Cyclistic Case Study Aug21

Hezar K

2022-11-29

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

```
Aug21 <- read_csv("C:/Users/theby/Documents/202108-divvy-tripdata.csv")
```

```
## Rows: 804352 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(Aug21)

```
colnames(Aug21)
```

```
##  [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(Aug21)
```

```
## [1] 804352
```

```
dim(Aug21)
```

```
## [1] 804352     13
```

```
head(Aug21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 99103BB87CC6C… electr… 2021-08-10 17:15:49 2021-08-10 17:22:44 <NA>    <NA>
## 2 EAFCCCFB0A3FC… electr… 2021-08-10 17:23:14 2021-08-10 17:39:24 <NA>    <NA>
## 3 9EF4F46C57AD2… electr… 2021-08-21 02:34:23 2021-08-21 02:50:36 <NA>    <NA>
## 4 5834D3208BFAF… electr… 2021-08-21 06:52:55 2021-08-21 07:08:13 <NA>    <NA>
## 5 CD825CB87ED1D… electr… 2021-08-19 11:55:29 2021-08-19 12:04:11 <NA>    <NA>
## 6 612F12C94A964… electr… 2021-08-19 12:41:12 2021-08-19 12:47:47 <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
```

```
tail(Aug21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 2D6861BE1B674… classi… 2021-08-07 10:52:09 2021-08-07 10:58:09 Paulin… TA1305…
## 2 5E5C9CD681E04… classi… 2021-08-07 18:07:43 2021-08-07 18:21:21 Wells … TA1308…
## 3 96FB57CF4AA45… electr… 2021-08-09 08:49:31 2021-08-09 09:03:51 Broadw… 13323
## 4 226A0910DCCE9… classi… 2021-08-12 16:55:57 2021-08-12 17:15:10 Dearbo… TA1305…
## 5 1A97D27AE23DE… classi… 2021-08-08 22:47:43 2021-08-08 23:08:12 Broadw… 13323
## 6 BBC36E4AA3652… electr… 2021-08-27 18:53:53 2021-08-27 19:02:16 Paulin… 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(Aug21)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:804352      Length:804352      Min.   :2021-08-01 00:00:04.00  
##  Class :character   Class :character   1st Qu.:2021-08-08 12:06:10.75  
##  Mode  :character   Mode  :character   Median :2021-08-16 07:57:11.50  
##                                        Mean   :2021-08-16 10:44:36.11  
##                                        3rd Qu.:2021-08-23 17:33:34.75  
##                                        Max.   :2021-08-31 23:59:35.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2021-08-01 00:03:11.00   Length:804352      Length:804352     
##  1st Qu.:2021-08-08 12:30:18.75   Class :character   Class :character  
##  Median :2021-08-16 08:12:14.00   Mode  :character   Mode  :character  
##  Mean   :2021-08-16 11:06:14.23                                        
##  3rd Qu.:2021-08-23 17:52:03.75                                        
##  Max.   :2021-09-01 17:37:35.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:804352      Length:804352      Min.   :41.65   Min.   :-87.84  
##  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.   :42.07   Max.   :-87.52  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   :41.58   Min.   :-87.85   Length:804352     
##  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.   :-87.51                     
##  NA's   :706     NA's   :706
```

```
str(Aug21)
```

```
## spc_tbl_ [804,352 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:804352] "99103BB87CC6C1BB" "EAFCCCFB0A3FC5A1" "9EF4F46C57AD234D" "5834D3208BFAF1
DA" ...
##  $ rideable_type     : chr [1:804352] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:804352], format: "2021-08-10 17:15:49" "2021-08-10 17:23:14" ...
##  $ ended_at          : POSIXct[1:804352], format: "2021-08-10 17:22:44" "2021-08-10 17:39:24" ...
##  $ start_station_name: chr [1:804352] NA NA NA NA ...
##  $ start_station_id  : chr [1:804352] NA NA NA NA ...
##  $ end_station_name  : chr [1:804352] NA NA NA NA ...
##  $ end_station_id    : chr [1:804352] NA NA NA NA ...
##  $ start_lat         : num [1:804352] 41.8 41.8 42 42 41.8 ...
##  $ start_lng         : num [1:804352] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:804352] 41.8 41.8 42 42 41.8 ...
##  $ end_lng           : num [1:804352] -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:804352] "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.

```
Aug21$date <- as.Date(Aug21$started_at)
Aug21$month <- format(as.Date(Aug21$date), "%m")
Aug21$day <- format(as.Date(Aug21$date), "%d")
Aug21$year <- format(as.Date(Aug21$date), "%Y")
Aug21$day_of_week <- format(as.Date(Aug21$date), "%A")
Aug21$ride_length <- difftime(Aug21$ended_at,Aug21$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(Aug21$ride_length)
```

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

Recheck *ride_length* data type.

```
Aug21$ride_length <- as.numeric(as.character(Aug21$ride_length))
is.numeric(Aug21$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
Aug21 <- na.omit(Aug21)
```

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.

```
Aug21 <- subset(Aug21, nchar(as.character(ride_id)) == 16)
```

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

```
Aug21 <- subset (Aug21, ride_length > "1")
```

STEP FOUR: ANALYZE DATA

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

```
mean(Aug21$ride_length)
```

```
## [1] 1269.376
```

```
median(Aug21$ride_length)
```

```
## [1] 768
```

```
max(Aug21$ride_length)
```

```
## [1] 2497750
```

```
min(Aug21$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(Aug21$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       2     443     768    1269    1370 2497750
```

Compare the members and casual users

```
aggregate(Aug21$ride_length ~ Aug21$member_casual, FUN = mean)
```

```
##   Aug21$member_casual Aug21$ride_length
## 1              casual         1714.0036
## 2              member          813.2978
```

```
aggregate(Aug21$ride_length ~ Aug21$member_casual, FUN = median)
```

```
##   Aug21$member_casual Aug21$ride_length
## 1              casual               983
## 2              member               605
```

```
aggregate(Aug21$ride_length ~ Aug21$member_casual, FUN = max)
```

```
##   Aug21$member_casual Aug21$ride_length
## 1              casual           2497750
## 2              member             89183
```

```
aggregate(Aug21$ride_length ~ Aug21$member_casual, FUN = min)
```

```
##   Aug21$member_casual Aug21$ride_length
## 1              casual                 2
## 2              member                 2
```

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

```
aggregate(Aug21$ride_length ~ Aug21$member_casual + Aug21$day_of_week, FUN = mean)
```

```
##    Aug21$member_casual Aug21$day_of_week Aug21$ride_length
## 1               casual            Friday         1626.7924
## 2               member            Friday          791.3637
## 3               casual            Monday         1703.8251
## 4               member            Monday          768.7180
## 5               casual          Saturday         1810.0526
## 6               member          Saturday          926.7132
## 7               casual            Sunday         1949.0105
## 8               member            Sunday          935.4238
## 9               casual          Thursday         1539.8663
## 10              member          Thursday          773.7512
## 11              casual           Tuesday         1548.7337
## 12              member           Tuesday          749.1267
## 13              casual         Wednesday         1507.5961
## 14              member         Wednesday          764.5086
```

Sort the days of the week in order.

```
Aug21$day_of_week <- ordered(Aug21$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(Aug21$ride_length ~ Aug21$member_casual + Aug21$day_of_week, FUN = mean)

head(x)
```

```
##   Aug21$member_casual Aug21$day_of_week Aug21$ride_length
## 1              casual            Sunday         1949.0105
## 2              member            Sunday          935.4238
## 3              casual            Monday         1703.8251
## 4              member            Monday          768.7180
## 5              casual           Tuesday         1548.7337
## 6              member           Tuesday          749.1267
```

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

```
y <- Aug21 %>%
  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           73381            1949.
## 2 casual              2           39852            1704.
## 3 casual              3           37633            1549.
## 4 casual              4           32306            1508.
## 5 casual              5           38219            1540.
## 6 casual              6           48164            1627.
```

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

```
table(Aug21$member_casual)
```

```
## 
## casual member 
## 341437 332864
```

```
table(Aug21$rideable_type)
```

```
## 
##  classic_bike   docked_bike electric_bike 
##        501737         45064        127500
```

```
table(Aug21$day_of_week)
```

```
## 
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday 
##    121132     90996     91936     76929     84982     92884    115442
```

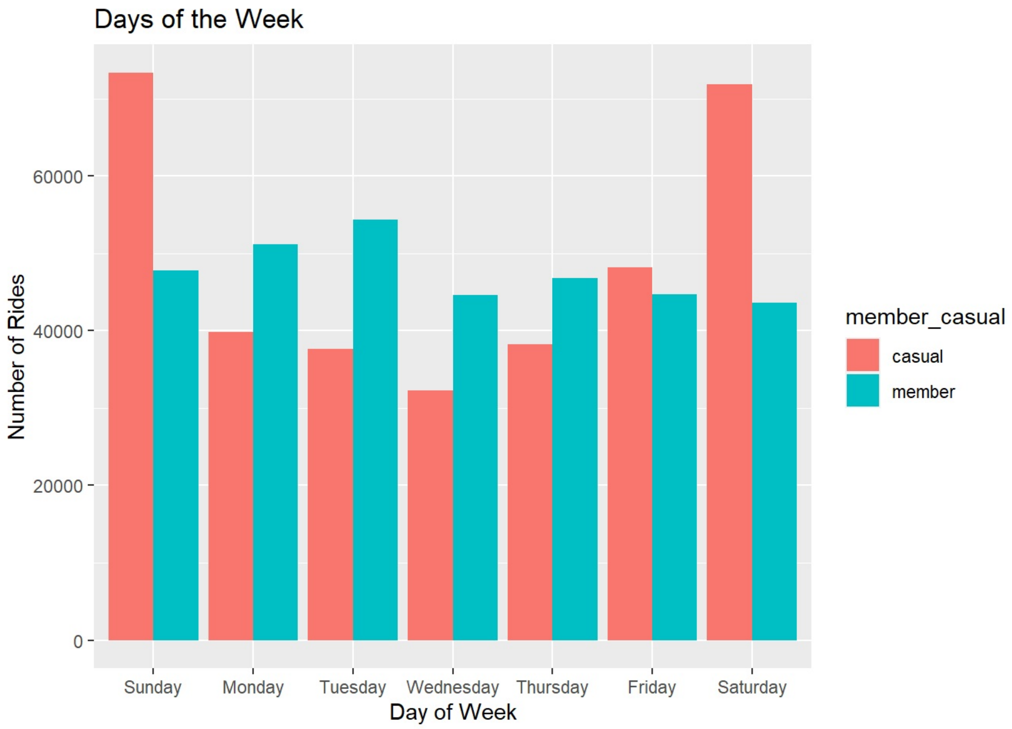
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

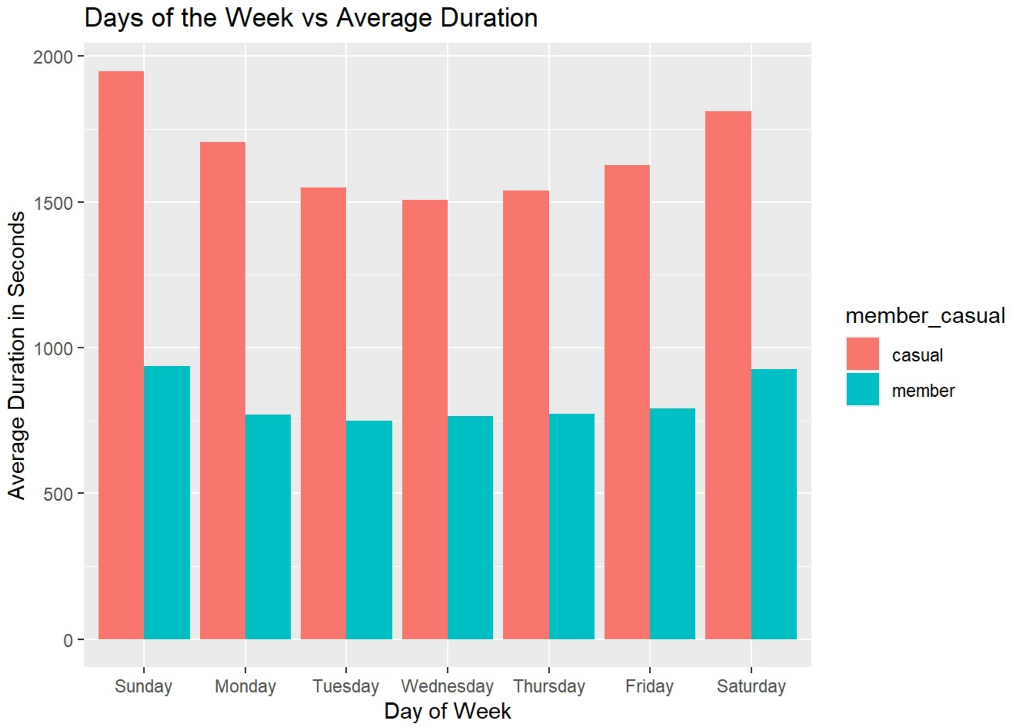
Plot the number of rides by user type during the week.

```
Aug21 %>%
  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.

```
Aug21 %>%
  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(Aug21$day_of_week,Aug21$member_casual))
```

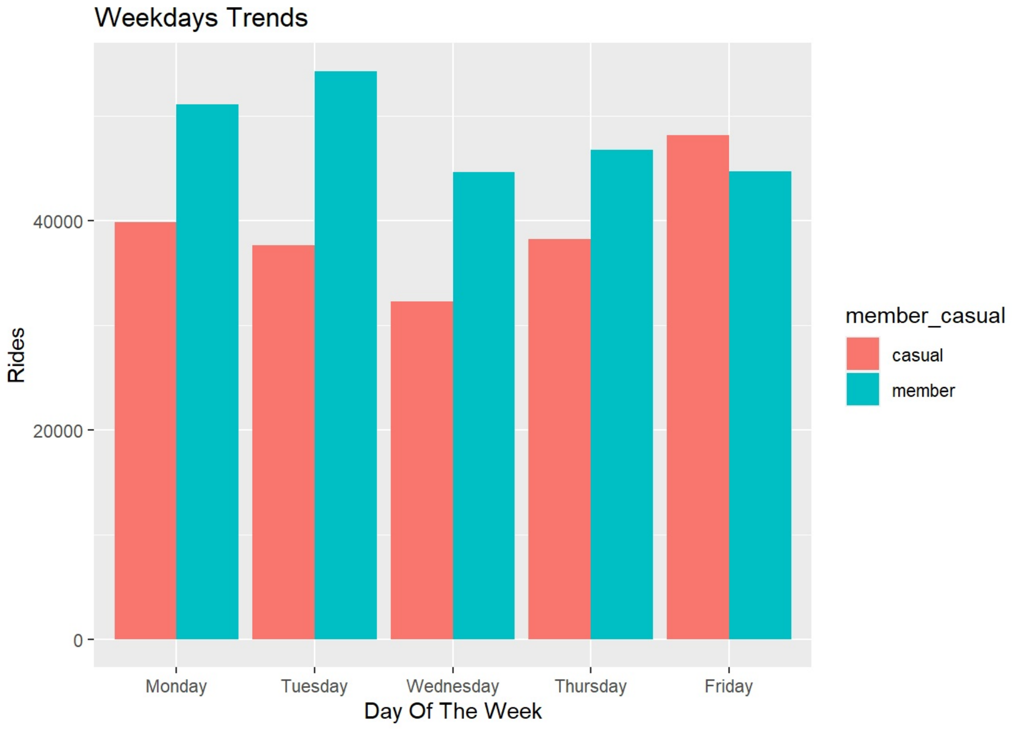
Rename columns

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

```
##   day_of_week member_casual  Freq
## 1      Sunday        casual 73381
## 2      Monday        casual 39852
## 3     Tuesday        casual 37633
## 4   Wednesday        casual 32306
## 5    Thursday        casual 38219
## 6      Friday        casual 48164
```

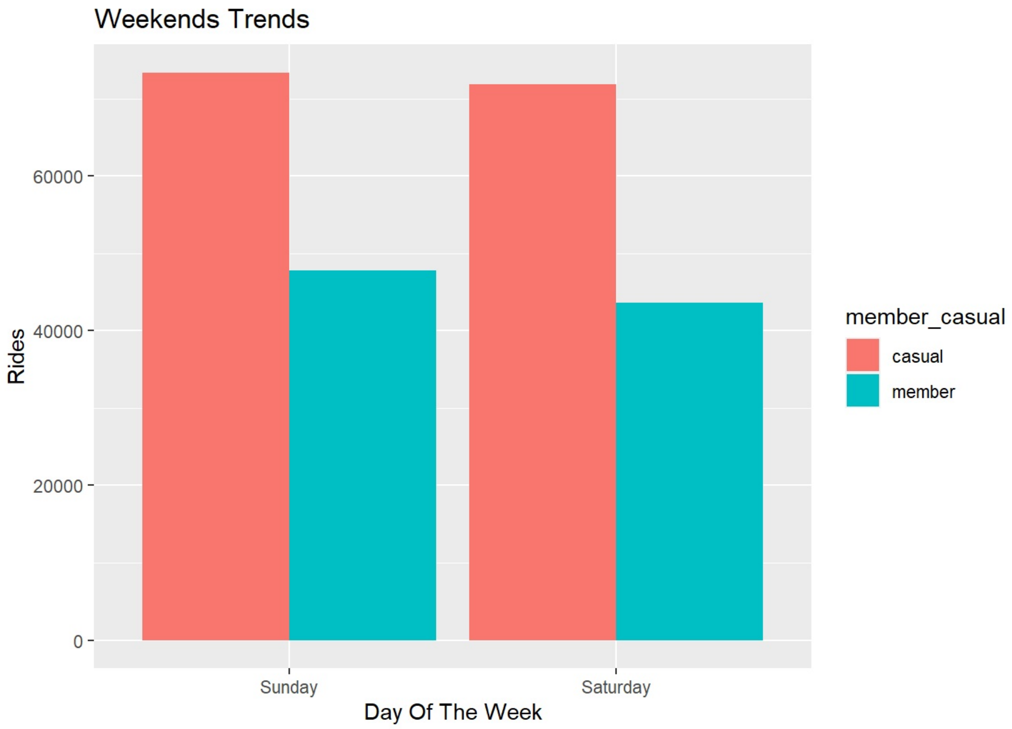
Weekday trends (Monday through Friday).

```
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(Aug21$rideable_type,Aug21$member_casual))
```

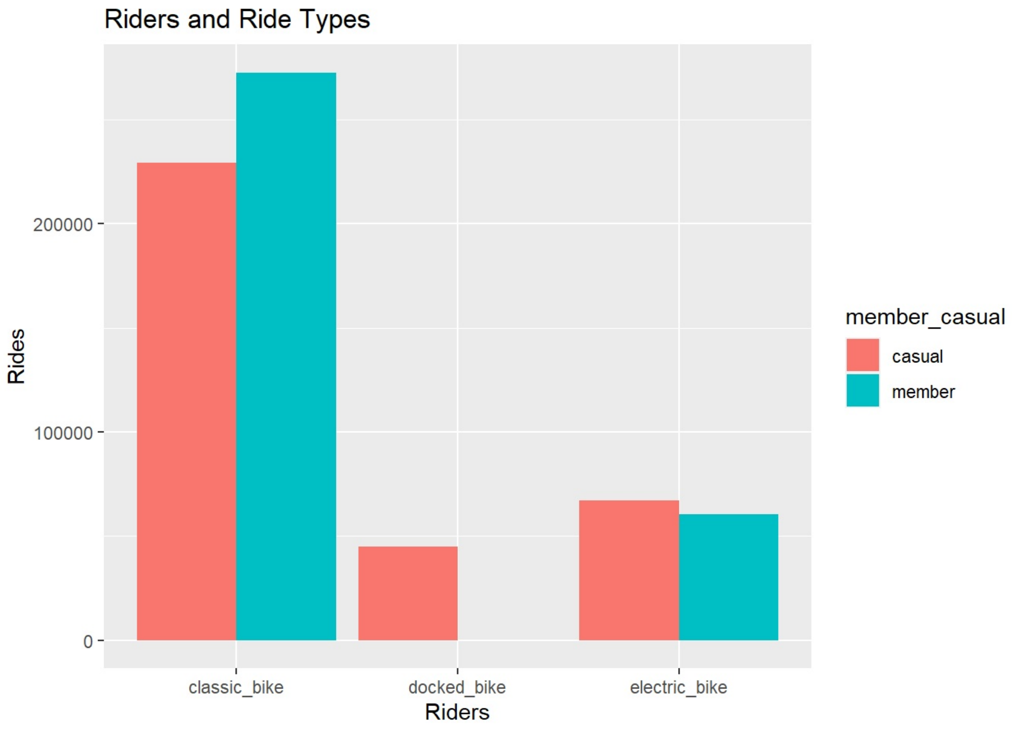
Rename columns.

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

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual 229128
## 2   docked_bike        casual  45064
## 3 electric_bike        casual  67245
## 4  classic_bike        member 272609
## 5   docked_bike        member      0
## 6 electric_bike        member  60255
```

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(Aug21, “Aug21.csv”)
```