Cyclistic Case Study Nov21

Hezar K

2022-11-29

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

```
Nov21 <- read_csv("C:/Users/theby/Documents/202111-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(Nov21)

```
colnames(Nov21)
```

```
##  [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(Nov21)
```

```
## [1] 359978
```

```
dim(Nov21)
```

```
## [1] 359978     13
```

```
head(Nov21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 7C00A93E10556… electr… 2021-11-27 13:27:38 2021-11-27 13:46:38 <NA>    <NA>
## 2 90854840DFD50… electr… 2021-11-27 13:38:25 2021-11-27 13:56:10 <NA>    <NA>
## 3 0A7D10CDD1440… electr… 2021-11-26 22:03:34 2021-11-26 22:05:56 <NA>    <NA>
## 4 2F3BE33085BCF… electr… 2021-11-27 09:56:49 2021-11-27 10:01:50 <NA>    <NA>
## 5 D67B4781A1992… electr… 2021-11-26 19:09:28 2021-11-26 19:30:41 <NA>    <NA>
## 6 02F85C2C3C5F7… electr… 2021-11-26 18:34:07 2021-11-26 18:52:49 Michig… 13042
## # … 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(Nov21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 2E383B4D2965B… electr… 2021-11-04 16:59:24 2021-11-04 17:08:41 Cityfr… 13427
## 2 E00E9F3500D69… electr… 2021-11-29 00:39:13 2021-11-29 00:51:41 Logan … TA1308…
## 3 8EAA66CE314E5… electr… 2021-11-03 13:56:33 2021-11-03 14:01:27 Logan … TA1308…
## 4 36C2DC8BB1E13… electr… 2021-11-02 19:32:18 2021-11-02 19:36:16 Logan … TA1308…
## 5 8E42FE5C67DF6… electr… 2021-11-10 20:15:06 2021-11-10 20:22:01 Logan … TA1308…
## 6 4F15069E2D251… electr… 2021-11-30 20:18:00 2021-11-30 20:37:27 Ogden … TA1305…
## # … 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(Nov21)
```

```
##    ride_id          rideable_type        started_at
##  Length:359978      Length:359978      Min.   :2021-11-01 00:00:14.00
##  Class :character   Class :character   1st Qu.:2021-11-06 17:34:18.25
##  Mode  :character   Mode  :character   Median :2021-11-12 08:32:12.50
##                                        Mean   :2021-11-13 21:27:31.15
##                                        3rd Qu.:2021-11-20 13:39:34.00
##                                        Max.   :2021-11-30 23:59:56.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2021-11-01 00:04:06.00   Length:359978      Length:359978
##  1st Qu.:2021-11-06 17:53:19.75   Class :character   Class :character
##  Median :2021-11-12 08:46:55.50   Mode  :character   Mode  :character
##  Mean   :2021-11-13 21:42:19.90
##  3rd Qu.:2021-11-20 13:57:54.75
##  Max.   :2021-12-02 06:41:33.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:359978      Length:359978      Min.   :41.65   Min.   :-87.84
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.89   Median :-87.64
##                                        Mean   :41.89   Mean   :-87.65
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :42.07   Max.   :-87.53
##
##     end_lat         end_lng       member_casual
##  Min.   :41.39   Min.   :-88.97   Length:359978
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.89   Median :-87.64   Mode  :character
##  Mean   :41.89   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.12   Max.   :-87.53
##  NA's   :191     NA's   :191
```

```
str(Nov21)
```

```
## spc_tbl_ [359,978 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:359978] "7C00A93E10556E47" "90854840DFD508BA" "0A7D10CDD144061C" "2F3BE33085BCFF
02" ...
##  $ rideable_type     : chr [1:359978] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:359978], format: "2021-11-27 13:27:38" "2021-11-27 13:38:25" ...
##  $ ended_at          : POSIXct[1:359978], format: "2021-11-27 13:46:38" "2021-11-27 13:56:10" ...
##  $ start_station_name: chr [1:359978] NA NA NA NA ...
##  $ start_station_id  : chr [1:359978] NA NA NA NA ...
##  $ end_station_name  : chr [1:359978] NA NA NA NA ...
##  $ end_station_id    : chr [1:359978] NA NA NA NA ...
##  $ start_lat         : num [1:359978] 41.9 42 42 41.9 41.9 ...
##  $ start_lng         : num [1:359978] -87.7 -87.7 -87.7 -87.8 -87.6 ...
##  $ end_lat           : num [1:359978] 42 41.9 42 41.9 41.9 ...
##  $ end_lng           : num [1:359978] -87.7 -87.7 -87.7 -87.8 -87.6 ...
##  $ member_casual     : chr [1:359978] "casual" "casual" "casual" "casual" ...
##  - 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.

```
Nov21$date <- as.Date(Nov21$started_at)
Nov21$month <- format(as.Date(Nov21$date), "%m")
Nov21$month <- month.name[as.numeric(Nov21$month)]
Nov21$day <- format(as.Date(Nov21$date), "%d")
Nov21$year <- format(as.Date(Nov21$date), "%Y")
Nov21$day_of_week <- format(as.Date(Nov21$date), "%A")
Nov21$ride_length <- difftime(Nov21$ended_at,Nov21$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(Nov21$ride_length)
```

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

Recheck *ride_length* data type.

```
Nov21$ride_length <- as.numeric(as.character(Nov21$ride_length))
is.numeric(Nov21$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
Nov21 <- na.omit(Nov21)
```

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.

```
Nov21 <- subset(Nov21, nchar(as.character(ride_id)) == 16)
```

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

```
Nov21 <- subset (Nov21, ride_length > 59)
```

STEP FOUR: ANALYZE DATA

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

```
mean(Nov21$ride_length)
```

```
## [1] 858.2326
```

```
median(Nov21$ride_length)
```

```
## [1] 540
```

```
max(Nov21$ride_length)
```

```
## [1] 1336784
```

```
min(Nov21$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(Nov21$ride_length)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      60.0     325.0     540.0     858.2     920.0 1336784.0
```

Compare the members and casual users

```
aggregate(Nov21$ride_length ~ Nov21$member_casual, FUN = mean)
```

```
##   Nov21$member_casual Nov21$ride_length
## 1              casual         1362.1555
## 2              member          667.4124
```

```
aggregate(Nov21$ride_length ~ Nov21$member_casual, FUN = median)
```

```
##   Nov21$member_casual Nov21$ride_length
## 1              casual               721
## 2              member               485
```

```
aggregate(Nov21$ride_length ~ Nov21$member_casual, FUN = max)
```

```
##   Nov21$member_casual Nov21$ride_length
## 1              casual           1336784
## 2              member             87634
```

```
aggregate(Nov21$ride_length ~ Nov21$member_casual, FUN = min)
```

```
##   Nov21$member_casual Nov21$ride_length
## 1              casual                60
## 2              member                60
```

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

```
aggregate(Nov21$ride_length ~ Nov21$member_casual + Nov21$day_of_week, FUN = mean)
```

```
##    Nov21$member_casual Nov21$day_of_week Nov21$ride_length
## 1               casual            Friday         1307.9598
## 2               member            Friday          645.4995
## 3               casual            Monday         1484.6359
## 4               member            Monday          656.2414
## 5               casual          Saturday         1494.3539
## 6               member          Saturday          738.4392
## 7               casual            Sunday         1617.5714
## 8               member            Sunday          742.8045
## 9               casual          Thursday         1286.2071
## 10              member          Thursday          641.7016
## 11              casual           Tuesday         1070.4137
## 12              member           Tuesday          638.9888
## 13              casual         Wednesday         1104.4710
## 14              member         Wednesday          652.9258
```

Sort the days of the week in order.

```
Nov21$day_of_week <- ordered(Nov21$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(Nov21$ride_length ~ Nov21$member_casual + Nov21$day_of_week, FUN = mean)

head(x)
```

```
##   Nov21$member_casual Nov21$day_of_week Nov21$ride_length
## 1              casual            Sunday         1617.5714
## 2              member            Sunday          742.8045
## 3              casual            Monday         1484.6359
## 4              member            Monday          656.2414
## 5              casual           Tuesday         1070.4137
## 6              member           Tuesday          638.9888
```

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

```
y <- Nov21 %>%
  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           12108            1618.
## 2 casual              2            9288            1485.
## 3 casual              3           10010            1070.
## 4 casual              4            8647            1104.
## 5 casual              5            6862            1286.
## 6 casual              6            8130            1308.
```

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

```
table(Nov21$member_casual)
```

```
##
## casual member
##  69267 182922
```

```
table(Nov21$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##        151763          7515         92911
```

```
table(Nov21$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##     30963     42329     47804     37571     29119     29366     35037
```

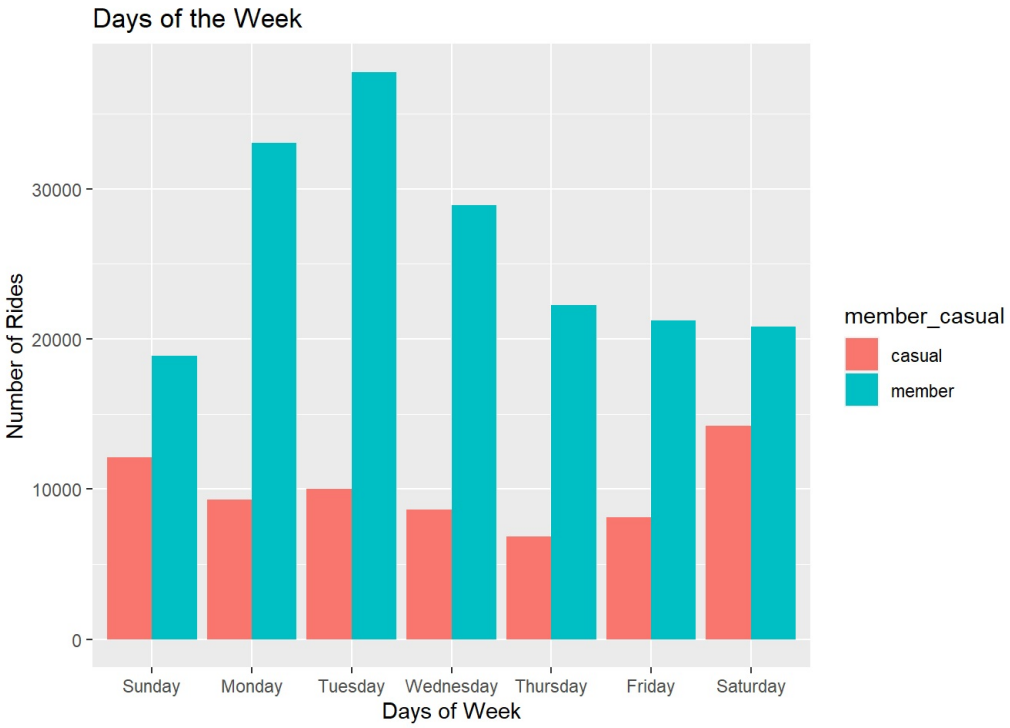
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

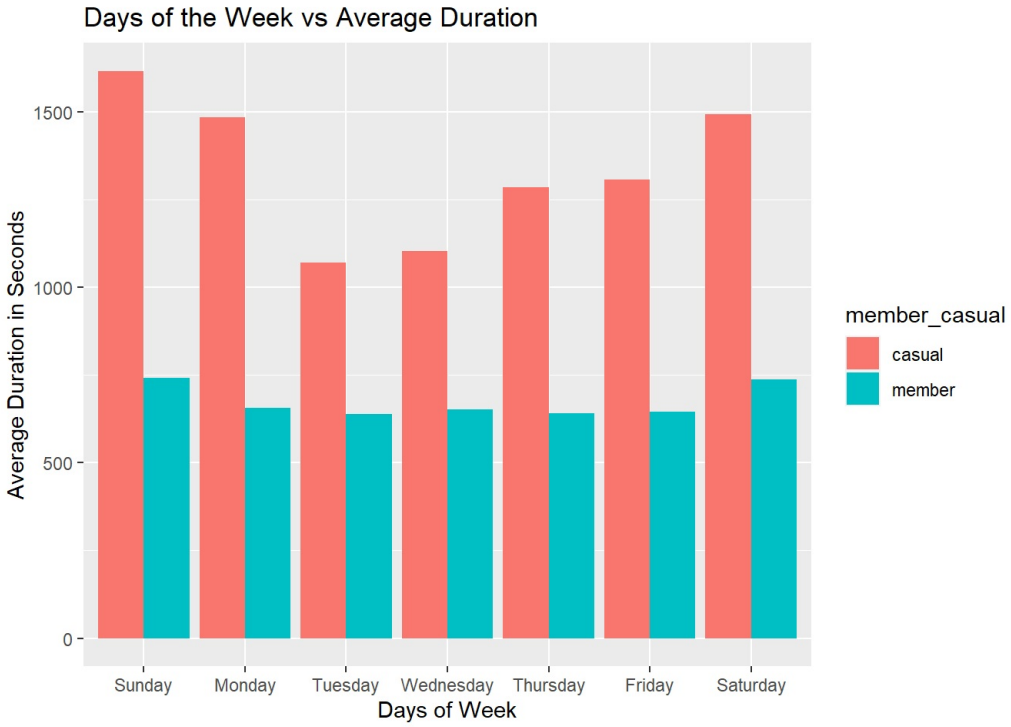
Plot the number of rides by user type during the week.

```
Nov21 %>%
  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.

```
Nov21 %>%
  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(Nov21$day_of_week,Nov21$member_casual))
```

Rename columns

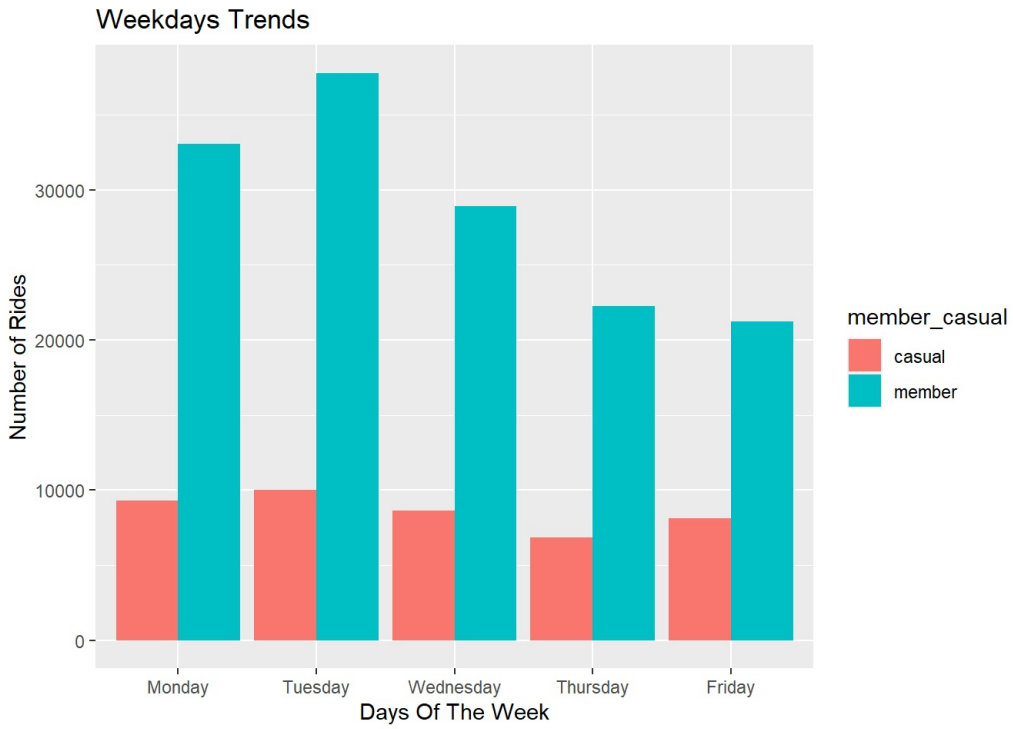
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)

head(mc)
```

```
##   day_of_week member_casual  Freq
## 1      Sunday        casual 12108
## 2      Monday        casual  9288
## 3     Tuesday        casual 10010
## 4   Wednesday        casual  8647
## 5    Thursday        casual  6862
## 6      Friday        casual  8130
```

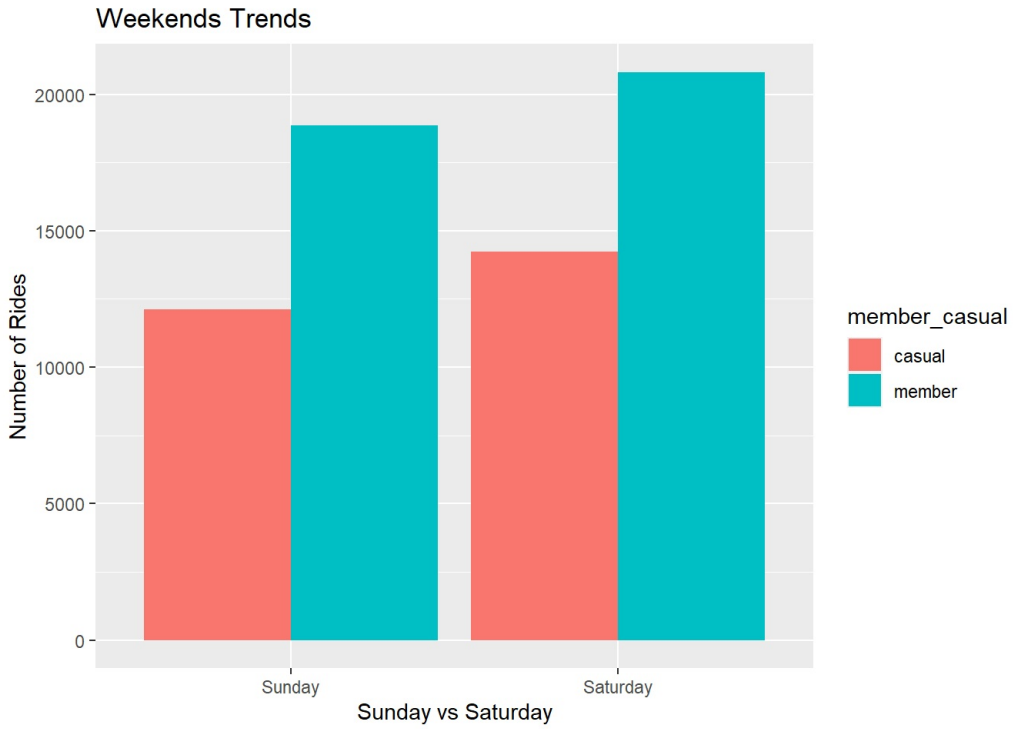
Weekday trends (Monday through Friday).

```
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(Nov21$rideable_type,Nov21$member_casual))
```

Rename columns.

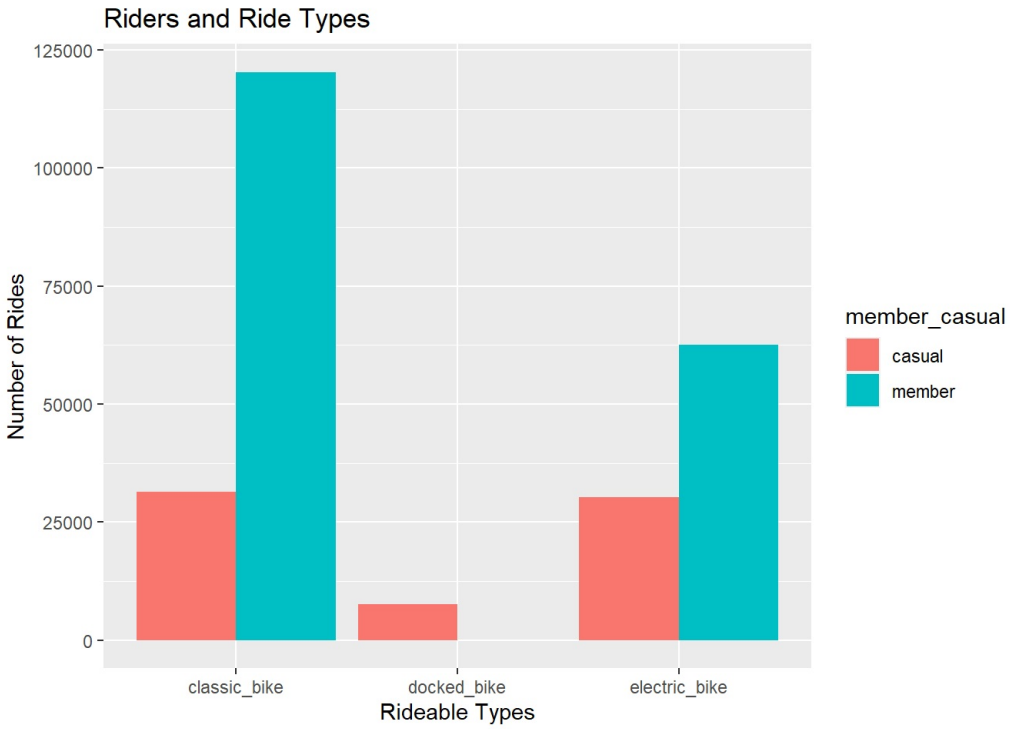
```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)

head(rt)
```

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual  31431
## 2   docked_bike        casual   7515
## 3 electric_bike        casual  30321
## 4  classic_bike        member 120332
## 5   docked_bike        member      0
## 6 electric_bike        member  62590
```

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 = "Riders and Ride Types",
       x= "Rideable Types",
       y = "Number of Rides")
```



STEP SIX: EXPORT ANALYZED DATA

Save the analyzed data as a new file. fwrite(Nov21, “Nov21.csv”)