Cyclistic Case Study Q1_2021

Hezar K

2022-11-29

This is an analysis for Cyclistic Case Study for Google Data Analytics Course. This is an analysis for 2021's first quarter.

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("202101-divvy-tripdata.csv")
```

```
## Rows: 96834 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.
```

```
Feb21 <- read_csv("202102-divvy-tripdata.csv")
```

```
## Rows: 49622 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.
```

```
Mar21 <- read_csv("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()**. Then, we need to combine all data one dataframe. Then we examine dataframes to find dimensions, **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()**.

```
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"
```

```
colnames(Feb21)
```

```
##  [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"
```

```
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"
```

Since all column names are the same. We can combine the data for each month into quarters.

```
q1_2021 <- bind_rows(Jan21, Feb21, Mar21)
```

View(q1_2021)

```
nrow(q1_2021)
```

```
## [1] 374952
```

```
dim(q1_2021)
```

```
## [1] 374952     13
```

```
head(q1_2021)
```

```
## # 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(q1_2021)
```

```
## # 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(q1_2021)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:374952      Length:374952      Min.   :2021-01-01 00:02:05.0  
##  Class :character   Class :character   1st Qu.:2021-01-29 19:27:50.0  
##  Mode  :character   Mode  :character   Median :2021-03-08 16:10:06.5  
##                                        Mean   :2021-02-26 11:12:08.9  
##                                        3rd Qu.:2021-03-21 13:36:25.0  
##                                        Max.   :2021-03-31 23:59:08.0  
##                                                                       
##     ended_at                      start_station_name start_station_id  
##  Min.   :2021-01-01 00:08:39.00   Length:374952      Length:374952     
##  1st Qu.:2021-01-29 19:40:06.50   Class :character   Class :character  
##  Median :2021-03-08 16:32:02.50   Mode  :character   Mode  :character  
##  Mean   :2021-02-26 11:33:15.50                                        
##  3rd Qu.:2021-03-21 14:06:50.25                                        
##  Max.   :2021-04-06 11:00:11.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:374952      Length:374952      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.   :42.07   Max.   :-87.53  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   :41.54   Min.   :-88.07   Length:374952     
##  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.   :-87.51                     
##  NA's   :484     NA's   :484
```

```
str(q1_2021)
```

```
## spc_tbl_ [374,952 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:374952] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA453A75AE377
DB" ...
##  $ rideable_type     : chr [1:374952] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:374952], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" ...
##  $ ended_at          : POSIXct[1:374952], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" ...
##  $ start_station_name: chr [1:374952] "California Ave & Cortez St" "California Ave & Cortez St" "California Av
e & Cortez St" "California Ave & Cortez St" ...
##  $ start_station_id  : chr [1:374952] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:374952] NA NA NA NA ...
##  $ end_station_id    : chr [1:374952] NA NA NA NA ...
##  $ start_lat         : num [1:374952] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:374952] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:374952] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:374952] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:374952] "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.

```
q1_2021$date <- as.Date(q1_2021$started_at)
q1_2021$month <- format(as.Date(q1_2021$date), "%m")
q1_2021$day <- format(as.Date(q1_2021$date), "%d")
q1_2021$year <- format(as.Date(q1_2021$date), "%Y")
q1_2021$day_of_week <- format(as.Date(q1_2021$date), "%A")
q1_2021$ride_length <- difftime(q1_2021$ended_at,q1_2021$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(q1_2021$ride_length)
```

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

Recheck *ride_length* data type.

```
q1_2021$ride_length <- as.numeric(as.character(q1_2021$ride_length))
is.numeric(q1_2021$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
q1_2021 <- na.omit(q1_2021)
```

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.

```
q1_2021 <- subset(q1_2021, nchar(as.character(ride_id)) == 16)
```

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

```
q1_2021 <- subset (q1_2021, ride_length > "1")
```

STEP FOUR: ANALYZE DATA

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

```
mean(q1_2021$ride_length)
```

```
## [1] 1231.463
```

```
median(q1_2021$ride_length)
```

```
## [1] 681
```

```
max(q1_2021$ride_length)
```

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

```
min(q1_2021$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(q1_2021$ride_length)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
##       2.0     389.2     681.0    1231.5    1250.0 1900899.0
```

Compare the members and casual users

```
aggregate(q1_2021$ride_length ~ q1_2021$member_casual, FUN = mean)
```

```
##   q1_2021$member_casual q1_2021$ride_length
## 1                casual           2246.2747
## 2                member            800.9471
```

```
aggregate(q1_2021$ride_length ~ q1_2021$member_casual, FUN = median)
```

```
##   q1_2021$member_casual q1_2021$ride_length
## 1                casual                1073
## 2                member                 578
```

```
aggregate(q1_2021$ride_length ~ q1_2021$member_casual, FUN = max)
```

```
##   q1_2021$member_casual q1_2021$ride_length
## 1                casual             1900899
## 2                member               88461
```

```
aggregate(q1_2021$ride_length ~ q1_2021$member_casual, FUN = min)
```

```
##   q1_2021$member_casual q1_2021$ride_length
## 1                casual                   2
## 2                member                   2
```

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

```
aggregate(q1_2021$ride_length ~ q1_2021$member_casual + q1_2021$day_of_week, FUN = mean)
```

```
##    q1_2021$member_casual q1_2021$day_of_week q1_2021$ride_length
## 1                 casual              Friday           1929.1317
## 2                 member              Friday            745.5758
## 3                 casual              Monday           2486.8692
## 4                 member              Monday            791.3900
## 5                 casual            Saturday           2594.2569
## 6                 member            Saturday            896.5394
## 7                 casual              Sunday           2370.6814
## 8                 member              Sunday            910.0566
## 9                 casual            Thursday           1606.4854
## 10                member            Thursday            719.0917
## 11                casual             Tuesday           2130.6938
## 12                member             Tuesday            783.8548
## 13                casual           Wednesday           1718.6392
## 14                member           Wednesday            763.0284
```

Sort the days of the week in order.

```
q1_2021$day_of_week <- ordered(q1_2021$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursda
y", "Friday", "Saturday"))
```

Assign the aggregate the average ride length by each day of the week for members and users to x.

```
x <- aggregate(q1_2021$ride_length ~ q1_2021$member_casual + q1_2021$day_of_week, FUN = mean)

head(x)
```

```
##   q1_2021$member_casual q1_2021$day_of_week q1_2021$ride_length
## 1                casual              Sunday           2370.6814
## 2                member              Sunday            910.0566
## 3                casual              Monday           2486.8692
## 4                member              Monday            791.3900
## 5                casual             Tuesday           2130.6938
## 6                member             Tuesday            783.8548
```

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

```
y <- q1_2021 %>%
  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           19507            2371.
## 2 casual              2           12887            2487.
## 3 casual              3           11615            2131.
## 4 casual              4           10280            1719.
## 5 casual              5            7547            1606.
## 6 casual              6           10289            1929.
```

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

```
table(q1_2021$member_casual)
```

```
##
## casual member
##  98939 233219
```

```
table(q1_2021$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##        248183         19034         64941
```

```
table(q1_2021$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##     47247     46732     47091     46423     37738     42944     63983
```

```
table(q1_2021$month)
```

```
##
##     01     02     03
##  83498  42986 205674
```

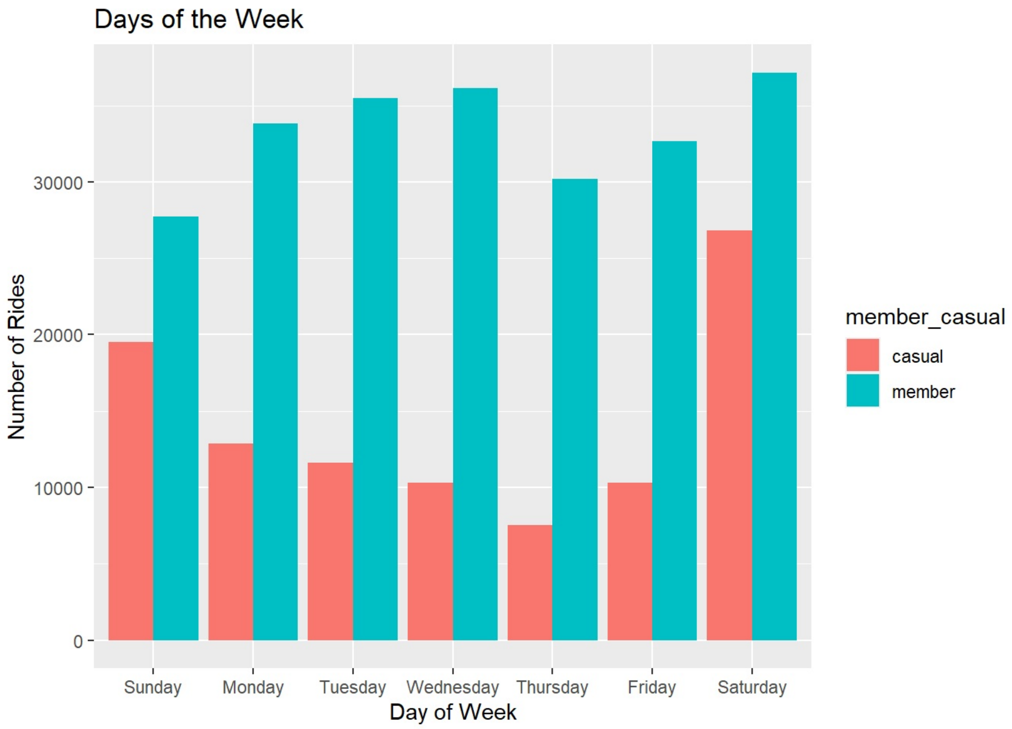
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

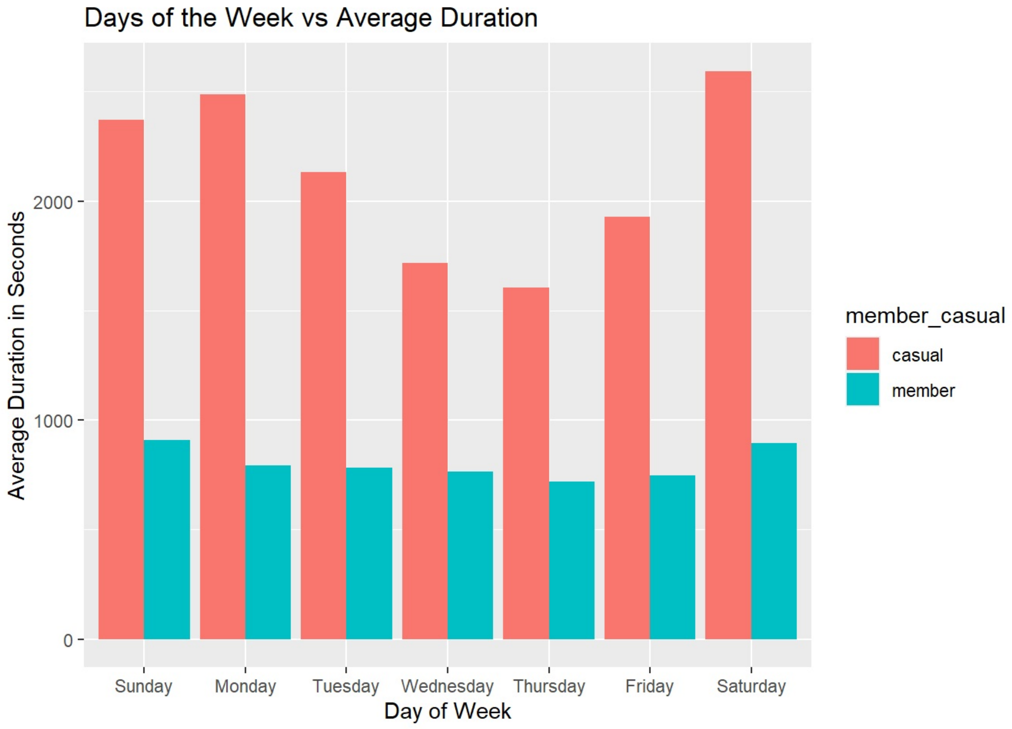
Plot the number of rides by user type during the week.

```
q1_2021 %>%
  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.

```
q1_2021 %>%
  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(q1_2021$day_of_week,q1_2021$member_casual))
```

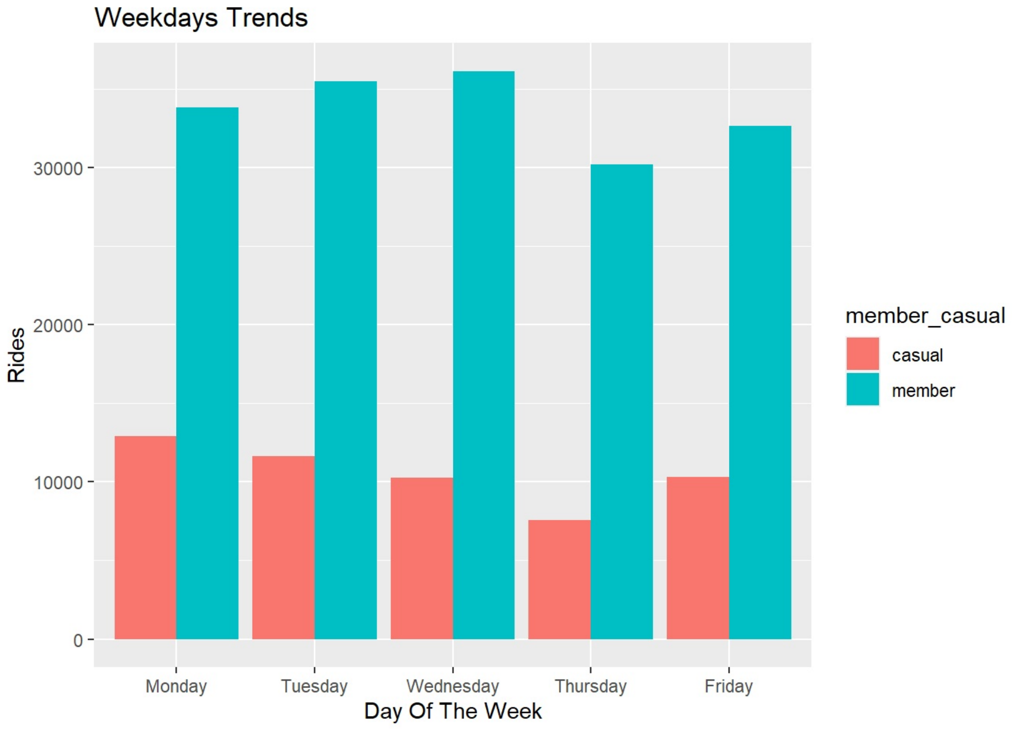
Rename columns

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

```
##   day_of_week member_casual  Freq
## 1      Sunday        casual 19507
## 2      Monday        casual 12887
## 3     Tuesday        casual 11615
## 4   Wednesday        casual 10280
## 5    Thursday        casual  7547
## 6      Friday        casual 10289
```

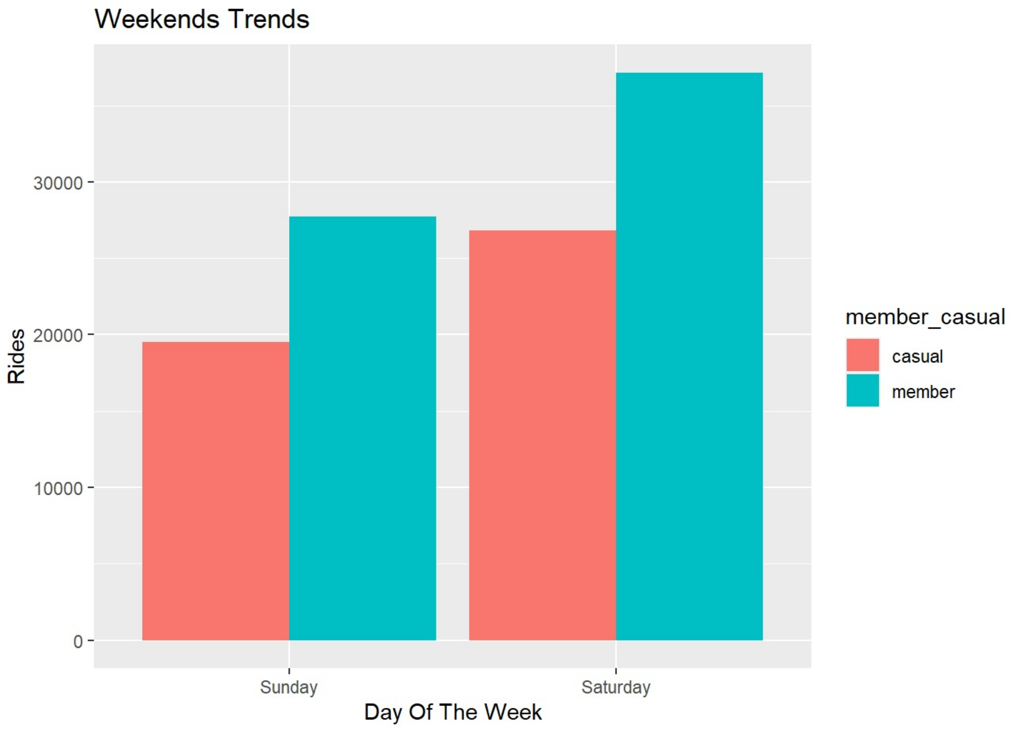
Weekday trends (Monday through Friday).

```
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(q1_2021$rideable_type,q1_2021$member_casual))
```

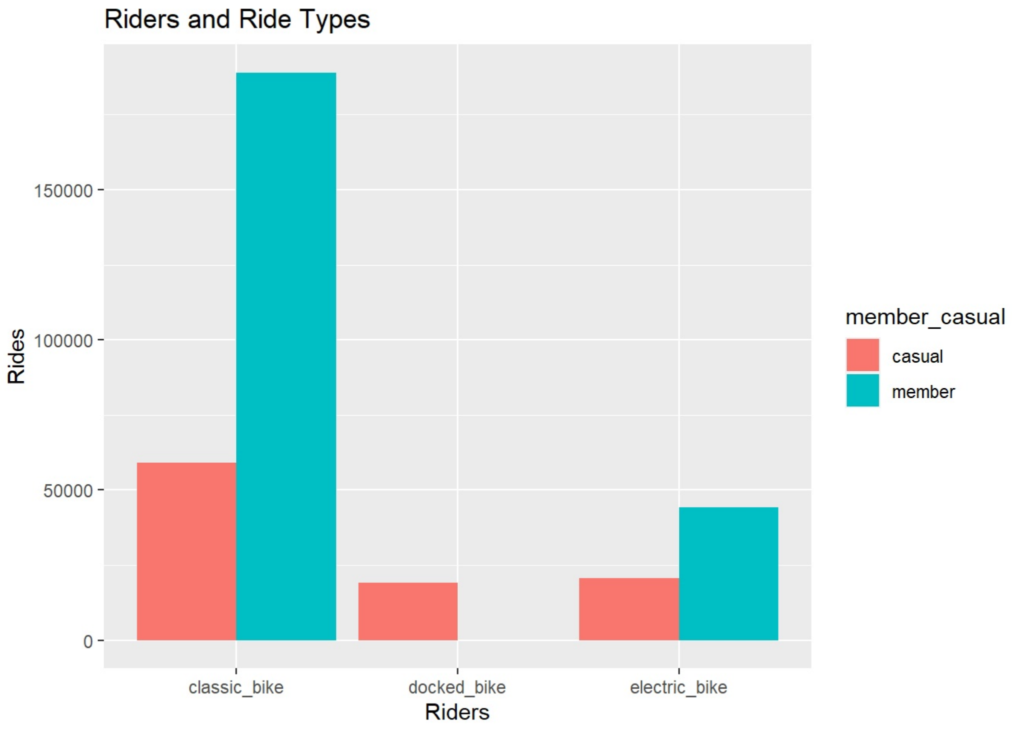
Rename columns.

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

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual  59171
## 2   docked_bike        casual  19033
## 3 electric_bike        casual  20735
## 4  classic_bike        member 189012
## 5   docked_bike        member      1
## 6 electric_bike        member  44206
```

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(q1_2021, “q1_2021.csv”)