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)
```

```
## ── 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.

```
Jan21 <- read_csv("C:/Users/theby/Documents/202101-divvy-tripdata.csv")
```

```
## Rows: 96834 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
## 
## 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(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 start…² ended…³ start…⁴ start…⁵ end_s…⁶ end_s…⁷ start…⁸
##   <chr>          <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>     <dbl>
## 1 E19E6F1B8D4C4… electr… 1/23/2… 1/23/2… Califo… 17660   <NA>    <NA>       41.9
## 2 DC88F20C2C55F… electr… 1/27/2… 1/27/2… Califo… 17660   <NA>    <NA>       41.9
## 3 EC45C94683FE3… electr… 1/21/2… 1/21/2… Califo… 17660   <NA>    <NA>       41.9
## 4 4FA453A75AE37… electr… 1/7/20… 1/7/20… Califo… 17660   <NA>    <NA>       41.9
## 5 BE5E8EB4E7263… electr… 1/23/2… 1/23/2… Califo… 17660   <NA>    <NA>       41.9
## 6 5D8969F88C773… electr… 1/9/20… 1/9/20… Califo… 17660   <NA>    <NA>       41.9
## # … with 4 more variables: start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names ¹rideable_type,
## #   ²started_at, ³ended_at, ⁴start_station_name, ⁵start_station_id,
## #   ⁶end_station_name, ⁷end_station_id, ⁸start_lat
```

```
tail(Jan21)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ start…² ended…³ start…⁴ start…⁵ end_s…⁶ end_s…⁷ start…⁸
##   <chr>          <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>     <dbl>
## 1 44DE07FCDD3AD… docked… 1/17/2… 1/17/2… Lake S… 13300   Lake S… 13300      41.9
## 2 B1A5336E1412D… classi… 1/19/2… 1/19/2… Lake S… 13300   Lakefr… KA1504…    41.9
## 3 57EA5CB7DCD75… classi… 1/5/20… 1/5/20… Lake S… 13300   Lakefr… KA1504…    41.9
## 4 815B319A078CC… classi… 1/7/20… 1/7/20… Lakefr… KA1504… Lakefr… KA1504…    42.0
## 5 6DB04151565CE… classi… 1/6/20… 1/6/20… Lakefr… KA1504… Lakefr… KA1504…    42.0
## 6 8008C9C998083… docked… 1/17/2… 1/17/2… Lake S… 13300   Lake S… 13300      41.9
## # … with 4 more variables: start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names ¹rideable_type,
## #   ²started_at, ³ended_at, ⁴start_station_name, ⁵start_station_id,
## #   ⁶end_station_name, ⁷end_station_id, ⁸start_lat
```

```
summary(Jan21)
```

```
##    ride_id          rideable_type       started_at          ended_at        
##  Length:96834       Length:96834       Length:96834       Length:96834      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##  start_station_name start_station_id   end_station_name   end_station_id    
##  Length:96834       Length:96834       Length:96834       Length:96834      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    start_lat       start_lng         end_lat         end_lng      
##  Min.   :41.64   Min.   :-87.78   Min.   :41.64   Min.   :-87.81  
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66  
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64  
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65  
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63  
##  Max.   :42.06   Max.   :-87.53   Max.   :42.07   Max.   :-87.51  
##                                   NA's   :103     NA's   :103     
##  member_casual     
##  Length:96834      
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
##                    
```

```
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        : chr [1:96834] "1/23/2021 16:14" "1/27/2021 18:43" "1/21/2021 22:35" "1/7/2021 13:31" ..
.
##  $ ended_at          : chr [1:96834] "1/23/2021 16:24" "1/27/2021 18:47" "1/21/2021 22:37" "1/7/2021 13:42" ..
.
##  $ 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_character(),
##   ..   ended_at = col_character(),
##   ..   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>
```

Columns *started_at* and *ended_at* need to be convert from character data type to date data type. **Str()** syntax confirms changes.

```
Jan21$started_at <- mdy_hm(Jan21$started_at)
Jan21$ended_at <- mdy_hm(Jan21$ended_at)
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 16:14:00" "2021-01-27 18:43:00" ...
##  $ ended_at          : POSIXct[1:96834], format: "2021-01-23 16:24:00" "2021-01-27 18:47:00" ...
##  $ 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_character(),
##   ..   ended_at = col_character(),
##   ..   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$day <- format(as.Date(Jan21$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 1 minute.

```
Jan21 <- subset (Jan21, ride_length > "1")
```

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] 878.8599
```

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

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

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

```
## [1] 1189560
```

```
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     360.0     540.0     878.9     960.0 1189560.0
```

Compare the members and casual users

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

```
##   Jan21$member_casual Jan21$ride_length
## 1              casual         1588.8783
## 2              member          726.8542
```

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

```
##   Jan21$member_casual Jan21$ride_length
## 1              casual               780
## 2              member               540
```

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

```
##   Jan21$member_casual Jan21$ride_length
## 1              casual           1189560
## 2              member             73560
```

```
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         1418.8809
## 2               member            Friday          708.9743
## 3               casual            Monday         1198.0084
## 4               member            Monday          687.4889
## 5               casual          Saturday         1998.5973
## 6               member          Saturday          791.2074
## 7               casual            Sunday         1857.6845
## 8               member            Sunday          783.9969
## 9               casual          Thursday         1229.8414
## 10              member          Thursday          695.9691
## 11              casual           Tuesday         1394.9593
## 12              member           Tuesday          698.6534
## 13              casual         Wednesday         1575.2793
## 14              member         Wednesday          731.4766
```

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         1857.6845
## 2              member            Sunday          783.9969
## 3              casual            Monday         1198.0084
## 4              member            Monday          687.4889
## 5              casual           Tuesday         1394.9593
## 6              member           Tuesday          698.6534
```

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            2358            1858.
## 2 casual              2            1657            1198.
## 3 casual              3            1476            1395.
## 4 casual              4            1665            1575.
## 5 casual              5            1892            1230.
## 6 casual              6            2216            1419.
```

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
##  14629  68332
```

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

```
##
##  classic_bike   docked_bike electric_bike
##         61013          2091         19857
```

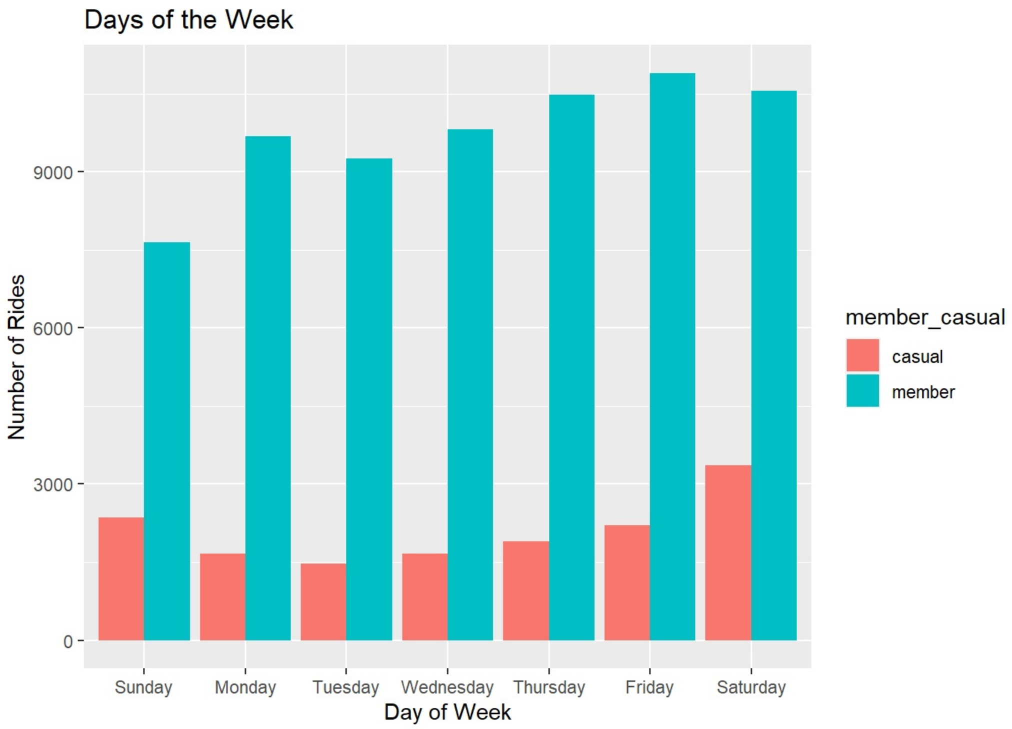
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

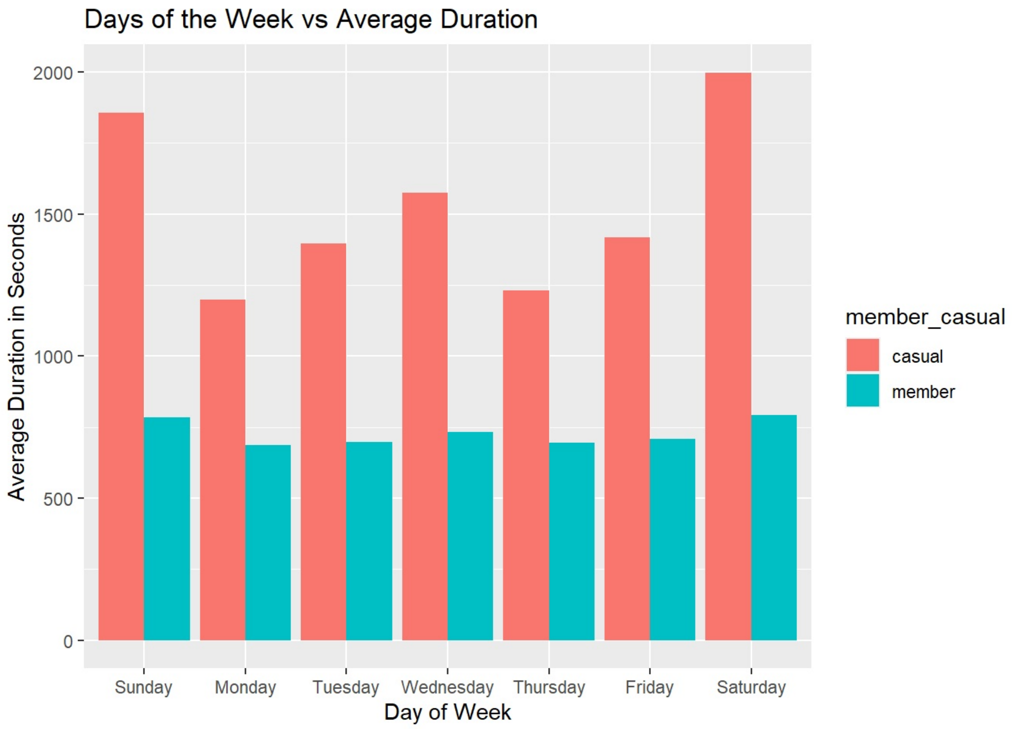
Plot the number of rides by user type during the week.

```
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 = "Day 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 = "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(Jan21$day_of_week,Jan21$member_casual))
```

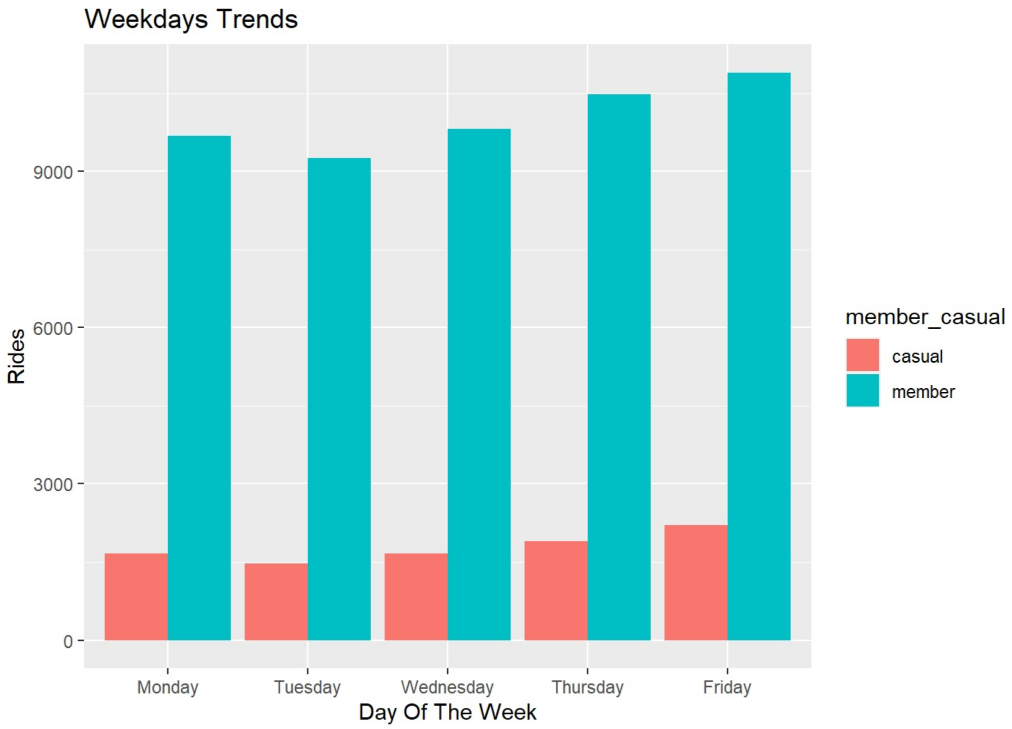
Rename columns

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

```
##   day_of_week member_casual Freq
## 1      Sunday        casual 2358
## 2      Monday        casual 1657
## 3     Tuesday        casual 1476
## 4   Wednesday        casual 1665
## 5    Thursday        casual 1892
## 6      Friday        casual 2216
```

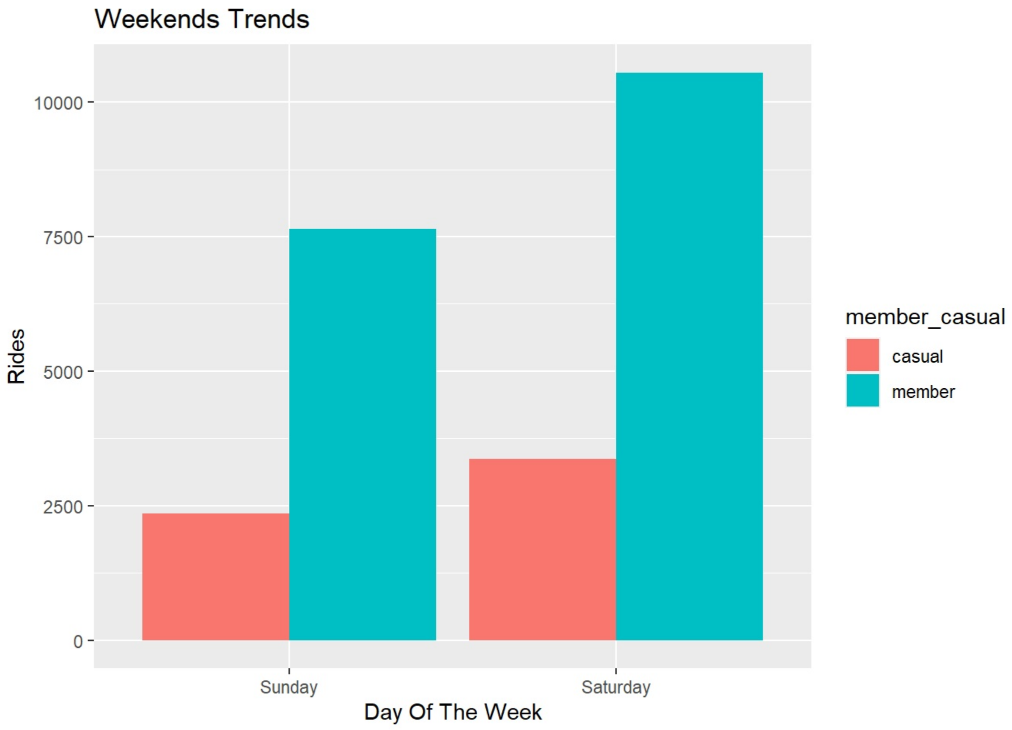
Weekday trends (Monday through Friday).

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

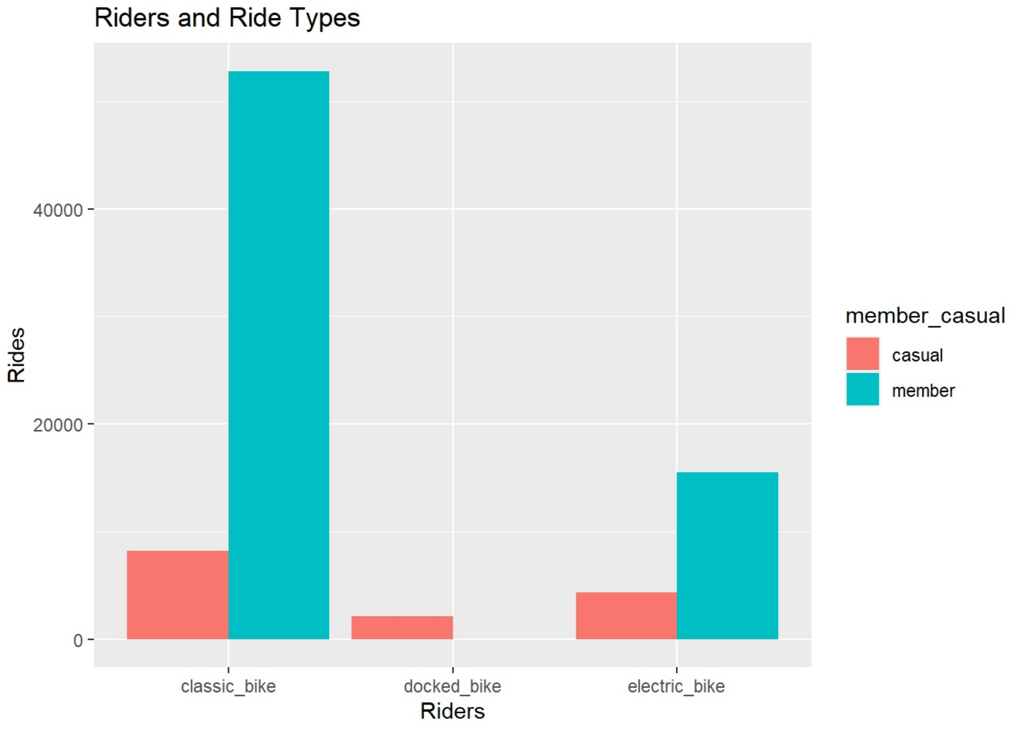
Rename columns.

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

```
##   rideable_type member_casual  Freq
## 1  classic_bike        casual  8191
## 2   docked_bike        casual  2090
## 3 electric_bike        casual  4348
## 4  classic_bike        member 52822
## 5   docked_bike        member     1
## 6 electric_bike        member 15509
```

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(Jan21, “Jan21.csv”)