# Cyclistic Case Study Jan21

Hezar K

2022-11-29

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

```
Jan21 <- read_csv("C:/Users/theby/Documents/202101-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(Jan21)

```
colnames(Jan21)
```

```
##  [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(Jan21)
```

```
## [1] 96834
```

```
dim(Jan21)
```

```
## [1] 96834    13
```

```
head(Jan21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 E19E6F1B8D4C4… electr… 2021-01-23 16:14:19 2021-01-23 16:24:44 Califo… 17660
## 2 DC88F20C2C55F… electr… 2021-01-27 18:43:08 2021-01-27 18:47:12 Califo… 17660
## 3 EC45C94683FE3… electr… 2021-01-21 22:35:54 2021-01-21 22:37:14 Califo… 17660
## 4 4FA453A75AE37… electr… 2021-01-07 13:31:13 2021-01-07 13:42:55 Califo… 17660
## 5 BE5E8EB4E7263… electr… 2021-01-23 02:24:02 2021-01-23 02:24:45 Califo… 17660
## 6 5D8969F88C773… electr… 2021-01-09 14:24:07 2021-01-09 15:17:54 Califo… 17660
## # … 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(Jan21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 44DE07FCDD3AD… docked… 2021-01-17 13:20:12 2021-01-17 14:15:33 Lake S… 13300
## 2 B1A5336E1412D… classi… 2021-01-19 19:03:17 2021-01-19 20:10:03 Lake S… 13300
## 3 57EA5CB7DCD75… classi… 2021-01-05 18:42:27 2021-01-05 19:33:33 Lake S… 13300
## 4 815B319A078CC… classi… 2021-01-07 17:59:47 2021-01-07 19:34:03 Lakefr… KA1504…
## 5 6DB04151565CE… classi… 2021-01-06 19:20:31 2021-01-06 20:41:57 Lakefr… KA1504…
## 6 8008C9C998083… docked… 2021-01-17 13:20:02 2021-01-17 14:17:00 Lake S… 13300
## # … 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(Jan21)
```

```
##    ride_id          rideable_type        started_at
##  Length:96834       Length:96834       Min.   :2021-01-01 00:02:05.00
##  Class :character   Class :character   1st Qu.:2021-01-08 20:55:02.75
##  Mode  :character   Mode  :character   Median :2021-01-15 06:05:04.00
##                                        Mean   :2021-01-15 17:57:29.96
##                                        3rd Qu.:2021-01-22 09:28:48.50
##                                        Max.   :2021-01-31 23:57:00.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2021-01-01 00:08:39.00   Length:96834       Length:96834
##  1st Qu.:2021-01-08 21:14:23.75   Class :character   Class :character
##  Median :2021-01-15 06:19:58.50   Mode  :character   Mode  :character
##  Mean   :2021-01-15 18:12:46.10
##  3rd Qu.:2021-01-22 09:41:18.75
##  Max.   :2021-02-01 15:33:15.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:96834       Length:96834       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   :-87.65
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :42.06   Max.   :-87.53
##
##     end_lat         end_lng       member_casual
##  Min.   :41.64   Min.   :-87.81   Length:96834
##  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.   :42.07   Max.   :-87.51
##  NA's   :103     NA's   :103
```

```
str(Jan21)
```

```
## spc_tbl_ [96,834 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:96834] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA453A75AE377D
B" ...
##  $ rideable_type     : chr [1:96834] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:96834], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" ...
##  $ ended_at          : POSIXct[1:96834], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" ...
##  $ start_station_name: chr [1:96834] "California Ave & Cortez St" "California Ave & Cortez St" "California Ave
& Cortez St" "California Ave & Cortez St" ...
##  $ start_station_id  : chr [1:96834] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:96834] NA NA NA NA ...
##  $ end_station_id    : chr [1:96834] NA NA NA NA ...
##  $ start_lat         : num [1:96834] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:96834] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:96834] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:96834] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:96834] "member" "member" "member" "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.

```
Jan21$date <- as.Date(Jan21$started_at)
Jan21$month <- format(as.Date(Jan21$date), "%m")
Jan21$month <- month.name[as.numeric(Jan21$month)]
Jan21$day <- format(as.Date(Jan21$date), "%d")
Jan21$year <- format(as.Date(Jan21$date), "%Y")
Jan21$day_of_week <- format(as.Date(Jan21$date), "%A")
Jan21$ride_length <- difftime(Jan21$ended_at,Jan21$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(Jan21$ride_length)
```

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

Recheck *ride_length* data type.

```
Jan21$ride_length <- as.numeric(as.character(Jan21$ride_length))
is.numeric(Jan21$ride_length)
```

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

### **STEP THREE:** CLEAN DATA

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

```
Jan21 <- na.omit(Jan21)
```

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.

```
Jan21 <- subset(Jan21, nchar(as.character(ride_id)) == 16)
```

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

```
Jan21 <- subset (Jan21, ride_length > 59)
```

### **STEP FOUR:** ANALYZE DATA

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

```
mean(Jan21$ride_length)
```

```
## [1] 882.3271
```

```
median(Jan21$ride_length)
```

```
## [1] 566
```

```
max(Jan21$ride_length)
```

```
## [1] 1189555
```

```
min(Jan21$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(Jan21$ride_length)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      60.0     345.0     566.0     882.3     980.0 1189555.0
```

Compare the members and casual users

```
aggregate(Jan21$ride_length ~ Jan21$member_casual, FUN = mean)
```

```
##   Jan21$member_casual Jan21$ride_length
## 1              casual         1593.6252
## 2              member          729.8724
```

```
aggregate(Jan21$ride_length ~ Jan21$member_casual, FUN = median)
```

```
##   Jan21$member_casual Jan21$ride_length
## 1              casual               766
## 2              member               531
```

```
aggregate(Jan21$ride_length ~ Jan21$member_casual, FUN = max)
```

```
##   Jan21$member_casual Jan21$ride_length
## 1              casual           1189555
## 2              member             73601
```

```
aggregate(Jan21$ride_length ~ Jan21$member_casual, FUN = min)
```

```
##   Jan21$member_casual Jan21$ride_length
## 1              casual                60
## 2              member                60
```

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

```
aggregate(Jan21$ride_length ~ Jan21$member_casual + Jan21$day_of_week, FUN = mean)
```

```
##    Jan21$member_casual Jan21$day_of_week Jan21$ride_length
## 1               casual            Friday         1427.2133
## 2               member            Friday          711.6515
## 3               casual            Monday         1199.5780
## 4               member            Monday          690.1516
## 5               casual          Saturday         2006.9722
## 6               member          Saturday          795.2698
## 7               casual            Sunday         1860.1673
## 8               member            Sunday          786.4764
## 9               casual          Thursday         1232.9294
## 10              member          Thursday          699.1626
## 11              casual           Tuesday         1399.1814
## 12              member           Tuesday          701.6795
## 13              casual         Wednesday         1576.5839
## 14              member         Wednesday          734.3590
```

Sort the days of the week in order.

```
Jan21$day_of_week <- ordered(Jan21$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(Jan21$ride_length ~ Jan21$member_casual + Jan21$day_of_week, FUN = mean)

head(x)
```

```
##   Jan21$member_casual Jan21$day_of_week Jan21$ride_length
## 1              casual            Sunday         1860.1673
## 2              member            Sunday          786.4764
## 3              casual            Monday         1199.5780
## 4              member            Monday          690.1516
## 5              casual           Tuesday         1399.1814
## 6              member           Tuesday          701.6795
```

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

```
y <- Jan21 %>%
  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            2355            1860.
## 2 casual              2            1654            1200.
## 3 casual              3            1472            1399.
## 4 casual              4            1663            1577.
## 5 casual              5            1885            1233.
## 6 casual              6            2203            1427.
```

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

```
table(Jan21$member_casual)
```

```
##
## casual member
##  14583  68039
```

```
table(Jan21$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##         60763          2085         19774
```

```
table(Jan21$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##      9969     11304     10685     11437     12320     13059     13848
```

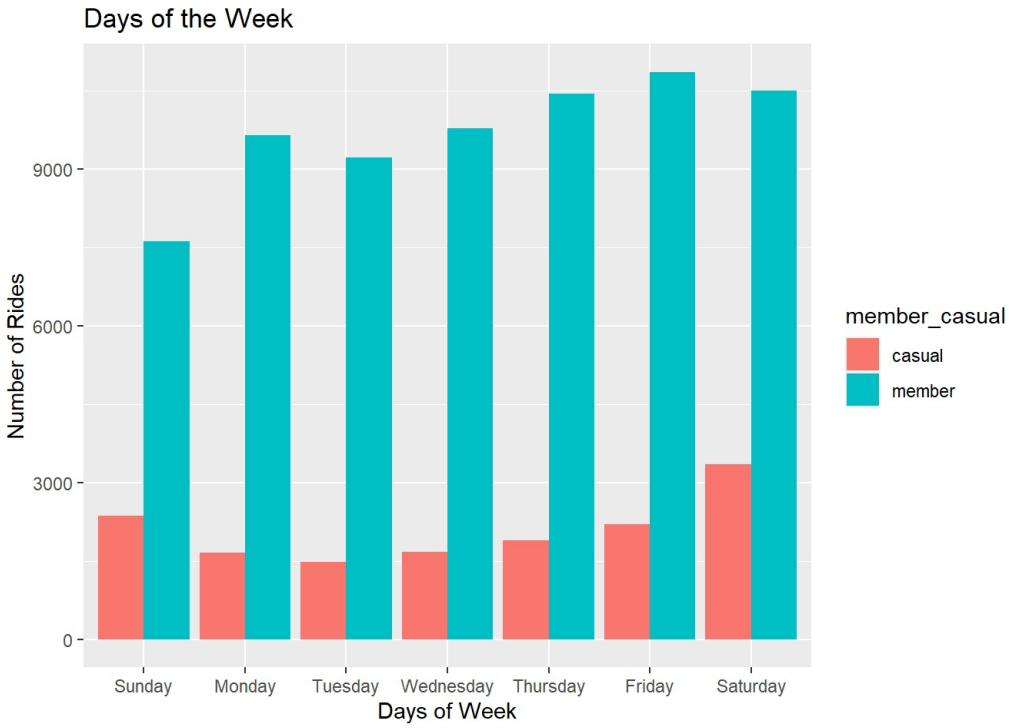
**STEP FIVE:** VISUALIZATION

Display full digits instead of scientific number.

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

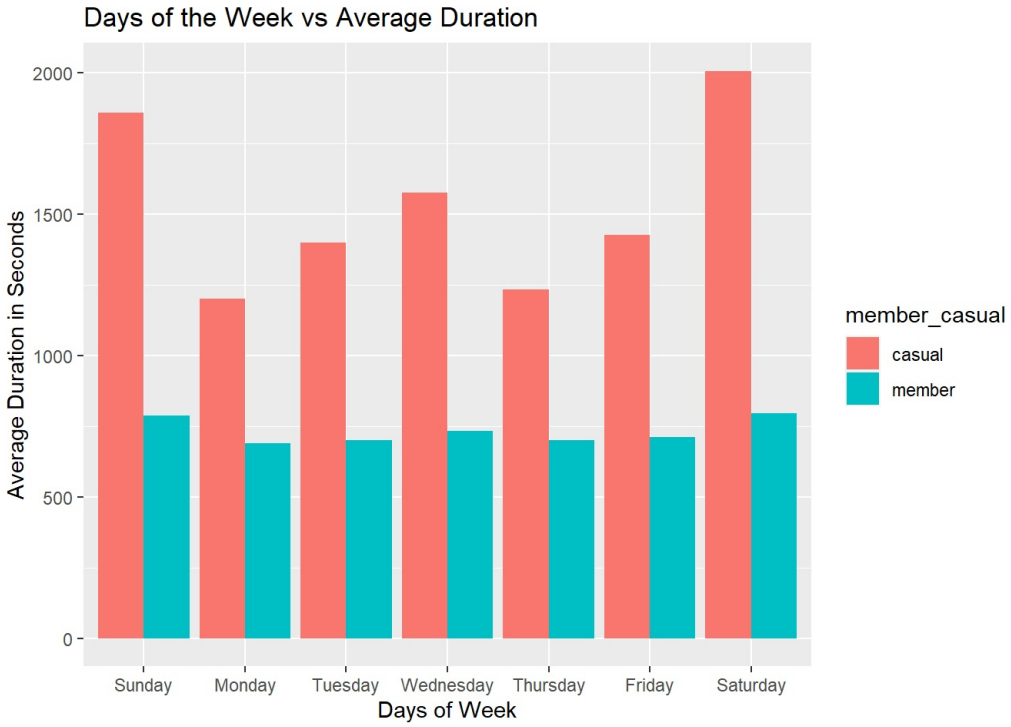
Plot the number of rides by user type during the week.

```
Jan21 %>%
  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.

```
Jan21 %>%
  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(Jan21$day_of_week,Jan21$member_casual))
```

Rename columns

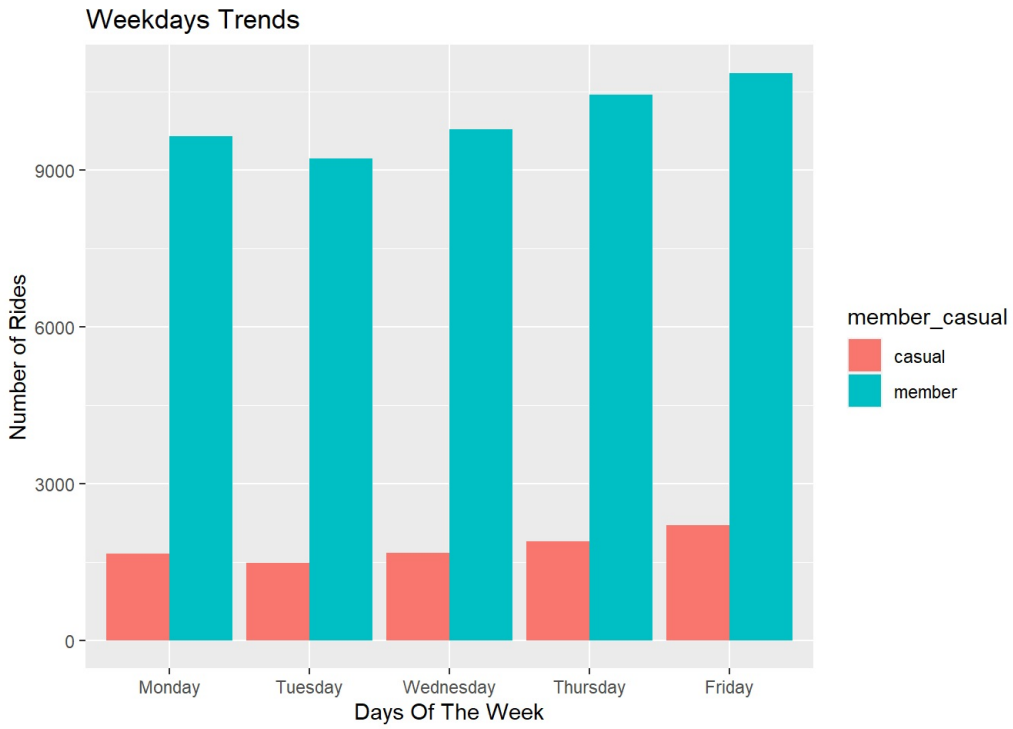
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)

head(mc)
```

```
##   day_of_week member_casual Freq
## 1      Sunday        casual 2355
## 2      Monday        casual 1654
## 3     Tuesday        casual 1472
## 4   Wednesday        casual 1663
## 5    Thursday        casual 1885
## 6      Friday        casual 2203
```

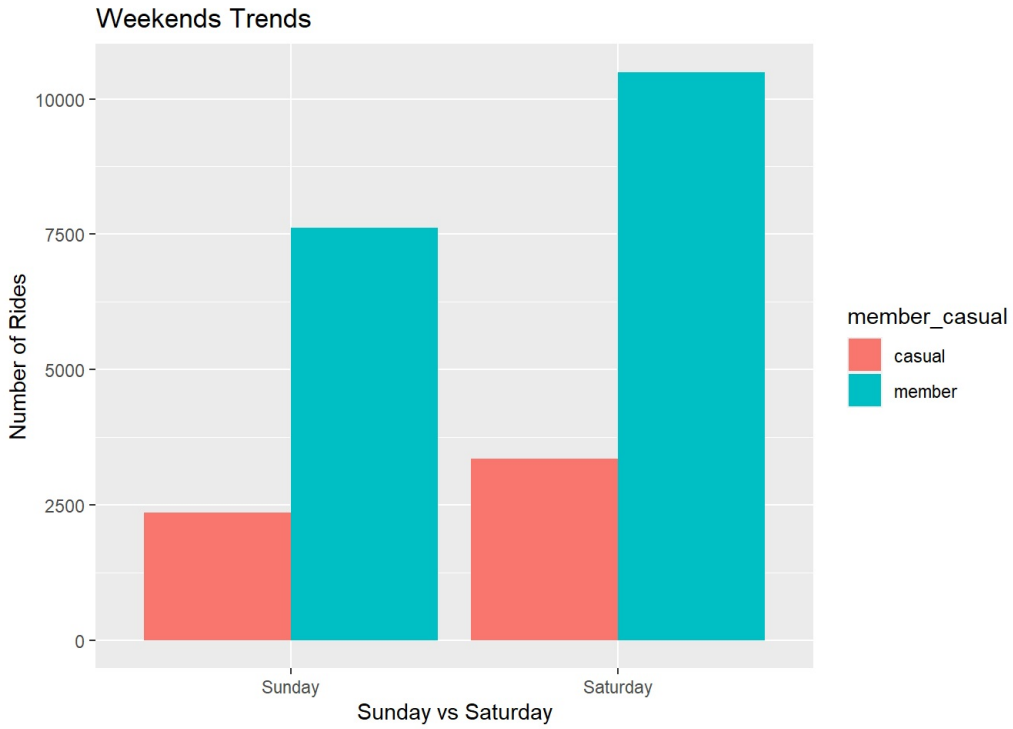
Weekday trends (Monday through Friday).

```
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(Jan21$rideable_type,Jan21$member_casual))
```

Rename columns.

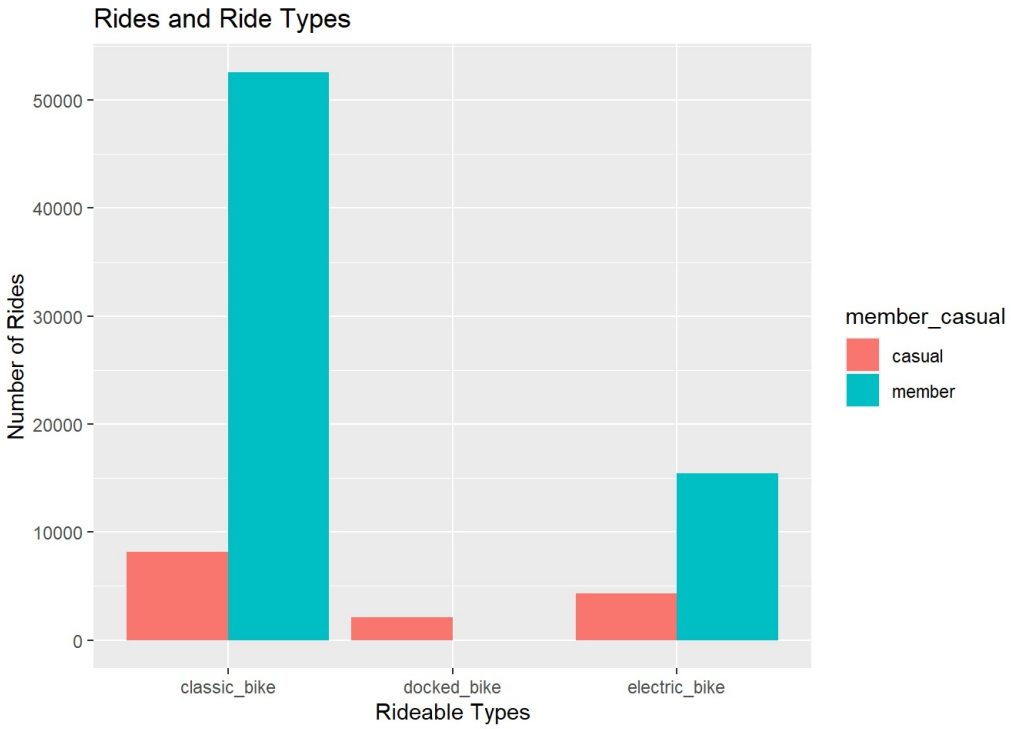
```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)

head(rt)
```

```
##   rideable_type member_casual  Freq
## 1  classic_bike        casual  8164
## 2   docked_bike        casual  2084
## 3 electric_bike        casual  4335
## 4  classic_bike        member 52599
## 5   docked_bike        member     1
## 6 electric_bike        member 15439
```

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(Jan21, “Jan21.csv”)