Cyclistic_Analysis

Ayodeji Yekeen

12/16/2021

Cyclistic_Case_Study_Full_Year_Analysis

This analysis is based on the Cyclistic case study

Install required packages

tidyverse for data import and wrangling lubridate for date functions ggplot for visualization

```
library(tidyverse)  #helps wrangle data
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.1 ──
```

```
## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.6     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   2.1.1     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── 
tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)  #helps wrangle date attributes
```

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

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

```
library(ggplot2)  #helps visualize data
getwd() #displays your working directory
```

```
## [1] "/home/ayodeji/Downloads/Google Data Analytics Professional
Certificate/CASE STUDY I/FILES/CSV"
```

```
setwd("~/Downloads/Google Data Analytics Professional Certificate/CASE STUDY
I/FILES/CSV/") #sets your working directory to simplify calls to data ...
```

STEP 1: COLLECT DATA

Read in the dataset here

```
all_trips <- read.csv('combined_dataset.csv')
```

STEP 2: WRANGLE DATA

```
all_trips$date <- as.Date(all_trips$started_at)
```

STEP 3: CLEAN UP AND ADD DATA TO PREPARE FOR ANALYSIS

Inspecting the newly created table.

```
colnames(all_trips)  #List of column names
```

```
##  [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"      "ride_length"        "day_of_week"
## [16] "date"
```

```
nrow(all_trips)  #How many rows are in data frame?
```

```
## [1] 5479096
```

```
dim(all_trips)  #Dimensions of the data frame?
```

```
## [1] 5479096      16
```

```
head(all_trips)  #See the first 6 rows of data frame.
```

```
##            ride_id rideable_type       started_at         ended_at
## 1 70B6A9A437D4C30D  classic_bike 27/12/2020 12:44 27/12/2020 12:55
## 2 158A465D4E74C54A electric_bike 18/12/2020 17:37 18/12/2020 17:44
## 3 5262016E0F1F2F9A electric_bike 15/12/2020 15:04 15/12/2020 15:11
## 4 BE119628E44F871E electric_bike 15/12/2020 15:54 15/12/2020 16:00
## 5 69AF78D57854E110 electric_bike 22/12/2020 12:08 22/12/2020 12:10
## 6 C1DECC4AB488831C electric_bike 22/12/2020 13:26 22/12/2020 13:34
##           start_station_name start_station_id          end_station_name
## 1 Aberdeen St & Jackson Blvd            13157 Desplaines St & Kinzie St
## 2
## 3
## 4
## 5
## 6
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   TA1306000003  41.87773 -87.65479 41.88872 -87.64445        member
## 2                 41.93000 -87.70000 41.91000 -87.70000        member
## 3                 41.91000 -87.69000 41.93000 -87.70000        member
## 4                 41.92000 -87.70000 41.91000 -87.70000        member
## 5                 41.80000 -87.59000 41.80000 -87.59000        member
```

```
## 6                  41.80000 -87.59000 41.78000 -87.60000        member
##   ride_length day_of_week     date
## 1     0:10:37           1 27-12-20
## 2     0:07:04           6 18-12-20
## 3     0:06:55           3 15-12-20
## 4     0:05:53           3 15-12-20
## 5     0:02:42           3 22-12-20
## 6     0:08:13           3 22-12-20

tail(all_trips)  #See the last 6 rows of data frame.

##                   ride_id rideable_type       started_at         ended_at
## 5479091 2E383B4D2965B154 electric_bike 04/11/2021 16:59 04/11/2021 17:08
## 5479092 E00E9F3500D69BAA electric_bike  29/11/2021 0:39  29/11/2021 0:51
## 5479093 8EAA66CE314E5FF1 electric_bike 03/11/2021 13:56 03/11/2021 14:01
## 5479094 36C2DC8BB1E13491 electric_bike 02/11/2021 19:32 02/11/2021 19:36
## 5479095 8E42FE5C67DF6A96 electric_bike 10/11/2021 20:15 10/11/2021 20:22
## 5479096 4F15069E2D2519BC electric_bike 30/11/2021 20:18 30/11/2021 20:37
##                      start_station_name start_station_id end_station_name
## 5479091 Cityfront Plaza Dr & Pioneer Ct            13427
## 5479092         Logan Blvd & Elston Ave     TA1308000031
## 5479093         Logan Blvd & Elston Ave     TA1308000031
## 5479094         Logan Blvd & Elston Ave     TA1308000031
## 5479095         Logan Blvd & Elston Ave     TA1308000031
## 5479096         Ogden Ave & Chicago Ave     TA1305000020
##         end_station_id start_lat start_lng end_lat end_lng member_casual
## 5479091                 41.89021 -87.62151   41.88  -87.63        member
## 5479092                 41.92945 -87.68420   41.93  -87.72        member
## 5479093                 41.92944 -87.68418   41.94  -87.69        member
## 5479094                 41.92945 -87.68414   41.94  -87.69        member
## 5479095                 41.92943 -87.68418   41.94  -87.69        member
## 5479096                 41.89635 -87.65398   41.95  -87.70        member
##         ride_length day_of_week     date
## 5479091     0:09:17           5  4-11-20
## 5479092     0:12:28           2 29-11-20
## 5479093     0:04:54           4  3-11-20
## 5479094     0:03:58           3  2-11-20
## 5479095     0:06:55           4 10-11-20
## 5479096     0:19:27           3 30-11-20

str(all_trips)  #See list of columns and data types (numeric, character, etc)

## 'data.frame':    5479096 obs. of  16 variables:
##  $ ride_id           : chr  "70B6A9A437D4C30D" "158A465D4E74C54A"
"5262016E0F1F2F9A" "BE119628E44F871E" ...
##  $ rideable_type     : chr  "classic_bike" "electric_bike" "electric_bike"
"electric_bike" ...
##  $ started_at        : chr  "27/12/2020 12:44" "18/12/2020 17:37"
"15/12/2020 15:04" "15/12/2020 15:54" ...
##  $ ended_at          : chr  "27/12/2020 12:55" "18/12/2020 17:44"
"15/12/2020 15:11" "15/12/2020 16:00" ...
```

```
##  $ start_station_name: chr  "Aberdeen St & Jackson Blvd" "" "" "" ...
##  $ start_station_id  : chr  "13157" "" "" "" ...
##  $ end_station_name  : chr  "Desplaines St & Kinzie St" "" "" "" ...
##  $ end_station_id    : chr  "TA1306000003" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.8 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.8 ...
##  $ end_lng           : num  -87.6 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
##  $ ride_length       : chr  "0:10:37" "0:07:04" "0:06:55" "0:05:53" ...
##  $ day_of_week       : int  1 6 3 3 3 3 5 5 7 6 ...
##  $ date              : Date, format: "27-12-20" "18-12-20" ...

summary(all_trips)  #Statistical summary of data. Mainly for numerics

##    ride_id          rideable_type       started_at          ended_at
##  Length:5479096     Length:5479096     Length:5479096     Length:5479096
##  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:5479096     Length:5479096     Length:5479096     Length:5479096
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    start_lat       start_lng         end_lat         end_lng
##  Min.   :41.64   Min.   :-87.84   Min.   :41.39   Min.   :-88.97
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.07   Max.   :-87.52   Max.   :42.17   Max.   :-87.49
##                                   NA's   :4738    NA's   :4738
##  member_casual      ride_length         day_of_week       date
##  Length:5479096     Length:5479096     Min.   :1.0   Min.   :1-01-20
##  Class :character   Class :character   1st Qu.:2.0   1st Qu.:8-07-20
##  Mode  :character   Mode  :character   Median :4.0   Median :16-03-20
##                                        Mean   :4.1   Mean   :16-01-30
##                                        3rd Qu.:6.0   3rd Qu.:23-06-20
##                                        Max.   :7.0   Max.   :31-12-20
##
```

There are a few problems needed to be fixed: - The data can only be aggregated at the ride-level, which is too granular. We will want to add some additional columns of data – such as

day, month, year – that provide additional opportunities to aggregate the data. - There are some rides where tripduration shows up as negative, including several hundred rides where Divvy took bikes out of circulation for Quality Control reasons. We will want to delete these rides.

Let's see how many observations fall under each usertype

```
table(all_trips$member_casual)

##
##  casual  member
## 2489347 2989749
```

Let's see how many observations fall under each rideable_type

```
table(all_trips$rideable_type)

##
##  classic_bike   docked_bike electric_bike
##       3221009        320419       1937668
```

Add a "ride_length" calculation to all_trips (in seconds)

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

Inspect the structure of the columns

```
str(all_trips)

## 'data.frame':    5479096 obs. of  16 variables:
##  $ ride_id           : chr  "70B6A9A437D4C30D" "158A465D4E74C54A"
"5262016E0F1F2F9A" "BE119628E44F871E" ...
##  $ rideable_type     : chr  "classic_bike" "electric_bike" "electric_bike"
"electric_bike" ...
##  $ started_at        : chr  "27/12/2020 12:44" "18/12/2020 17:37"
"15/12/2020 15:04" "15/12/2020 15:54" ...
##  $ ended_at          : chr  "27/12/2020 12:55" "18/12/2020 17:44"
"15/12/2020 15:11" "15/12/2020 16:00" ...
##  $ start_station_name: chr  "Aberdeen St & Jackson Blvd" "" "" "" ...
##  $ start_station_id  : chr  "13157" "" "" "" ...
##  $ end_station_name  : chr  "Desplaines St & Kinzie St" "" "" "" ...
##  $ end_station_id    : chr  "TA1306000003" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.8 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.8 ...
##  $ end_lng           : num  -87.6 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
##  $ ride_length       : 'difftime' num  0 0 0 0 ...
##   ..- attr(*, "units")= chr "secs"
##  $ day_of_week       : int  1 6 3 3 3 3 5 5 7 6 ...
##  $ date              : Date, format: "27-12-20" "18-12-20" ...
```

Convert “ride_length” from Factor to numeric so we can run calculations on the data

```
is.factor(all_trips$ride_length)

## [1] FALSE

all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
is.numeric(all_trips$ride_length)

## [1] TRUE
```

Remove “bad” data

The dataframe includes a few hundred entries when bikes were taken out of docks and checked for quality by Divvy or ride_length was negative filter the dataframe since data is being removed

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

STEP 4: CONDUCT DESCRIPTIVE ANALYSIS

Descriptive analysis on ride_length (all figures in seconds)

```
mean(all_trips$ride_length) #straight average (total ride length / rides)

## [1] 280960.9

median(all_trips$ride_length) #midpoint number in the ascending array of ride
lengths

## [1] 0

max(all_trips$ride_length) #longest ride

## [1] 915148800

min(all_trips$ride_length) #shortest ride

## [1] 0
```

You can condense the four lines above to one line using summary() on the specific attribute

```
summary(all_trips$ride_length)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##         0         0         0    280961         0 915148800
```

Compare members and casual users

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

##   all_trips$member_casual all_trips$ride_length
## 1                  casual              455382.0
## 2                  member              135781.5
```

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

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

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

```
##   all_trips$member_casual all_trips$ride_length
## 1                  casual             915148800
## 2                  member             344563200
```

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

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

See the average ride time by each day for members vs casual users

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

```
##    all_trips$member_casual all_trips$day_of_week all_trips$ride_length
## 1                   casual                     1             363922.19
## 2                   member                     1              80645.63
## 3                   casual                     2             350153.67
## 4                   member                     2              69753.46
## 5                   casual                     3             434140.60
## 6                   member                     3             271122.87
## 7                   casual                     4             331406.40
## 8                   member                     4              76160.17
## 9                   casual                     5             424813.41
## 10                  member                     5              96326.77
## 11                  casual                     6             620907.86
## 12                  member                     6             160751.97
## 13                  casual                     7             569323.39
## 14                  member                     7             180274.41
```

Notice that the days of the week are out of order. Let's fix that.

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

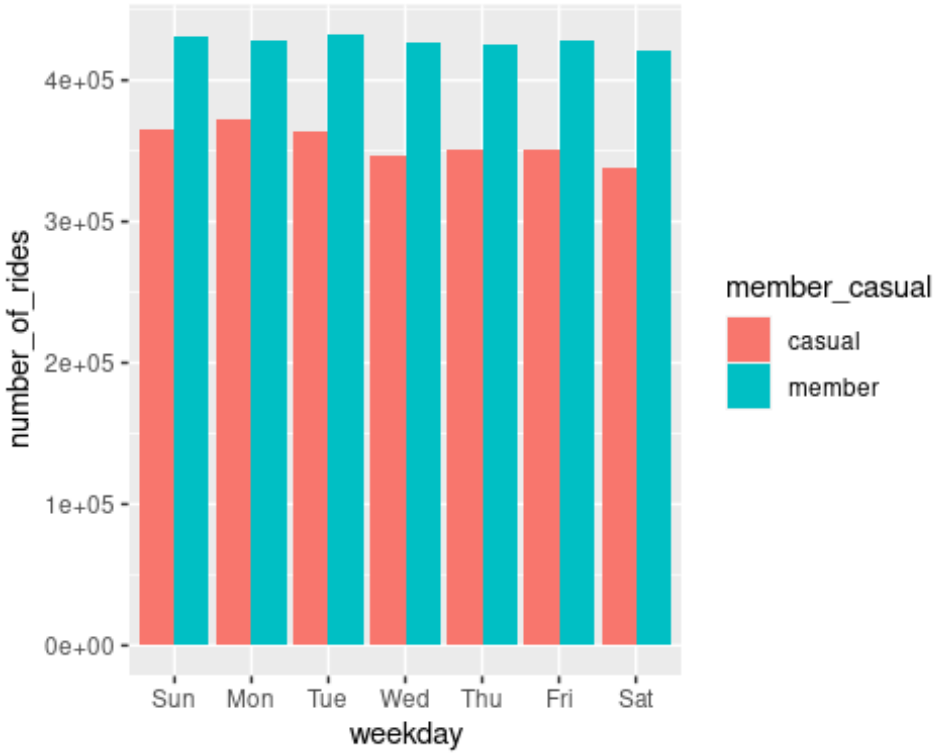
Analyze ridership data by type and weekday

```
all_trips %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday
field using wday()
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                           #calculates the
number of rides and average duration
```

```
  ,average_duration = mean(ride_length)) %>%        # calculates the average
duration
  arrange(member_casual, weekday)                                  # sorts

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

## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              365539          489055.
##  2 casual        Mon              372635          491548.
##  3 casual        Tue              363325          437391.
##  4 casual        Wed              346852          461730.
##  5 casual        Thu              351422          474137.
##  6 casual        Fri              350917          411204.
##  7 casual        Sat              337562          418237.
##  8 member        Sun              430372          320717.
##  9 member        Mon              427439          109117.
## 10 member        Tue              431841          105368.
## 11 member        Wed              426194          101579.
## 12 member        Thu              424878          106886.
## 13 member        Fri              427735           98566.
## 14 member        Sat              420972          106594.
```

Let's visualize the number of rides by rider type

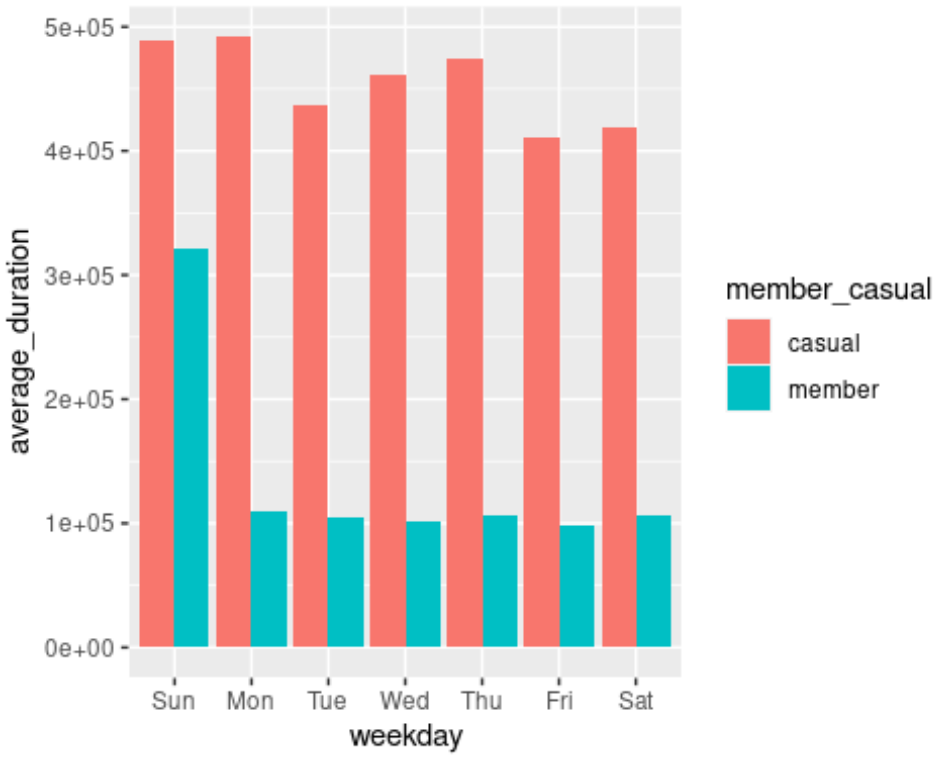
```
all_trips %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")

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



Let's create a visualization for average duration

```
all_trips %>%
  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.
```


5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
average_duration
Sun
Mon
Tue
Wed
Thu
Fri
Sat
weekday
member_casual
casual
member