H:%M:%S"),
        ended_at = col_datetime(format = "%Y-%m-%d %H:%M:%S")))
```

In total, there was a collection of 426,887 trips documented over the quarter. To verify, I checked for distinct values in the trip identifier.

```
uniq_id <- nrow(twen1 %>% distinct(ride_id))
nrow(twen1) == uniq_id
```

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

Next, I needed to identify and analyze the number of ride observations that had at least one "N/A" on a variable entry. A summary table was generated by the following code:

```
na_counts_dplyr <- twen1 %>%
  summarise_all(~ sum(is.na(.)))
```

The summary table identifies one unknown in each of the following columns: end_station_name, end_station_id, end_lng, and end_lat. I suspected these were in the same observation, but I verified.

```
twen1_no_na <- na.omit(twen1)

proportion_na <- nrow(twen1_no_na)/nrow(twen1)

na_stats <- c(proportion_na, 1-proportion_na)
label_na_stats <- c("Definitive Observations", "At least one N/A")

png(file = 'na_pie.png')
pie(na_stats, label_na_stats, main = 'N/A Proportion')

write.csv(twen1_no_na, "q1_2020.csv")
```

Additional Calculations

To flesh out these observations more, we will add some additional aspects through calculations. We will add the duration of the ride, and the weekday on which the trip occurred.

```
# Add duration
twen1_add_dur <- twen1_no_na  %>% mutate(ride_length = difftime(ended_at, started_at))

# Add weekday
twen1_calc <- twen1_add_dur  %>% mutate(day_of_week = weekdays(started_at))

# Cleaning calculation columns
rm(twen1_add_dur)

twen1_calc$ride_length <- as.numeric(as.character(twen1_calc$ride_length))
is.numeric(twen1_calc$ride_length)
```

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

```
twen1_v2 <- twen1_calc[!(twen1_calc$start_station_name == "HQ QR" | twen1_calc$ride_length<0),]
```

Analysis

To begin our analysis, we will break down the summary statistics for the trip duration, and compare how long rides were for annual subscribers versus casual riders.

```
mean(twen1_v2$ride_length)
```

```
## [1] 1338.697
```

```
median(twen1_v2$ride_length)
```

```
## [1] 555
```

```
max(twen1_v2$ride_length)
```

```
## [1] 9387024
```

```
min(twen1_v2$ride_length)
```

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

```
aggregate(twen1_v2$ride_length ~ twen1_v2$member_casual, FUN = mean)
```

```
##   twen1_v2$member_casual twen1_v2$ride_length
## 1                 casual            6230.7734
## 2                 member             760.6287
```

```
aggregate(twen1_v2$ride_length ~ twen1_v2$member_casual, FUN = median)
```

```
##   twen1_v2$member_casual twen1_v2$ride_length
## 1                 casual                 1389
## 2                 member                  515
```

```
aggregate(twen1_v2$ride_length ~ twen1_v2$member_casual, FUN = max)
```

```
##   twen1_v2$member_casual twen1_v2$ride_length
## 1                 casual              9387024
## 2                 member              5627611
```

```
aggregate(twen1_v2$ride_length ~ twen1_v2$member_casual, FUN = min)
```

```
##   twen1_v2$member_casual twen1_v2$ride_length
## 1                 casual                    2
## 2                 member                    1
```

```
aggregate(twen1_v2$ride_length ~ twen1_v2$member_casual + twen1_v2$day_of_week, FUN = mean)
```

```
##    twen1_v2$member_casual twen1_v2$day_of_week twen1_v2$ride_length
## 1                  casual               Friday            7907.8883
## 2                  member               Friday             757.3241
## 3                  casual               Monday            5818.3439
## 4                  member               Monday             778.6286
## 5                  casual             Saturday            6017.1560
## 6                  member             Saturday             929.9892
## 7                  casual               Sunday            5710.5665
## 8                  member               Sunday             949.3401
## 9                  casual             Thursday            8744.6574
## 10                 member             Thursday             693.2325
## 11                 casual              Tuesday            5832.3594
## 12                 member              Tuesday             692.0323
## 13                 casual            Wednesday            5132.6226
```

```
## 14                 member            Wednesday             699.5471
```

```
twen1_v2$day_of_week <- ordered(twen1_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesda
```

```
aggregate(twen1_v2$ride_length ~ twen1_v2$member_casual + twen1_v2$day_of_week, FUN = mean)
```

```
##    twen1_v2$member_casual twen1_v2$day_of_week twen1_v2$ride_length
## 1                  casual               Sunday            5710.5665
## 2                  member               Sunday             949.3401
## 3                  casual               Monday            5818.3439
## 4                  member               Monday             778.6286
## 5                  casual              Tuesday            5832.3594
## 6                  member              Tuesday             692.0323
## 7                  casual            Wednesday            5132.6226
## 8                  member            Wednesday             699.5471
## 9                  casual             Thursday            8744.6574
## 10                 member             Thursday             693.2325
## 11                 casual               Friday            7907.8883
## 12                 member               Friday             757.3241
## 13                 casual             Saturday            6017.1560
## 14                 member             Saturday             929.9892
```

```
twen1_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>% group_by(member_casual, weekday) %>% summarise(nu
```

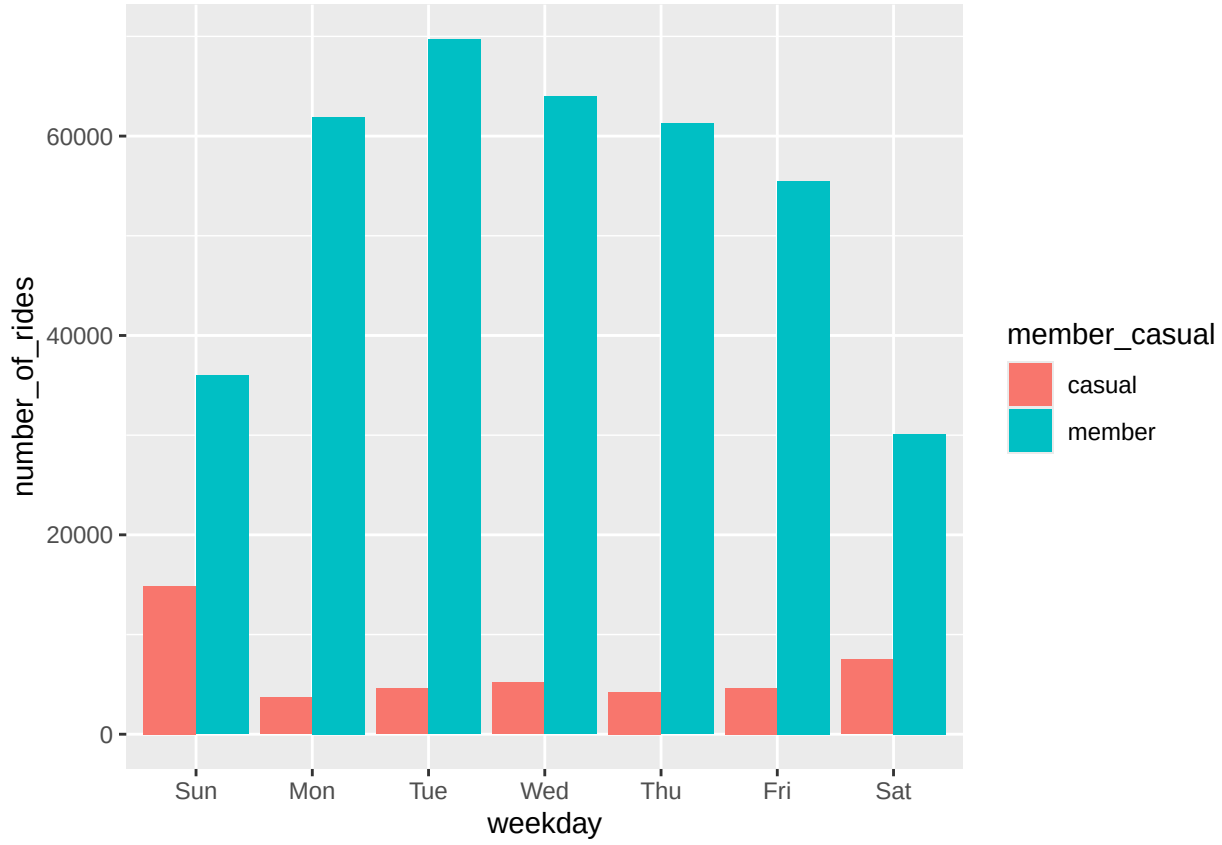
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun               14886            5711.
##  2 casual        Mon                3699            5818.
##  3 casual        Tue                4583            5832.
##  4 casual        Wed                5201            5133.
##  5 casual        Thu                4227            8745.
##  6 casual        Fri                4638            7908.
##  7 casual        Sat                7480            6017.
##  8 member        Sun               35964             949.
##  9 member        Mon               61923             779.
## 10 member        Tue               69697             692.
## 11 member        Wed               63977             700.
## 12 member        Thu               61245             693.
## 13 member        Fri               55496             757.
## 14 member        Sat               30104             930.
```

```
twen1_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 = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

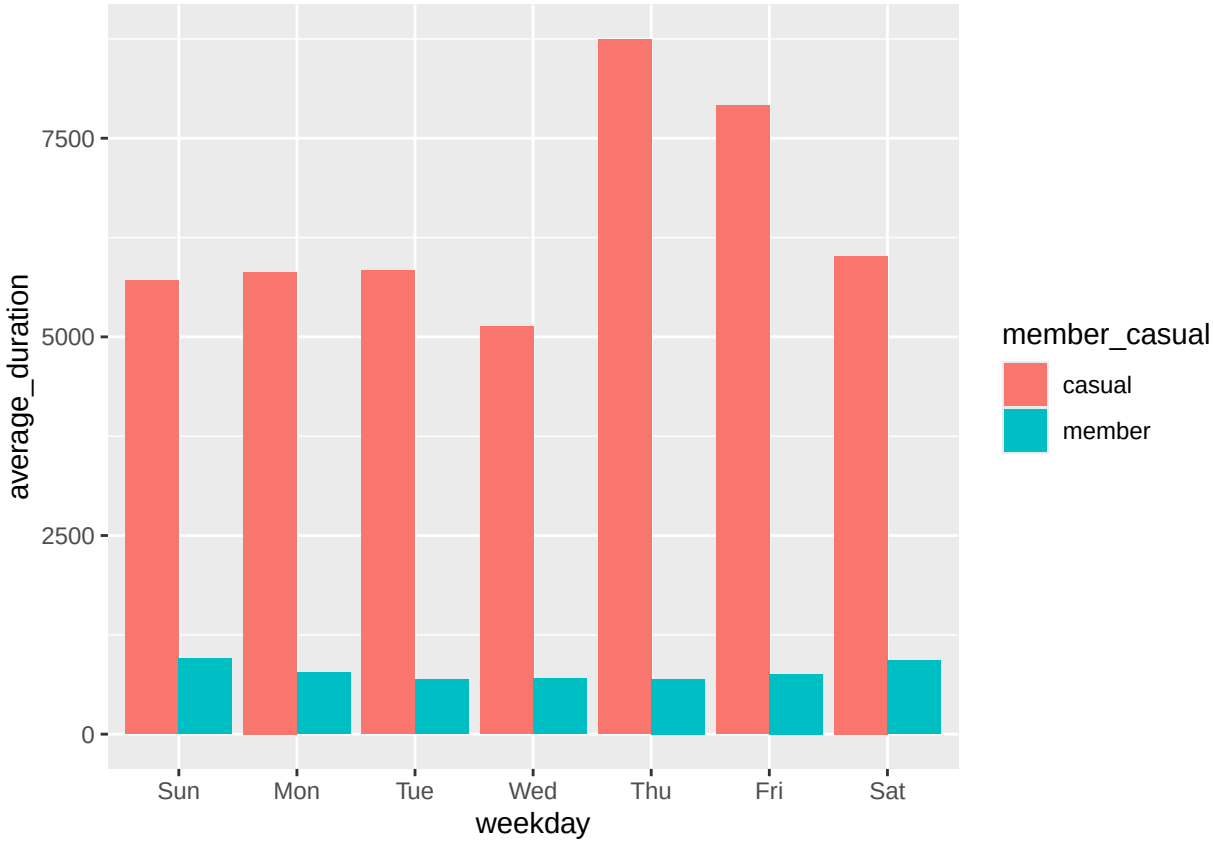
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
```

```
## `.groups` argument.
```




```
twen1_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")
```

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




```
counts <- aggregate(twen1_v2$ride_length ~ twen1_v2$member_casual + twen1_v2$day_of_week, FUN = mean)
write.csv(counts, file = 'avg_ride_length.csv')
```