Cyclistic Case Study Apr21

Hezar K

2022-11-29

This is an analysis for Cyclistic Case Study for Google Data Analytics Course. This is an analysis for April 2021.

STEP ONE: INSTALL REQUIRED PACKAGES AND IMPORT DATA

Install the required packages. **Tidyverse** package to import and wrangling the data and **ggplot2** package for visualization of the data. **Lubridate** package for date parsing and **anytime** package for the datetime conversion.

- install.packages("tidyverse")
- install.packages("ggplot2")
- install.packages("lubridate")
- install.packages("anytime")

```
library(tidyverse)
library(lubridate)
library(data.table)
library(ggplot2)
library(anytime)
```

Import data from local drive.

```
Apr21 <- read_csv("C:/Users/theby/Documents/202104-divvy-tripdata.csv")
```

STEP TWO: EXAMINE THE DATA

Examine the dataframe for an overview of the data. Review column names, **colnames()**, dimensions of the dataframe by row and column, **dim()**, the first, **head()**, and the last, **tail()**, six rows in the dataframe, the summary, **summary()**, statistics on the columns of the dataframe, and review the data type structure of columns, **str()**.

View(Apr21)

```
colnames(Apr21)
```

```
##  [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"
```

```
nrow(Apr21)
```

```
## [1] 337230
```

```
dim(Apr21)
```

```
## [1] 337230     13
```

```
head(Apr21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 6C992BD37A98A… classi… 2021-04-12 18:25:36 2021-04-12 18:56:55 State … TA1307…
## 2 1E0145613A209… docked… 2021-04-27 17:27:11 2021-04-27 18:31:29 Dorche… KA1503…
## 3 E498E15508A80… docked… 2021-04-03 12:42:45 2021-04-07 11:40:24 Loomis… 20121
## 4 1887262AD101C… classi… 2021-04-17 09:17:42 2021-04-17 09:42:48 Honore… TA1305…
## 5 C123548CAB2A3… docked… 2021-04-03 12:42:25 2021-04-03 14:13:42 Loomis… 20121
## 6 097E76F3651B1… classi… 2021-04-25 18:43:18 2021-04-25 18:43:59 Clinto… 15542
## # … with 7 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names ¹rideable_type,
## #   ²start_station_name, ³start_station_id
```

```
tail(Apr21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 6B0D434599FAC… classi… 2021-04-23 05:42:14 2021-04-23 05:48:27 Mies v… 15529
## 2 461A6B0728E06… classi… 2021-04-09 17:09:03 2021-04-09 17:16:16 Mies v… 15529
## 3 CF1D3A35E3654… docked… 2021-04-04 13:27:08 2021-04-04 14:41:11 Mies v… 15529
## 4 4308ADB9171AC… classi… 2021-04-30 18:15:40 2021-04-30 19:12:44 Mies v… 15529
## 5 04DFB53077A17… electr… 2021-04-18 11:40:37 2021-04-18 11:46:03 Mies v… 15529
## 6 DB6F78ABBECA3… classi… 2021-04-23 19:22:16 2021-04-23 19:41:07 Kedzie… 13292
## # … with 7 more variables: end_station_name <chr>, end_station_id <chr>,
## #   start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names ¹rideable_type,
## #   ²start_station_name, ³start_station_id
```

```
summary(Apr21)
```

```
##    ride_id          rideable_type        started_at
##  Length:337230      Length:337230      Min.   :2021-04-01 00:03:18.00
##  Class :character   Class :character   1st Qu.:2021-04-07 12:07:56.75
##  Mode  :character   Mode  :character   Median :2021-04-15 22:37:04.50
##                                        Mean   :2021-04-15 22:47:10.36
##                                        3rd Qu.:2021-04-24 08:31:49.50
##                                        Max.   :2021-04-30 23:59:53.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2021-04-01 00:14:29.00   Length:337230      Length:337230
##  1st Qu.:2021-04-07 12:31:51.75   Class :character   Class :character
##  Median :2021-04-15 23:00:10.00   Mode  :character   Mode  :character
##  Mean   :2021-04-15 23:11:18.80
##  3rd Qu.:2021-04-24 08:52:47.75
##  Max.   :2021-05-05 22:14:39.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:337230      Length:337230      Min.   :41.64   Min.   :-87.78
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.64
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :42.07   Max.   :-87.52
##
##     end_lat         end_lng       member_casual
##  Min.   :41.59   Min.   :-87.85   Length:337230
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.90   Median :-87.64   Mode  :character
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.15   Max.   :-87.52
##  NA's   :267     NA's   :267
```

```
str(Apr21)
```

```
## spc_tbl_ [337,230 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:337230] "6C992BD37A98A63F" "1E0145613A209000" "E498E15508A80BAD" "1887262AD101C6
04" ...
##  $ rideable_type     : chr [1:337230] "classic_bike" "docked_bike" "docked_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:337230], format: "2021-04-12 18:25:36" "2021-04-27 17:27:11" ...
##  $ ended_at          : POSIXct[1:337230], format: "2021-04-12 18:56:55" "2021-04-27 18:31:29" ...
##  $ start_station_name: chr [1:337230] "State St & Pearson St" "Dorchester Ave & 49th St" "Loomis Blvd & 84th S
t" "Honore St & Division St" ...
##  $ start_station_id  : chr [1:337230] "TA1307000061" "KA1503000069" "20121" "TA1305000034" ...
##  $ end_station_name  : chr [1:337230] "Southport Ave & Waveland Ave" "Dorchester Ave & 49th St" "Loomis Blvd &
84th St" "Southport Ave & Waveland Ave" ...
##  $ end_station_id    : chr [1:337230] "13235" "KA1503000069" "20121" "13235" ...
##  $ start_lat         : num [1:337230] 41.9 41.8 41.7 41.9 41.7 ...
##  $ start_lng         : num [1:337230] -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:337230] 41.9 41.8 41.7 41.9 41.7 ...
##  $ end_lng           : num [1:337230] -87.7 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:337230] "member" "casual" "casual" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   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>
```

Create new columns as for *date*, *month*, *day*, *year*, *day_of_week*, and *ride_length* in seconds.

```
Apr21$date <- as.Date(Apr21$started_at)
Apr21$month <- format(as.Date(Apr21$date), "%m")
Apr21$month <- month.name[as.numeric(Apr21$month)]
Apr21$day <- format(as.Date(Apr21$date), "%d")
Apr21$year <- format(as.Date(Apr21$date), "%Y")
Apr21$day_of_week <- format(as.Date(Apr21$date), "%A")
Apr21$ride_length <- difftime(Apr21$ended_at,Apr21$started_at)
```

Convert *ride_length* column to numeric in order to run calculations on the data. First, check to see if the data type is numeric, and then convert if needed.

```
is.numeric(Apr21$ride_length)
```

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

Recheck *ride_length* data type.

```
Apr21$ride_length <- as.numeric(as.character(Apr21$ride_length))
is.numeric(Apr21$ride_length)
```

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

STEP THREE: CLEAN DATA

na.omit() will remove all NA from the dataframe.

```
Apr21 <- na.omit(Apr21)
```

Remove rows with the *ride_id* column character length is not 16. This will remove all the scientific ride ids that we noticed while examining the data.

```
Apr21 <- subset(Apr21, nchar(as.character(ride_id)) == 16)
```

Remove rows with the *ride_length* less than 60 seconds or 1 minute.

```
Apr21 <- subset (Apr21, ride_length > 59)
```

STEP FOUR: ANALYZE DATA

Analyze the dataframe by find the **mean**, **median**, **max** (maximum), and **min** (minimum) of *ride_length*.

```
mean(Apr21$ride_length)
```

```
## [1] 1458.812
```

```
median(Apr21$ride_length)
```

```
## [1] 785
```

```
max(Apr21$ride_length)
```

```
## [1] 2866602
```

```
min(Apr21$ride_length)
```

```
## [1] 60
```

Run a statistical summary of the *ride_length*.

```
summary(Apr21$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      60     444     785    1459    1454 2866602
```

Compare the members and casual users

```
aggregate(Apr21$ride_length ~ Apr21$member_casual, FUN = mean)
```

```
##   Apr21$member_casual Apr21$ride_length
## 1              casual         2326.6553
## 2              member          867.7648
```

```
aggregate(Apr21$ride_length ~ Apr21$member_casual, FUN = median)
```

```
##   Apr21$member_casual Apr21$ride_length
## 1              casual              1139
## 2              member               636
```

```
aggregate(Apr21$ride_length ~ Apr21$member_casual, FUN = max)
```

```
##   Apr21$member_casual Apr21$ride_length
## 1              casual           2866602
## 2              member             87175
```

```
aggregate(Apr21$ride_length ~ Apr21$member_casual, FUN = min)
```

```
##   Apr21$member_casual Apr21$ride_length
## 1              casual                60
## 2              member                60
```

Aggregate the average ride length by each day of the week for members and users.

```
aggregate(Apr21$ride_length ~ Apr21$member_casual + Apr21$day_of_week, FUN = mean)
```

```
##    Apr21$member_casual Apr21$day_of_week Apr21$ride_length
## 1               casual            Friday         2529.4595
## 2               member            Friday          830.5253
## 3               casual            Monday         2241.6693
## 4               member            Monday          853.5002
## 5               casual          Saturday         2217.3352
## 6               member          Saturday          974.3257
## 7               casual            Sunday         2598.7935
## 8               member            Sunday          996.0700
## 9               casual          Thursday         1450.2979
## 10              member          Thursday          781.4299
## 11              casual           Tuesday         2446.6405
## 12              member           Tuesday          868.8307
## 13              casual         Wednesday         2393.5004
## 14              member         Wednesday          789.9626
```

Sort the days of the week in order.

```
Apr21$day_of_week <- ordered(Apr21$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday"))
```

Assign the aggregate the average ride length by each day of the week for members and users to x.

```
x <- aggregate(Apr21$ride_length ~ Apr21$member_casual + Apr21$day_of_week, FUN = mean)

head(x)
```

```
##   Apr21$member_casual Apr21$day_of_week Apr21$ride_length
## 1              casual            Sunday         2598.7935
## 2              member            Sunday          996.0700
## 3              casual            Monday         2241.6693
## 4              member            Monday          853.5002
## 5              casual           Tuesday         2446.6405
## 6              member           Tuesday          868.8307
```

Find the average ride length of member riders and casual riders per day and assign it to y.

```
y <- Apr21 %>%
  mutate(weekday = wday(started_at)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, weekday)

head(y)
```

```
## # A tibble: 6 × 4
##   member_casual weekday number_of_rides average_duration
##   <chr>           <int>           <int>            <dbl>
## 1 casual              1           22653            2599.
## 2 casual              2           13942            2242.
## 3 casual              3           17742            2447.
## 4 casual              4           10256            2394.
## 5 casual              5           10337            1450.
## 6 casual              6           19614            2529.
```

Analyze the dataframe to find the frequency of member riders, casual riders, classic bikes, docked bikes, and electric bikes.

```
table(Apr21$member_casual)
```

```
##
## casual member
## 119362 175261
```

```
table(Apr21$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##        211383         24568         58672
```

```
table(Apr21$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##     44740     38261     45676     32506     34628     50791     48021
```

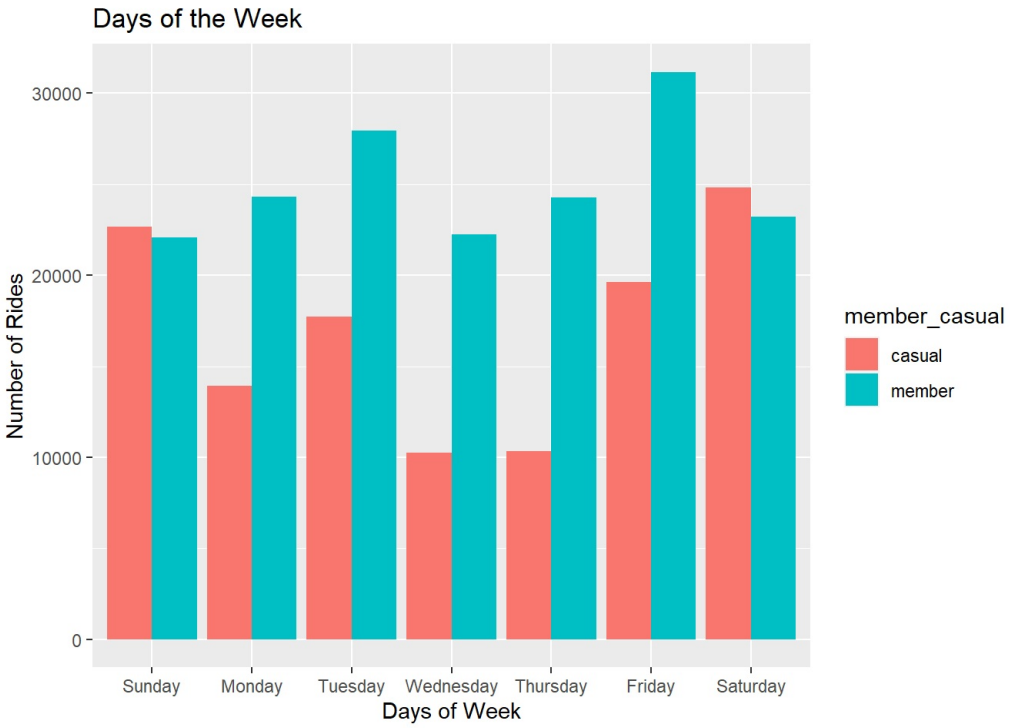
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

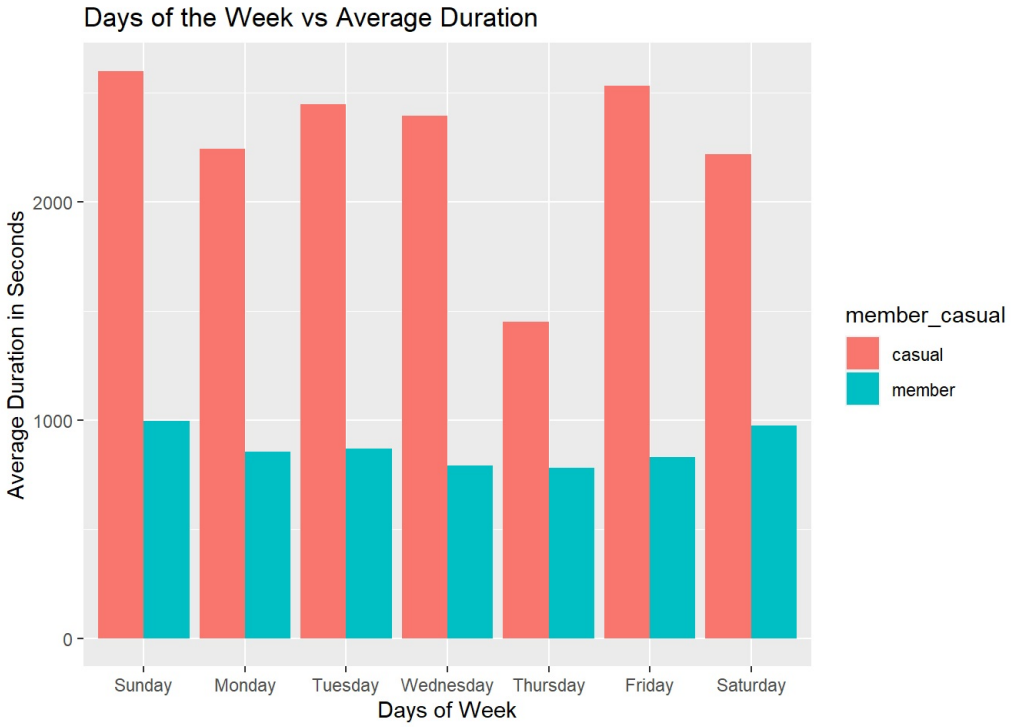
Plot the number of rides by user type during the week.

```
Apr21 %>%
  mutate(day_of_week) %>%
  group_by(member_casual,day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")+
labs(x = "Days of Week",
     y= "Number of Rides",
     title= "Days of the Week")
```



Plot the duration of the ride by user type during the week.

```
Apr21 %>%
  mutate(day_of_week) %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(x = "Days of Week",
       y= "Average Duration in Seconds",
       title= "Days of the Week vs Average Duration")
```



Create new dataframe for plots for weekday trends vs weekend trends.

```
mc<- as.data.frame(table(Apr21$day_of_week,Apr21$member_casual))
```

Rename columns

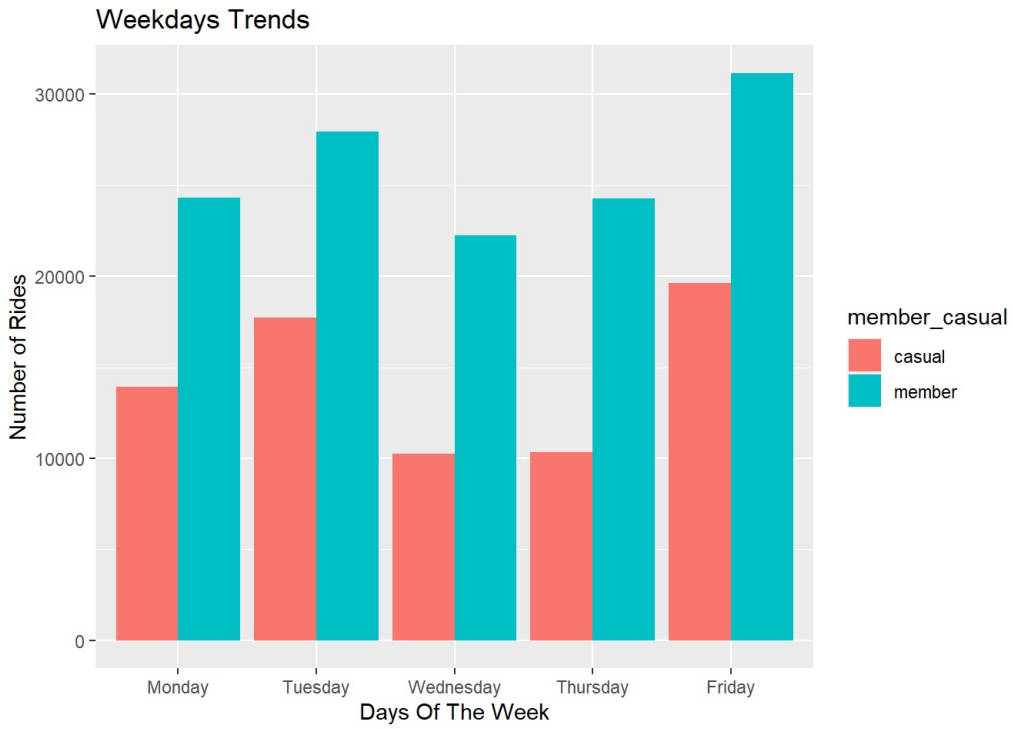
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)

head(mc)
```

```
##   day_of_week member_casual  Freq
## 1      Sunday        casual 22653
## 2      Monday        casual 13942
## 3     Tuesday        casual 17742
## 4   Wednesday        casual 10256
## 5    Thursday        casual 10337
## 6      Friday        casual 19614
```

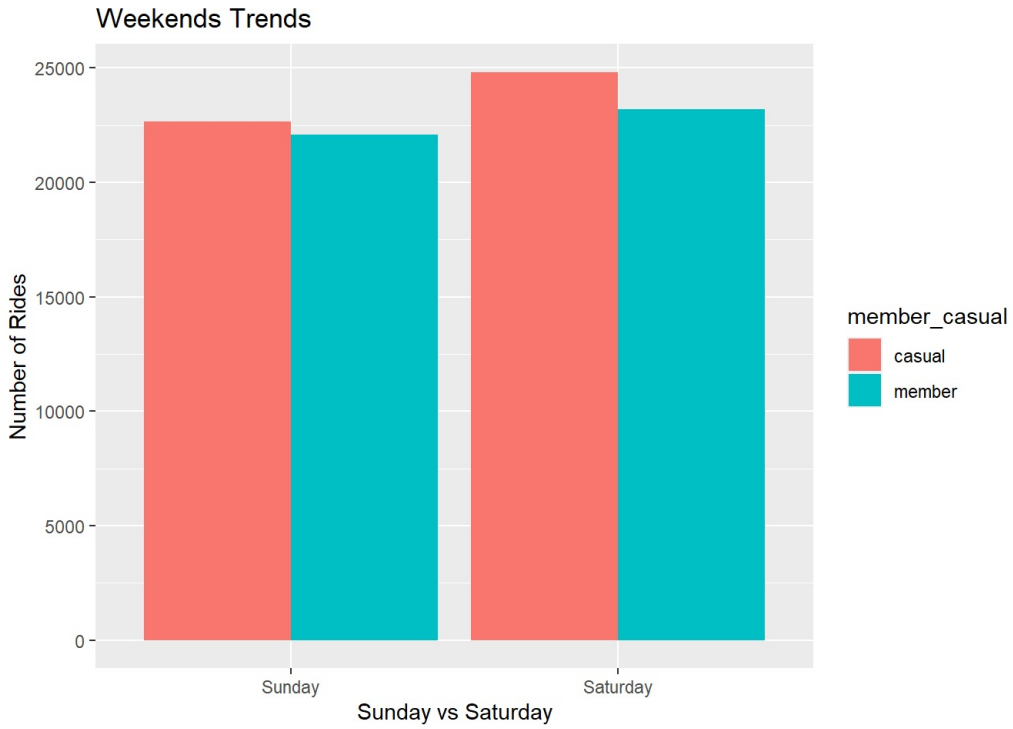
Weekday trends (Monday through Friday).

```
mc %>%
  filter(day_of_week == "Monday" |
           day_of_week == "Tuesday" |
           day_of_week == "Wednesday" |
           day_of_week == "Thursday" |
           day_of_week == "Friday") %>%
  ggplot(aes(x = day_of_week, y = Freq, fill = member_casual))+
  geom_bar(stat = "identity" , position = "dodge") +
  labs(title = "Weekdays Trends",
       x= "Days Of The Week",
       y = "Number of Rides")
```



Weekend trends (Sunday and Saturday).

```
mc %>%
  filter(day_of_week == "Sunday" |
           day_of_week == "Saturday") %>%
  ggplot(aes(x = day_of_week, y = Freq, fill = member_casual))+
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Weekends Trends",
       x= "Sunday vs Saturday",
       y = "Number of Rides")
```



Create dataframe for member and casual riders vs ride type

```
rt<- as.data.frame(table(Apr21$rideable_type,Apr21$member_casual))
```

Rename columns.

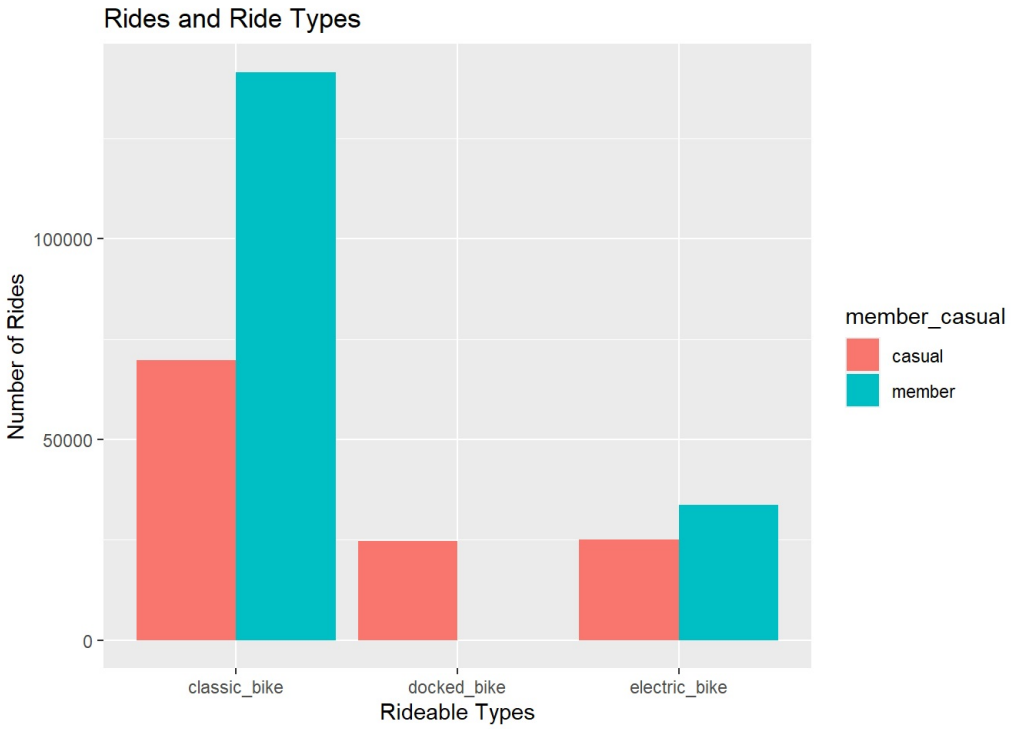
```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)

head(rt)
```

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual  69798
## 2   docked_bike        casual  24568
## 3 electric_bike        casual  24996
## 4  classic_bike        member 141585
## 5   docked_bike        member      0
## 6 electric_bike        member  33676
```

Plot for bike user vs bike type.

```
rt %>%
  filter(member_casual == "member" |
           member_casual == "casual") %>%
  ggplot(aes(x = rideable_type, y = Freq, fill = member_casual))+
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Rides and Ride Types",
       x= "Rideable Types",
       y = "Number of Rides")
```



STEP SIX: EXPORT ANALYZED DATA

Save the analyzed data as a new file.
fwrite(Apr21, “Apr21.csv”)