Cyclistic Case Study Analysis

Zamir Said

2023-09-15

Introduction

This is the analysis for the Cyclistic Case Study project from the Google Data Analytics Professional Certificate. In this analysis I will import the data, clean the data, analyze the data, and produce visualizations to help answer the question that was asked at the beginning at the project which is: “How do annual members and casual riders use Cyclistic bikes differently?”. The datasets are separated by month and year, and for this analysis I will only use one month of data, January 2023. This is because I am performing this analysis in R Studio Cloud, and performing an analysis on the combined year of data crashes the server.

Step 1: Import The Data

Import libraries.

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(lubridate)
library(ggplot2)
```

Import the data for Janurary 2023.

```r
jan_2023 <- read_csv("/cloud/project/cyclistic_data/202301-tripdata.csv")
```

```
## New names:
## Rows: 190301 Columns: 14
## -- 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 lgl (1): ...14
## 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.
## * `` -> `...14`
```

Step 2: Data Wrangling

Inspect the column names and the structure of the columns.

```
colnames(jan_2023)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"        
##  [4] "ended_at"           "start_station_name" "start_station_id"  
##  [7] "end_station_name"   "end_station_id"     "start_lat"         
## [10] "start_lng"          "end_lat"            "end_lng"           
## [13] "member_casual"      "...14"
```

```
str(jan_2023)
```

```
## spc_tbl_ [190,301 x 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:190301] "F96D5A74A3E41399" "13CB7EB698CEDB88" "BD88A2E670661CE5" "C9079
##  $ rideable_type     : chr [1:190301] "electric_bike" "classic_bike" "electric_bike" "classic_bike" 
##  $ started_at        : chr [1:190301] "2023-01-21 20:05" "2023-01-10 15:37" "2023-01-02 7:51" "2023-0
##  $ ended_at          : chr [1:190301] "2023-01-21 20:16" "2023-01-10 15:46" "2023-01-02 8:05" "2023-0
##  $ start_station_name: chr [1:190301] "Lincoln Ave & Fullerton Ave" "Kimbark Ave & 53rd St" "Western 
##  $ start_station_id  : chr [1:190301] "TA1309000058" "TA1309000037" "RP-005" "TA1309000037" ...
##  $ end_station_name  : chr [1:190301] "Hampden Ct & Diversey Ave" "Greenwood Ave & 47th St" "Valli P
##  $ end_station_id    : chr [1:190301] "202480" "TA1308000002" "599" "TA1308000002" ...
##  $ start_lat         : num [1:190301] 41.9 41.8 42 41.8 41.8 ...
##  $ start_lng         : num [1:190301] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ end_lat           : num [1:190301] 41.9 41.8 42 41.8 41.8 ...
##  $ end_lng           : num [1:190301] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:190301] "member" "member" "casual" "member" ...
##  $ ...14             : logi [1:190301] NA NA NA NA NA NA ...
##  - 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(),
##   ..   ...14 = col_logical()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Remove columns that are irrelevant for this analysis.

```
jan_2023 <- jan_2023 %>%
  select(-c(start_lat, start_lng, end_lat, end_lng))
```

Step 3: Data Cleaning

Inspect the column names and the structure of the newly created table.

```
colnames(jan_2023)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "member_casual"
## [10] "...14"
```

```
str(jan_2023)
```

```
## tibble [190,301 x 10] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:190301] "F96D5A74A3E41399" "13CB7EB698CEDB88" "BD88A2E670661CE5" "C9079
##  $ rideable_type     : chr [1:190301] "electric_bike" "classic_bike" "electric_bike" "classic_bike"
##  $ started_at        : chr [1:190301] "2023-01-21 20:05" "2023-01-10 15:37" "2023-01-02 7:51" "2023-0
##  $ ended_at          : chr [1:190301] "2023-01-21 20:16" "2023-01-10 15:46" "2023-01-02 8:05" "2023-0
##  $ start_station_name: chr [1:190301] "Lincoln Ave & Fullerton Ave" "Kimbark Ave & 53rd St" "Western
##  $ start_station_id  : chr [1:190301] "TA1309000058" "TA1309000037" "RP-005" "TA1309000037" ...
##  $ end_station_name  : chr [1:190301] "Hampden Ct & Diversey Ave" "Greenwood Ave & 47th St" "Valli Pi
##  $ end_station_id    : chr [1:190301] "202480" "TA1308000002" "599" "TA1308000002" ...
##  $ member_casual     : chr [1:190301] "member" "member" "casual" "member" ...
##  $ ...14             : logi [1:190301] NA NA NA NA NA NA ...
```

Inspect number of rows and columns in the data frame.

```
nrow(jan_2023)
```

```
## [1] 190301
```

```
dim(jan_2023)
```

```
## [1] 190301     10
```

Inspect first 6 rows.

```
head(jan_2023)
```

```
## # A tibble: 6 x 10
##   ride_id  rideable_type started_at ended_at start_station_name start_station_id
##   <chr>    <chr>         <chr>      <chr>    <chr>              <chr>
## 1 F96D5A7~ electric_bike 2023-01-2~ 2023-01~ Lincoln Ave & Ful~ TA1309000058
## 2 13CB7EB~ classic_bike  2023-01-1~ 2023-01~ Kimbark Ave & 53r~ TA1309000037
## 3 BD88A2E~ electric_bike 2023-01-0~ 2023-01~ Western Ave & Lun~ RP-005
## 4 C90792D~ classic_bike  2023-01-2~ 2023-01~ Kimbark Ave & 53r~ TA1309000037
## 5 3397017~ classic_bike  2023-01-1~ 2023-01~ Kimbark Ave & 53r~ TA1309000037
## 6 58E6815~ electric_bike 2023-01-3~ 2023-01~ Lakeview Ave & Fu~ TA1309000019
## # i 4 more variables: end_station_name <chr>, end_station_id <chr>,
## #   member_casual <chr>, ...14 <lgl>
```

Statistical summary of the data.

```
summary(jan_2023)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:190301      Length:190301      Length:190301      Length:190301     
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##  start_station_name start_station_id   end_station_name   end_station_id    
##  Length:190301      Length:190301      Length:190301      Length:190301     
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##  member_casual        ...14         
##  Length:190301      Mode:logical    
##  Class :character   NA's:190301     
##  Mode  :character
```

Add columns that list the date, day, and day of week.

```
jan_2023$date <- as.Date(jan_2023$started_at)
jan_2023$day <- format(as.Date(jan_2023$date), "%d")
jan_2023$day_of_week <- format(as.Date(jan_2023$date), "%A")
```

Add a ride length column (in seconds).

```
jan_2023$ride_length <- difftime(jan_2023$ended_at,jan_2023$started_at)
```

Convert ride_length column from factor to numeric so we can run calculations on the data.

```
is.factor(jan_2023$ride_length)
```

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

```
jan_2023$ride_length <- as.numeric(as.character(jan_2023$ride_length))
is.numeric(jan_2023$ride_length)
```

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

Step 4: Descriptive Analysis

How many observations fall under each type of rider?

```
table(jan_2023$member_casual)
```

```
## 
## casual member 
##  40008 150293
```

Descriptive analysis on ride_length (all figures in seconds).

```
summary(jan_2023$ride_length)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
##       0.0     240.0     420.0     780.1     720.0 2016240.0
```

Compare descriptive statistics for members and casual users.

```
aggregate(jan_2023$ride_length ~ jan_2023$member_casual, FUN = mean)
```

```
##   jan_2023$member_casual jan_2023$ride_length
## 1                 casual            1375.0165
## 2                 member             621.7128
```

```
aggregate(jan_2023$ride_length ~ jan_2023$member_casual, FUN = median)
```

```
##   jan_2023$member_casual jan_2023$ride_length
## 1                 casual                  480
## 2                 member                  420
```

```
aggregate(jan_2023$ride_length ~ jan_2023$member_casual, FUN = max)
```

```
##   jan_2023$member_casual jan_2023$ride_length
## 1                 casual              2016240
## 2                 member                90000
```

```
aggregate(jan_2023$ride_length ~ jan_2023$member_casual, FUN = min)
```

```
##   jan_2023$member_casual jan_2023$ride_length
## 1                 casual                    0
## 2                 member                    0
```

Inspect average ride time by each day for members compared to casual riders.

```
jan_2023$day_of_week <- ordered(jan_2023$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday
aggregate(jan_2023$ride_length ~ jan_2023$member_casual + jan_2023$day_of_week, FUN = mean)
```

```
##    jan_2023$member_casual jan_2023$day_of_week jan_2023$ride_length
## 1                  casual               Sunday            1996.8606
## 2                  member               Sunday             694.9928
## 3                  casual               Monday            1325.4124
## 4                  member               Monday             618.7584
## 5                  casual              Tuesday            1090.8546
## 6                  member              Tuesday             603.2005
## 7                  casual            Wednesday            1138.8960
## 8                  member            Wednesday             606.5352
## 9                  casual             Thursday            1310.5974
## 10                 member             Thursday             590.1965
## 11                 casual               Friday            1213.0846
## 12                 member               Friday             630.2879
## 13                 casual             Saturday            1539.5854
## 14                 member             Saturday             645.7885
```

Analyze ridership data by rider type and weekday.

```
jan_2023 %>%
  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)
```

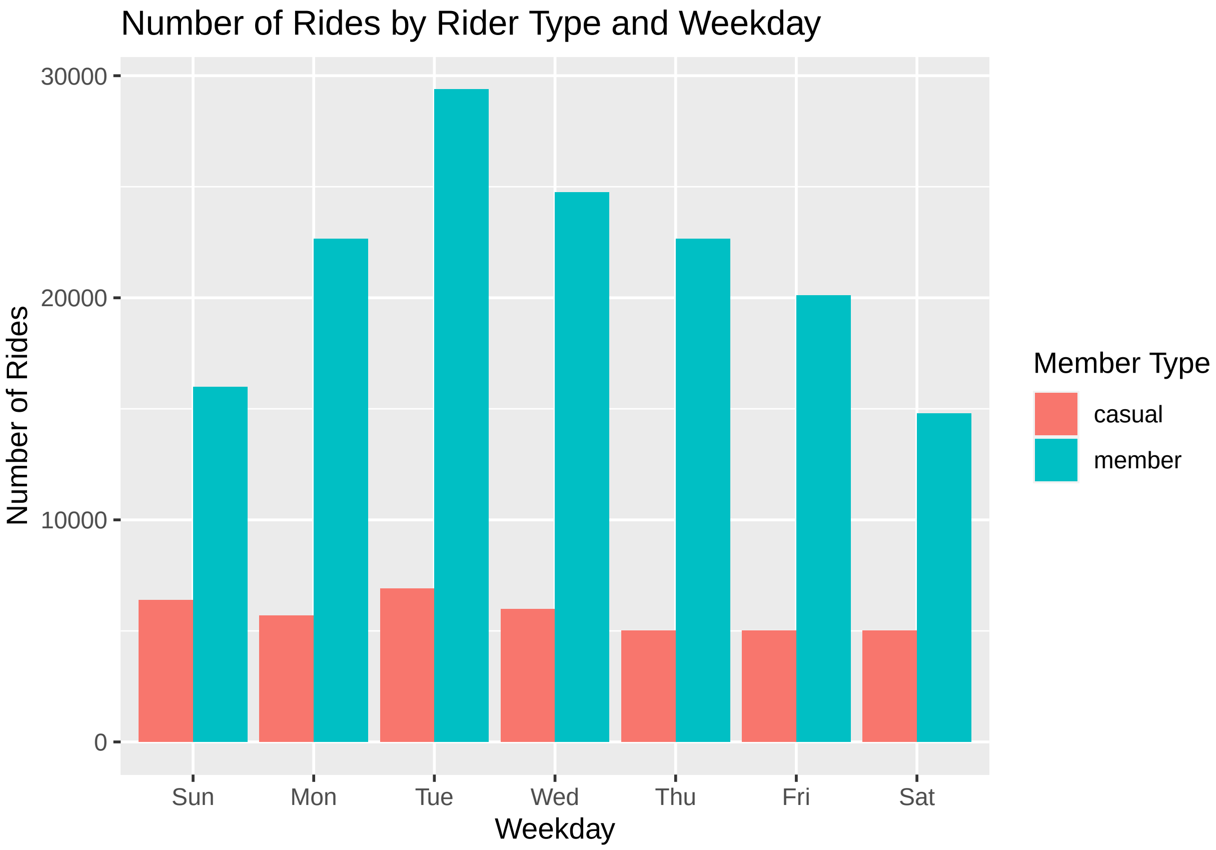
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun                6377            1997.
##  2 casual        Mon                5698            1325.
##  3 casual        Tue                6904            1091.
##  4 casual        Wed                5978            1139.
##  5 casual        Thu                5022            1311.
##  6 casual        Fri                5012            1213.
##  7 casual        Sat                5017            1540.
##  8 member        Sun               15989             695.
##  9 member        Mon               22649             619.
## 10 member        Tue               29377             603.
## 11 member        Wed               24743             607.
## 12 member        Thu               22645             590.
## 13 member        Fri               20109             630.
## 14 member        Sat               14781             646.
```

Visualize the number of rides by rider type and weekday.

```
jan_2023 %>%
  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") +
  labs(x = "Weekday", y = "Number of Rides", title = "Number of Rides by Rider Type and Weekday", fill =
```

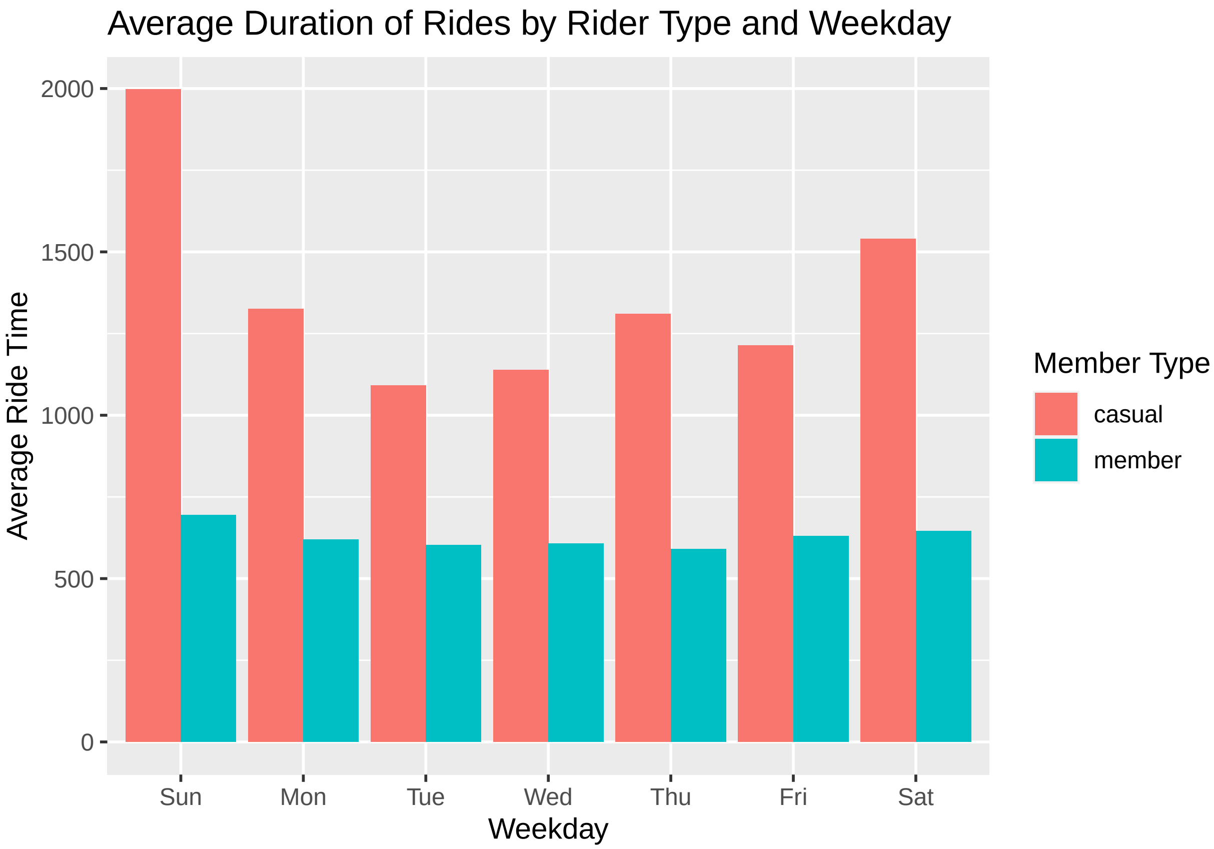
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



Visualization the average duration of rides by rider type and weekday.

```
jan_2023 %>%
  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(x = "Weekday", y = "Average Ride Time", title = "Average Duration of Rides by Rider Type and Weel
```

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



Note:

This R code is based off of a template R script provided by the Google Data Analytics course.