Cyclistic Case Study Mar21

Hezar K

2022-11-29

This is an analysis for Cyclistic Case Study for Google Data Analytics Course. This is an analysis for March 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)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.4.0      ✔ purrr   0.3.5
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(ggplot2)
library(anytime)
```

Import data from local drive.

```
Mar21 <- read_csv("C:/Users/theby/Documents/202103-divvy-tripdata.csv")
```

```
## Rows: 228496 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

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(Mar21)

```
colnames(Mar21)
```

```
##  [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(Mar21)
```

```
## [1] 228496
```

```
dim(Mar21)
```

```
## [1] 228496     13
```

```
head(Mar21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 CFA86D4455AA1… classi… 2021-03-16 08:32:30 2021-03-16 08:36:34 Humbol… 15651
## 2 30D9DC61227D1… classi… 2021-03-28 01:26:28 2021-03-28 01:36:55 Humbol… 15651
## 3 846D87A15682A… classi… 2021-03-11 21:17:29 2021-03-11 21:33:53 Shield… 15443
## 4 994D05AA75A16… classi… 2021-03-11 13:26:42 2021-03-11 13:55:41 Winthr… TA1308…
## 5 DF7464FBE92D8… classi… 2021-03-21 09:09:37 2021-03-21 09:27:33 Glenwo… 525
## 6 CEBA8516FD17F… classi… 2021-03-20 11:08:47 2021-03-20 11:29:39 Glenwo… 525
## # … 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(Mar21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 081549DEA616C… electr… 2021-03-14 01:59:38 2021-03-14 03:13:09 Larrab… TA1309…
## 2 9397BDD14798A… docked… 2021-03-20 14:58:56 2021-03-20 17:22:47 Michig… 13042
## 3 BBBEB8D51AAD4… classi… 2021-03-02 11:35:10 2021-03-02 11:43:37 Kingsb… KA1503…
## 4 637FF754DA0BD… classi… 2021-03-09 11:07:36 2021-03-09 11:49:11 Michig… 13042
## 5 F8F43A0B978A7… classi… 2021-03-01 18:11:57 2021-03-01 18:18:37 Kingsb… KA1503…
## 6 3AE64EA5BF43C… electr… 2021-03-26 17:58:14 2021-03-26 18:06:43 <NA>    <NA>
## # … 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(Mar21)
```

```
##    ride_id          rideable_type        started_at                  
##  Length:228496      Length:228496      Min.   :2021-03-01 00:01:09.00  
##  Class :character   Class :character   1st Qu.:2021-03-10 10:45:36.75  
##  Mode  :character   Mode  :character   Median :2021-03-19 17:37:20.50  
##                                        Mean   :2021-03-17 23:22:08.81  
##                                        3rd Qu.:2021-03-25 08:39:23.25  
##                                        Max.   :2021-03-31 23:59:08.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2021-03-01 00:06:28.00   Length:228496      Length:228496     
##  1st Qu.:2021-03-10 11:04:40.25   Class :character   Class :character  
##  Median :2021-03-19 17:55:05.00   Mode  :character   Mode  :character  
##  Mean   :2021-03-17 23:45:00.76                                        
##  3rd Qu.:2021-03-25 08:54:12.75                                        
##  Max.   :2021-04-06 11:00:11.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:228496      Length:228496      Min.   :41.65   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.   :-87.53  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   :41.64   Min.   :-88.07   Length:228496     
##  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.   :42.08   Max.   :-87.53                     
##  NA's   :167     NA's   :167
```

```
str(Mar21)
```

```
## spc_tbl_ [228,496 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:228496] "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05AA75A168
F2" ...
##  $ rideable_type     : chr [1:228496] "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:228496], format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ ended_at          : POSIXct[1:228496], format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ start_station_name: chr [1:228496] "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "Shields A
ve & 28th Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ start_station_id  : chr [1:228496] "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name  : chr [1:228496] "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave" "Halsted
St & 35th St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : chr [1:228496] "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat         : num [1:228496] 41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num [1:228496] -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num [1:228496] 41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num [1:228496] -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr [1:228496] "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.

```
Mar21$date <- as.Date(Mar21$started_at)
Mar21$month <- format(as.Date(Mar21$date), "%m")
Mar21$day <- format(as.Date(Mar21$date), "%d")
Mar21$year <- format(as.Date(Mar21$date), "%Y")
Mar21$day_of_week <- format(as.Date(Mar21$date), "%A")
Mar21$ride_length <- difftime(Mar21$ended_at,Mar21$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(Mar21$ride_length)
```

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

Recheck *ride_length* data type.

```
Mar21$ride_length <- as.numeric(as.character(Mar21$ride_length))
is.numeric(Mar21$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
Mar21 <- na.omit(Mar21)
```

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.

```
Mar21 <- subset(Mar21, nchar(as.character(ride_id)) == 16)
```

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

```
Mar21 <- subset (Mar21, ride_length > "1")
```

STEP FOUR: ANALYZE DATA

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

```
mean(Mar21$ride_length)
```

```
## [1] 1367.549
```

```
median(Mar21$ride_length)
```

```
## [1] 748
```

```
max(Mar21$ride_length)
```

```
## [1] 1900899
```

```
min(Mar21$ride_length)
```

```
## [1] 2
```

Run a statistical summary of the *ride_length*.

```
summary(Mar21$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2     417     748    1368    1399 1900899
```

Compare the members and casual users

```
aggregate(Mar21$ride_length ~ Mar21$member_casual, FUN = mean)
```

```
##   Mar21$member_casual Mar21$ride_length
## 1              casual         2308.8900
## 2              member          819.9872
```

```
aggregate(Mar21$ride_length ~ Mar21$member_casual, FUN = median)
```

```
##   Mar21$member_casual Mar21$ride_length
## 1              casual              1166
## 2              member               602
```

```
aggregate(Mar21$ride_length ~ Mar21$member_casual, FUN = max)
```

```
##   Mar21$member_casual Mar21$ride_length
## 1              casual           1900899
## 2              member             88022
```

```
aggregate(Mar21$ride_length ~ Mar21$member_casual, FUN = min)
```

```
##   Mar21$member_casual Mar21$ride_length
## 1              casual                 2
## 2              member                 2
```

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

```
aggregate(Mar21$ride_length ~ Mar21$member_casual + Mar21$day_of_week, FUN = mean)
```

```
##    Mar21$member_casual Mar21$day_of_week Mar21$ride_length
## 1               casual            Friday         1775.2845
## 2               member            Friday          747.6784
## 3               casual            Monday         2712.9499
## 4               member            Monday          824.0340
## 5               casual          Saturday         2529.1231
## 6               member          Saturday          930.0881
## 7               casual            Sunday         2464.6047
## 8               member            Sunday          953.4953
## 9               casual          Thursday         1805.6213
## 10              member          Thursday          709.2299
## 11              casual           Tuesday         2205.9162
## 12              member           Tuesday          796.1308
## 13              casual         Wednesday         1752.6361
## 14              member         Wednesday          754.9535
```

Sort the days of the week in order.

```
Mar21$day_of_week <- ordered(Mar21$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(Mar21$ride_length ~ Mar21$member_casual + Mar21$day_of_week, FUN = mean)

head(x)
```

```
##   Mar21$member_casual Mar21$day_of_week Mar21$ride_length
## 1              casual            Sunday         2464.6047
## 2              member            Sunday          953.4953
## 3              casual            Monday         2712.9499
## 4              member            Monday          824.0340
## 5              casual           Tuesday         2205.9162
## 6              member           Tuesday          796.1308
```

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

```
y <- Mar21 %>%
  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           15926            2465.
## 2 casual              2           10767            2713.
## 3 casual              3            9299            2206.
## 4 casual              4            7684            1753.
## 5 casual              5            4806            1806.
## 6 casual              6            6846            1775.
```

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

```
table(Mar21$member_casual)
```

```
##
## casual member
##  75639 130035
```

```
table(Mar21$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##        152149         15657         37868
```

```
table(Mar21$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##     32440     31369     30444     28149     19218     22848     41206
```

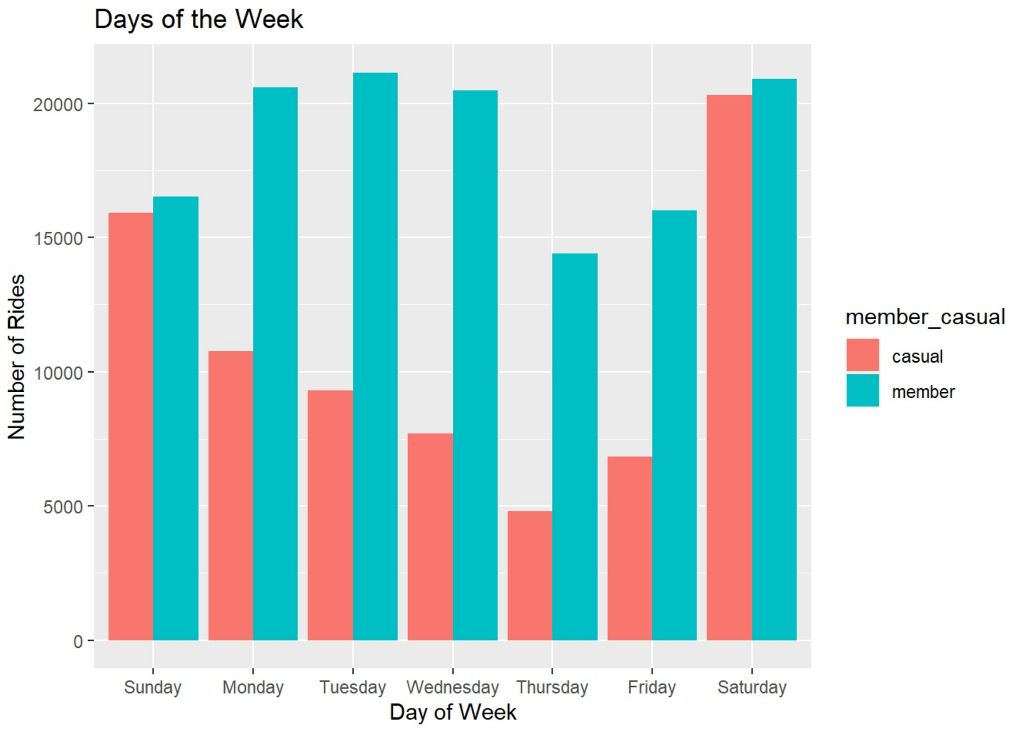
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

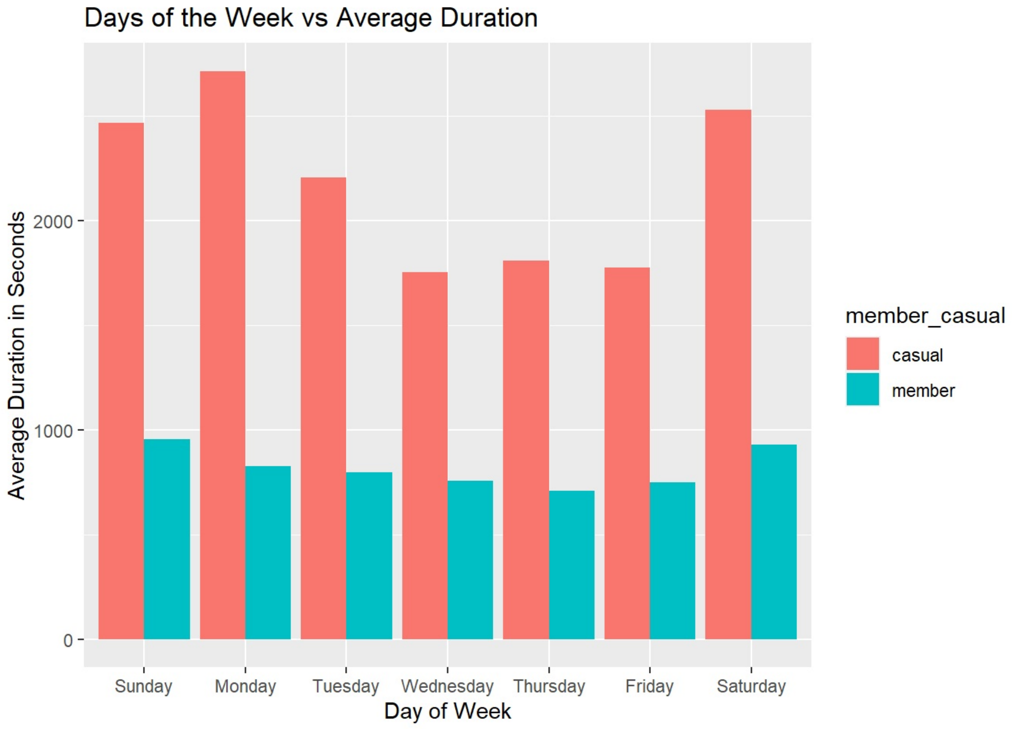
Plot the number of rides by user type during the week.

```
Mar21 %>%
  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 = "Day of Week",
     y= "Number of Rides",
     title= "Days of the Week")
```



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

```
Mar21 %>%
  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 = "Day 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(Mar21$day_of_week,Mar21$member_casual))
```

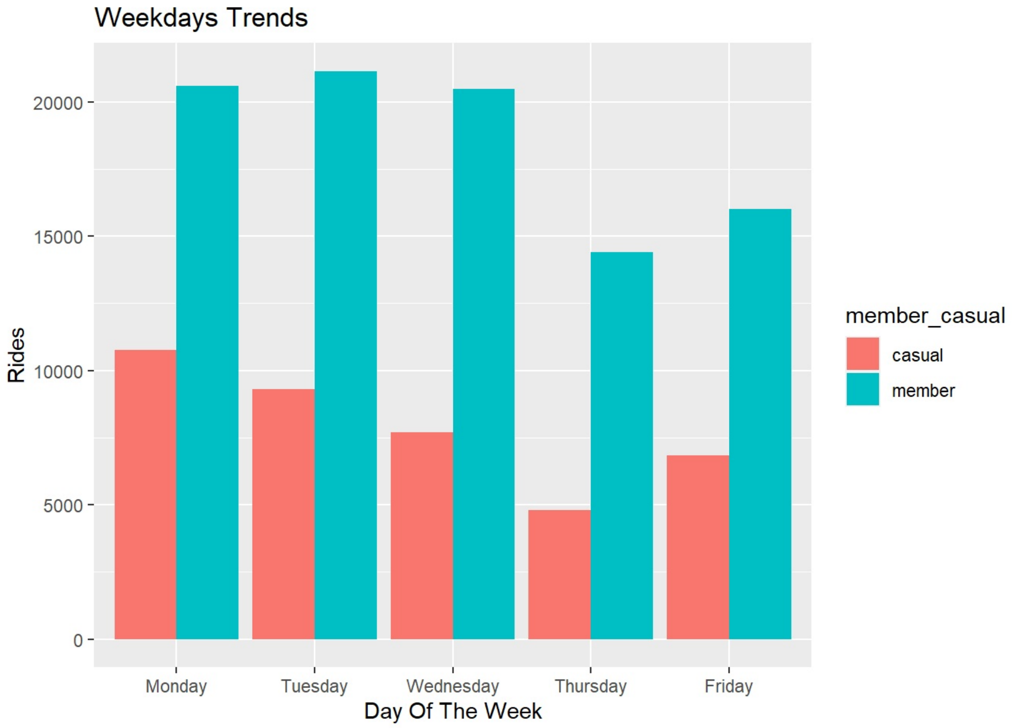
Rename columns

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

```
##   day_of_week member_casual  Freq
## 1      Sunday        casual 15926
## 2      Monday        casual 10767
## 3     Tuesday        casual  9299
## 4   Wednesday        casual  7684
## 5    Thursday        casual  4806
## 6      Friday        casual  6846
```

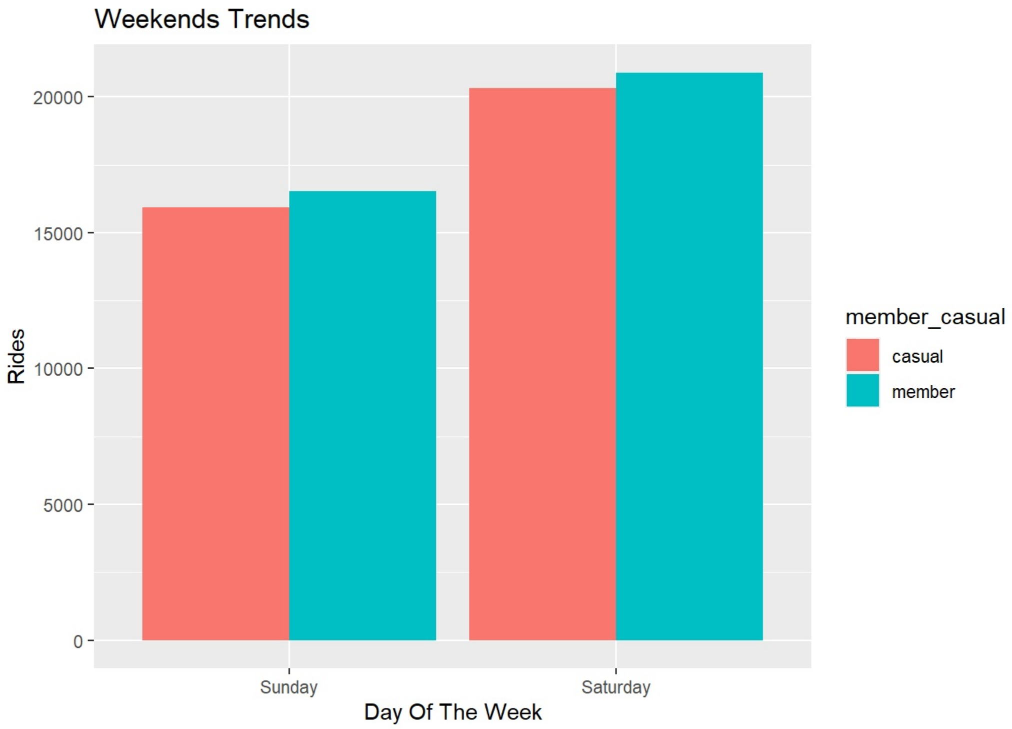
Weekday trends (Monday through Friday).

```
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= "Day Of The Week",
       y = "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= "Day Of The Week",
       y = "Rides")
```



Create dataframe for member and casual riders vs ride type

```
rt<- as.data.frame(table(Mar21$rideable_type,Mar21$member_casual))
```

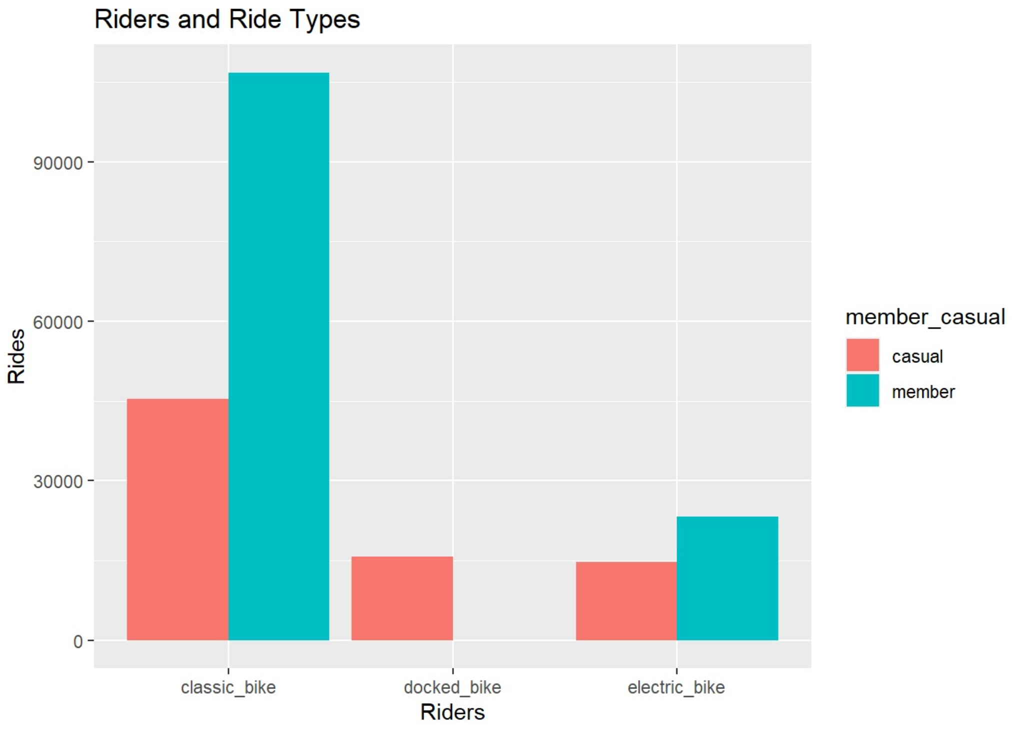
Rename columns.

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

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual  45355
## 2   docked_bike        casual  15657
## 3 electric_bike        casual  14627
## 4  classic_bike        member 106794
## 5   docked_bike        member      0
## 6 electric_bike        member  23241
```

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= "Riders",
       y = "Rides")
```



STEP SIX: EXPORT ANALYZED DATA

Save the analyzed data as a new file. fwrite(Mar21, “Mar21.csv”)