Cyclistic Case Study Dec21

Hezar K

2022-11-29

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

```
Dec21 <- read_csv("C:/Users/theby/Documents/202112-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(Dec21)

```
colnames(Dec21)
```

```
##  [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(Dec21)
```

```
## [1] 247540
```

```
dim(Dec21)
```

```
## [1] 247540     13
```

```
head(Dec21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 46F8167220E44… electr… 2021-12-07 15:06:07 2021-12-07 15:13:42 Laflin… 13307
## 2 73A77762838B3… electr… 2021-12-11 03:43:29 2021-12-11 04:10:23 LaSall… KP1705…
## 3 4CF42452054F5… electr… 2021-12-15 23:10:28 2021-12-15 23:23:14 Halste… KA1504…
## 4 3278BA87BF698… classi… 2021-12-26 16:16:10 2021-12-26 16:30:53 Halste… KA1504…
## 5 6FF54232576A3… electr… 2021-12-30 11:31:05 2021-12-30 11:51:21 Leavit… 18058
## 6 93E8D79490E3A… classi… 2021-12-01 18:28:36 2021-12-01 18:38:03 Wabash… SL-012
## # … 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(Dec21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 92BBAB97D1683… electr… 2021-12-24 15:42:09 2021-12-24 19:29:35 Canal … 13341
## 2 847431F3D5353… electr… 2021-12-12 13:36:55 2021-12-12 13:56:08 Canal … 13341
## 3 CF407BBC3B9FA… electr… 2021-12-06 19:37:50 2021-12-06 19:44:51 Canal … 13341
## 4 60BB69EBF5440… electr… 2021-12-02 08:57:04 2021-12-02 09:05:21 Canal … 13341
## 5 C414F654A2863… electr… 2021-12-13 09:00:26 2021-12-13 09:14:39 Lawnda… 362.0
## 6 37AC57E34B2E7… classi… 2021-12-13 08:45:32 2021-12-13 08:49:09 Michig… TA1309…
## # … 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(Dec21)
```

```
##    ride_id          rideable_type        started_at
##  Length:247540      Length:247540      Min.   :2021-12-01 00:00:01.00
##  Class :character   Class :character   1st Qu.:2021-12-06 12:51:05.25
##  Mode  :character   Mode  :character   Median :2021-12-13 13:04:54.50
##                                        Mean   :2021-12-13 23:39:29.21
##                                        3rd Qu.:2021-12-20 10:14:01.00
##                                        Max.   :2021-12-31 23:59:48.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2021-12-01 00:02:40.00   Length:247540      Length:247540
##  1st Qu.:2021-12-06 13:02:03.50   Class :character   Class :character
##  Median :2021-12-13 13:18:39.00   Mode  :character   Mode  :character
##  Mean   :2021-12-13 23:54:00.61
##  3rd Qu.:2021-12-20 10:24:38.25
##  Max.   :2022-01-03 17:32:18.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:247540      Length:247540      Min.   :41.64   Min.   :-87.84
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.67
##  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.   :42.07   Max.   :-87.52
##
##     end_lat         end_lng       member_casual
##  Min.   :41.48   Min.   :-87.85   Length:247540
##  1st Qu.:41.88   1st Qu.:-87.67   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.   :-87.52
##  NA's   :144     NA's   :144
```

```
str(Dec21)
```

```
## spc_tbl_ [247,540 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:247540] "46F8167220E4431F" "73A77762838B32FD" "4CF42452054F59C5" "3278BA87BF6983
39" ...
##  $ rideable_type     : chr [1:247540] "electric_bike" "electric_bike" "electric_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:247540], format: "2021-12-07 15:06:07" "2021-12-11 03:43:29" ...
##  $ ended_at          : POSIXct[1:247540], format: "2021-12-07 15:13:42" "2021-12-11 04:10:23" ...
##  $ start_station_name: chr [1:247540] "Laflin St & Cullerton St" "LaSalle Dr & Huron St" "Halsted St & North B
ranch St" "Halsted St & North Branch St" ...
##  $ start_station_id  : chr [1:247540] "13307" "KP1705001026" "KA1504000117" "KA1504000117" ...
##  $ end_station_name  : chr [1:247540] "Morgan St & Polk St" "Clarendon Ave & Leland Ave" "Broadway & Barry Ave
" "LaSalle Dr & Huron St" ...
##  $ end_station_id    : chr [1:247540] "TA1307000130" "TA1307000119" "13137" "KP1705001026" ...
##  $ start_lat         : num [1:247540] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:247540] -87.7 -87.6 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num [1:247540] 41.9 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:247540] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:247540] "member" "casual" "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.

```
Dec21$date <- as.Date(Dec21$started_at)
Dec21$month <- format(as.Date(Dec21$date), "%m")
Dec21$month <- month.name[as.numeric(Dec21$month)]
Dec21$day <- format(as.Date(Dec21$date), "%d")
Dec21$year <- format(as.Date(Dec21$date), "%Y")
Dec21$day_of_week <- format(as.Date(Dec21$date), "%A")
Dec21$ride_length <- difftime(Dec21$ended_at,Dec21$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(Dec21$ride_length)
```

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

Recheck *ride_length* data type.

```
Dec21$ride_length <- as.numeric(as.character(Dec21$ride_length))
is.numeric(Dec21$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
Dec21 <- na.omit(Dec21)
```

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.

```
Dec21 <- subset(Dec21, nchar(as.character(ride_id)) == 16)
```

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

```
Dec21 <- subset (Dec21, ride_length > 59)
```

STEP FOUR: ANALYZE DATA

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

```
mean(Dec21$ride_length)
```

```
## [1] 864.8466
```

```
median(Dec21$ride_length)
```

```
## [1] 523
```

```
max(Dec21$ride_length)
```

```
## [1] 1824033
```

```
min(Dec21$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(Dec21$ride_length)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      60.0     316.0     523.0     864.8     887.0 1824033.0
```

Compare the members and casual users

```
aggregate(Dec21$ride_length ~ Dec21$member_casual, FUN = mean)
```

```
##   Dec21$member_casual Dec21$ride_length
## 1              casual         1503.4362
## 2              member          644.1961
```

```
aggregate(Dec21$ride_length ~ Dec21$member_casual, FUN = median)
```

```
##   Dec21$member_casual Dec21$ride_length
## 1              casual               700
## 2              member               475
```

```
aggregate(Dec21$ride_length ~ Dec21$member_casual, FUN = max)
```

```
##   Dec21$member_casual Dec21$ride_length
## 1              casual           1824033
## 2              member             73852
```

```
aggregate(Dec21$ride_length ~ Dec21$member_casual, FUN = min)
```

```
##   Dec21$member_casual Dec21$ride_length
## 1              casual                60
## 2              member                60
```

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

```
aggregate(Dec21$ride_length ~ Dec21$member_casual + Dec21$day_of_week, FUN = mean)
```

```
##    Dec21$member_casual Dec21$day_of_week Dec21$ride_length
## 1               casual            Friday         1366.2397
## 2               member            Friday          649.4601
## 3               casual            Monday         1336.0252
## 4               member            Monday          621.6280
## 5               casual          Saturday         1410.7997
## 6               member          Saturday          699.6805
## 7               casual            Sunday         1917.8171
## 8               member            Sunday          696.3255
## 9               casual          Thursday         1523.9460
## 10              member          Thursday          642.9513
## 11              casual           Tuesday         1464.3460
## 12              member           Tuesday          609.6207
## 13              casual         Wednesday         1557.9895
## 14              member         Wednesday          627.0149
```

Sort the days of the week in order.

```
Dec21$day_of_week <- ordered(Dec21$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(Dec21$ride_length ~ Dec21$member_casual + Dec21$day_of_week, FUN = mean)

head(x)
```

```
##   Dec21$member_casual Dec21$day_of_week Dec21$ride_length
## 1              casual            Sunday         1917.8171
## 2              member            Sunday          696.3255
## 3              casual            Monday         1336.0252
## 4              member            Monday          621.6280
## 5              casual           Tuesday         1464.3460
## 6              member           Tuesday          609.6207
```

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

```
y <- Dec21 %>%
  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            5490            1918.
## 2 casual              2            4884            1336.
## 3 casual              3            3873            1464.
## 4 casual              4            6595            1558.
## 5 casual              5            8063            1524.
## 6 casual              6            8289            1366.
```

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

```
table(Dec21$member_casual)
```

```
##
## casual member
##  44684 129321
```

```
table(Dec21$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##         99068          4842         70095
```

```
table(Dec21$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##     16324     21714     20433     31656     33700     29366     20812
```

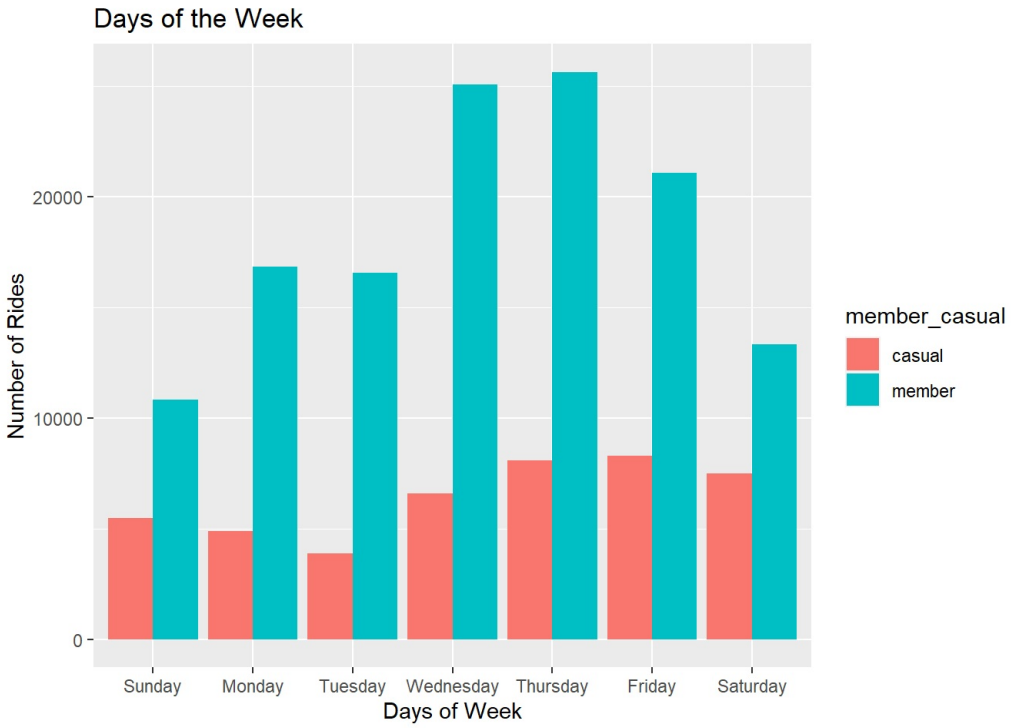
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

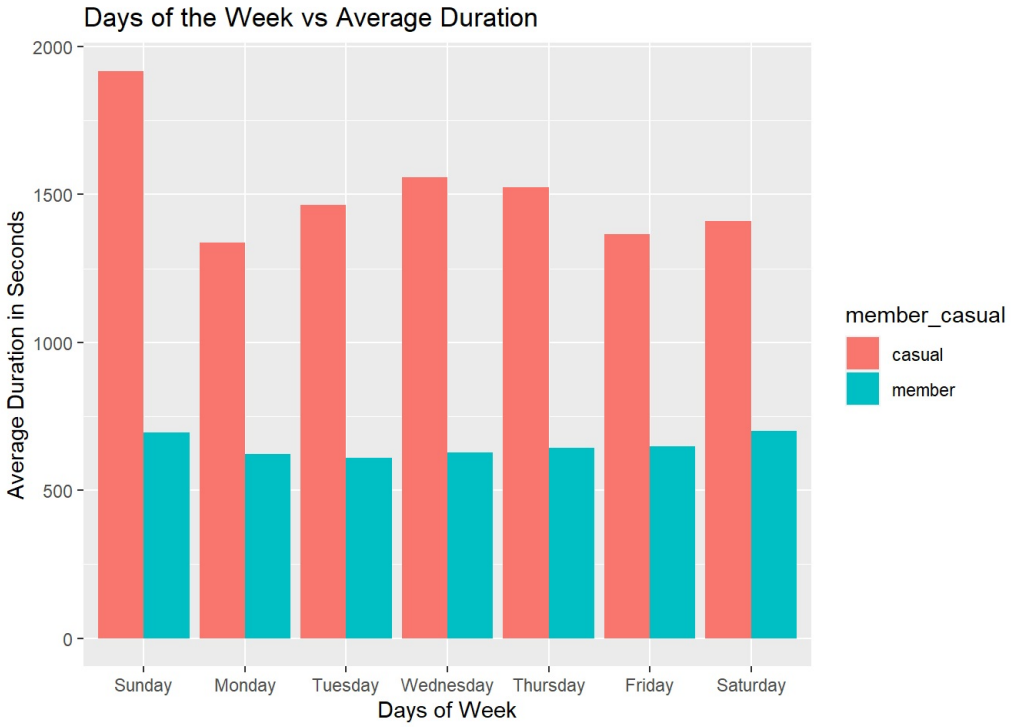
Plot the number of rides by user type during the week.

```
Dec21 %>%
  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.

```
Dec21 %>%
  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(Dec21$day_of_week,Dec21$member_casual))
```

Rename columns

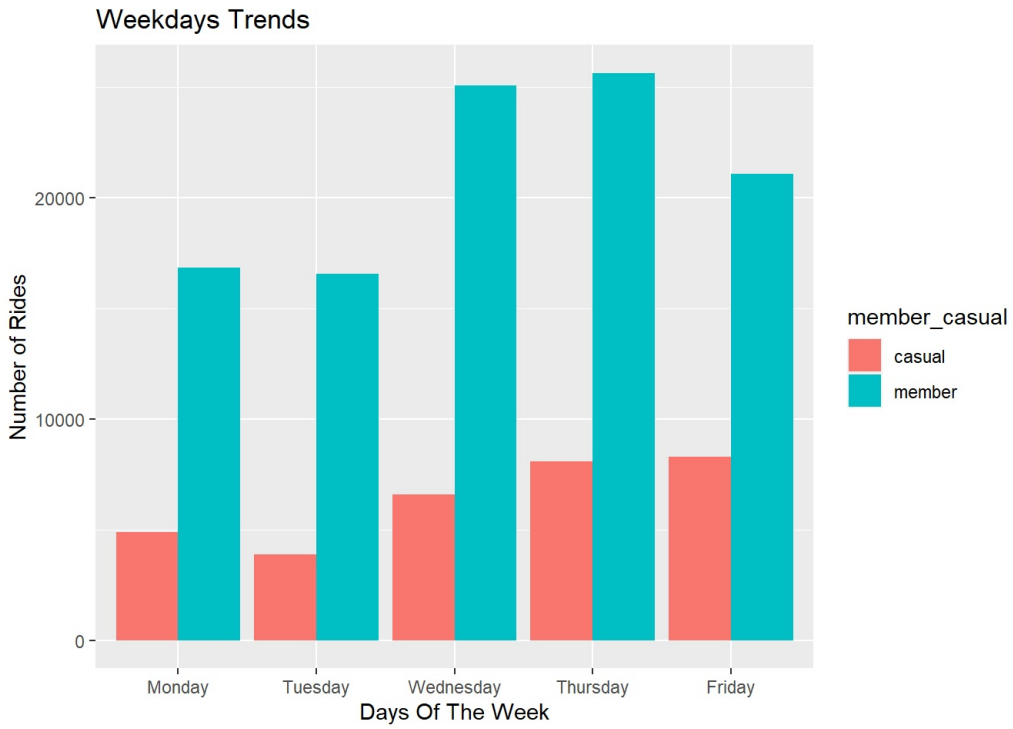
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)

head(mc)
```

```
##   day_of_week member_casual Freq
## 1      Sunday        casual 5490
## 2      Monday        casual 4884
## 3     Tuesday        casual 3873
## 4   Wednesday        casual 6595
## 5    Thursday        casual 8063
## 6      Friday        casual 8289
```

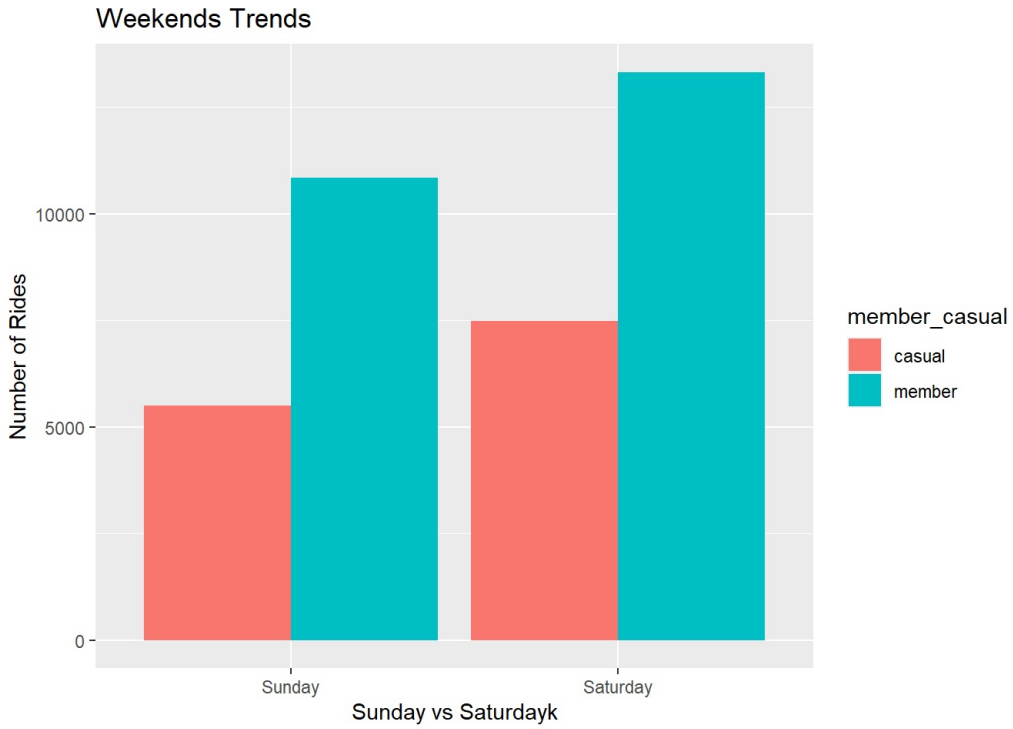
Weekday trends (Monday through Friday).

```
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 Saturdayk",
       y = "Number of Rides")
```



Create dataframe for member and casual riders vs ride type

```
rt<- as.data.frame(table(Dec21$rideable_type,Dec21$member_casual))
```

Rename columns.

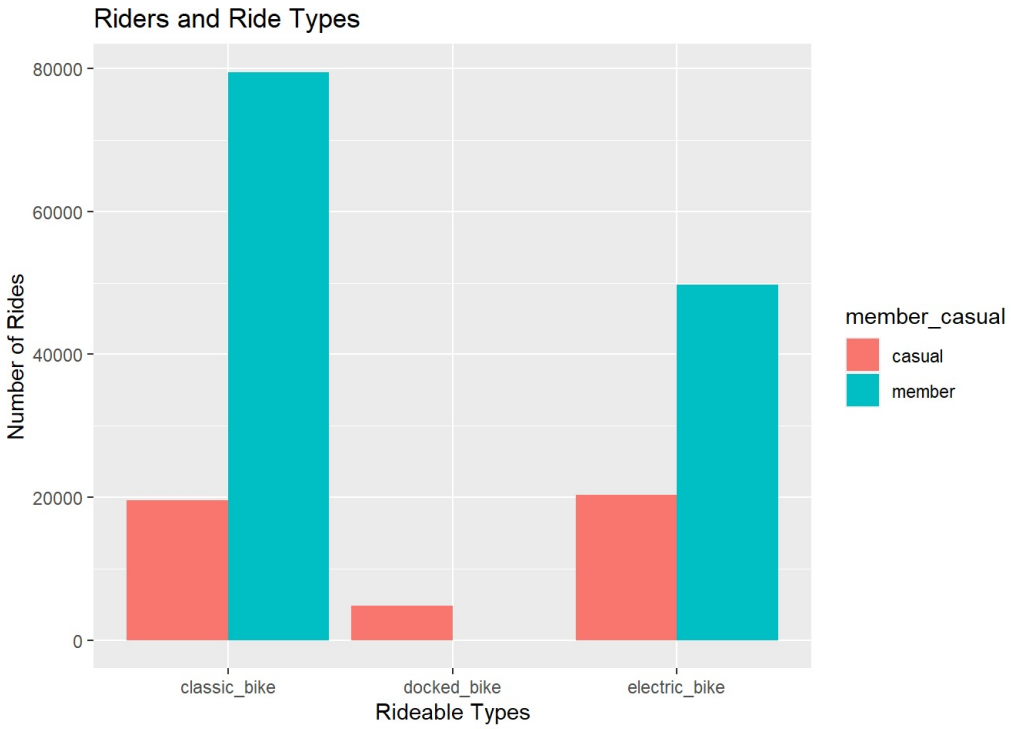
```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)

head(rt)
```

```
##   rideable_type member_casual  Freq
## 1  classic_bike        casual 19537
## 2   docked_bike        casual  4842
## 3 electric_bike        casual 20305
## 4  classic_bike        member 79531
## 5   docked_bike        member     0
## 6 electric_bike        member 49790
```

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(Dec21, “Dec21.csv”)