Cyclistic Case Study Q3_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 third 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.

```
Jul21 <- read_csv("202107-divvy-tripdata.csv")
```

```
## Rows: 822410 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.
```

```
Aug21 <- read_csv("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.
```

```
Sep21 <- read_csv("202109-divvy-tripdata.csv")
```

```
## Rows: 756147 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(Jul21)
```

```
##  [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(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"
```

```
colnames(Sep21)
```

```
##  [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.

```
q3_2021 <- bind_rows(Jul21, Aug21, Sep21)
```

```
View(q3_2021)
```

```
nrow(q3_2021)
```

```
## [1] 2382909
```

```
dim(q3_2021)
```

```
## [1] 2382909      13
```

```
head(q3_2021)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 0A1B623926EF4… docked… 2021-07-02 14:44:36 2021-07-02 15:19:58 Michig… 13001
## 2 B2D5583A5A5E7… classi… 2021-07-07 16:57:42 2021-07-07 17:16:09 Califo… 17660
## 3 6F264597DDBF4… classi… 2021-07-25 11:30:55 2021-07-25 11:48:45 Wabash… SL-012
## 4 379B58EAB20E8… classi… 2021-07-08 22:08:30 2021-07-08 22:23:32 Califo… 17660
## 5 6615C1E4EB08E… electr… 2021-07-28 16:08:06 2021-07-28 16:27:09 Califo… 17660
## 6 62DC2B32872F9… electr… 2021-07-29 17:09:08 2021-07-29 17:15:00 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(q3_2021)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 0A6AA3B1A1EC5… classi… 2021-09-14 23:00:37 2021-09-14 23:10:55 Ellis … KA1503…
## 2 FA66BCAB0D73D… classi… 2021-09-22 15:46:57 2021-09-22 16:01:15 Ellis … 584
## 3 1D44DEFB5D36C… classi… 2021-09-25 16:25:23 2021-09-25 16:40:29 Ellis … KA1503…
## 4 6A346EA57FC23… classi… 2021-09-25 16:26:05 2021-09-25 16:40:30 Ellis … KA1503…
## 5 49360AFD77110… classi… 2021-09-15 17:57:48 2021-09-15 18:24:06 Ellis … KA1503…
## 6 343190A2DC023… electr… 2021-09-11 18:01:06 2021-09-11 18:08:26 Wells … TA1306…
## # … 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(q3_2021)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:2382909     Length:2382909     Min.   :2021-07-01 00:00:22.00  
##  Class :character   Class :character   1st Qu.:2021-07-24 05:40:49.00  
##  Mode  :character   Mode  :character   Median :2021-08-15 00:35:16.00  
##                                        Mean   :2021-08-15 12:51:39.25  
##                                        3rd Qu.:2021-09-06 19:30:07.00  
##                                        Max.   :2021-09-30 23:59:48.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2021-07-01 00:04:51.00   Length:2382909     Length:2382909    
##  1st Qu.:2021-07-24 06:21:28.00   Class :character   Class :character  
##  Median :2021-08-15 00:58:46.00   Mode  :character   Mode  :character  
##  Mean   :2021-08-15 13:13:49.28                                        
##  3rd Qu.:2021-09-06 19:55:00.00                                        
##  Max.   :2021-10-01 22:55:35.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:2382909     Length:2382909     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.   :41.57   Min.   :-87.87   Length:2382909    
##  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.   :42.17   Max.   :-87.49                     
##  NA's   :2032    NA's   :2032
```

```
str(q3_2021)
```

```
## spc_tbl_ [2,382,909 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:2382909] "0A1B623926EF4E16" "B2D5583A5A5E76EE" "6F264597DDBF427A" "379B58EAB20E8
AA5" ...
##  $ rideable_type     : chr [1:2382909] "docked_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:2382909], format: "2021-07-02 14:44:36" "2021-07-07 16:57:42" ...
##  $ ended_at          : POSIXct[1:2382909], format: "2021-07-02 15:19:58" "2021-07-07 17:16:09" ...
##  $ start_station_name: chr [1:2382909] "Michigan Ave & Washington St" "California Ave & Cortez St" "Wabash Ave
& 16th St" "California Ave & Cortez St" ...
##  $ start_station_id  : chr [1:2382909] "13001" "17660" "SL-012" "17660" ...
##  $ end_station_name  : chr [1:2382909] "Halsted St & North Branch St" "Wood St & Hubbard St" "Rush St & Hubbar
d St" "Carpenter St & Huron St" ...
##  $ end_station_id    : chr [1:2382909] "KA1504000117" "13432" "KA1503000044" "13196" ...
##  $ start_lat         : num [1:2382909] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:2382909] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num [1:2382909] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:2382909] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr [1:2382909] "casual" "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.

```
q3_2021$date <- as.Date(q3_2021$started_at)
q3_2021$month <- format(as.Date(q3_2021$date), "%m")
q3_2021$day <- format(as.Date(q3_2021$date), "%d")
q3_2021$year <- format(as.Date(q3_2021$date), "%Y")
q3_2021$day_of_week <- format(as.Date(q3_2021$date), "%A")
q3_2021$ride_length <- difftime(q3_2021$ended_at,q3_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(q3_2021$ride_length)
```

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

Recheck *ride_length* data type.

```
q3_2021$ride_length <- as.numeric(as.character(q3_2021$ride_length))
is.numeric(q3_2021$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
q3_2021 <- na.omit(q3_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.

```
q3_2021 <- subset(q3_2021, nchar(as.character(ride_id)) == 16)
```

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

```
q3_2021 <- subset (q3_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(q3_2021$ride_length)
```

```
## [1] 1314.768
```

```
median(q3_2021$ride_length)
```

```
## [1] 769
```

```
max(q3_2021$ride_length)
```

```
## [1] 2946429
```

```
min(q3_2021$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(q3_2021$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       2     442     769    1315    1371 2946429
```

Compare the members and casual users

```
aggregate(q3_2021$ride_length ~ q3_2021$member_casual, FUN = mean)
```

```
##   q3_2021$member_casual q3_2021$ride_length
## 1                casual           1809.9536
## 2                member            809.5788
```

```
aggregate(q3_2021$ride_length ~ q3_2021$member_casual, FUN = median)
```

```
##   q3_2021$member_casual q3_2021$ride_length
## 1                casual                 988
## 2                member                 604
```

```
aggregate(q3_2021$ride_length ~ q3_2021$member_casual, FUN = max)
```

```
##   q3_2021$member_casual q3_2021$ride_length
## 1                casual             2946429
## 2                member               89183
```

```
aggregate(q3_2021$ride_length ~ q3_2021$member_casual, FUN = min)
```

```
##   q3_2021$member_casual q3_2021$ride_length
## 1                casual                   2
## 2                member                   2
```

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

```
aggregate(q3_2021$ride_length ~ q3_2021$member_casual + q3_2021$day_of_week, FUN = mean)
```

```
##    q3_2021$member_casual q3_2021$day_of_week q3_2021$ride_length
## 1                 casual              Friday           1719.3444
## 2                 member              Friday            792.9569
## 3                 casual              Monday           1919.8219
## 4                 member              Monday            788.4771
## 5                 casual            Saturday           1942.8085
## 6                 member            Saturday            913.5273
## 7                 casual              Sunday           2057.9685
## 8                 member              Sunday            932.2590
## 9                 casual            Thursday           1625.1168
## 10                member            Thursday            764.8478
## 11                casual             Tuesday           1547.7248
## 12                member             Tuesday            745.8214
## 13                casual           Wednesday           1550.9503
## 14                member           Wednesday            761.1493
```

Sort the days of the week in order.

```
q3_2021$day_of_week <- ordered(q3_2021$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(q3_2021$ride_length ~ q3_2021$member_casual + q3_2021$day_of_week, FUN = mean)

head(x)
```

```
##   q3_2021$member_casual q3_2021$day_of_week q3_2021$ride_length
## 1                casual              Sunday           2057.9685
## 2                member              Sunday            932.2590
## 3                casual              Monday           1919.8219
## 4                member              Monday            788.4771
## 5                casual             Tuesday           1547.7248
## 6                member             Tuesday            745.8214
```

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

```
y <- q3_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          190367            2058.
## 2 casual              2          114976            1920.
## 3 casual              3           98677            1548.
## 4 casual              4          105457            1551.
## 5 casual              5          124370            1625.
## 6 casual              6          146797            1719.
```

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

```
table(q3_2021$member_casual)
```

```
## 
##  casual  member 
## 1003691  983815
```

```
table(q3_2021$rideable_type)
```

```
## 
##  classic_bike   docked_bike electric_bike 
##       1468149        138096        381261
```

```
table(q3_2021$day_of_week)
```

```
## 
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday 
##    310210    246884    241353    255427    286224    289218    358190
```

```
table(q3_2021$month)
```

```
## 
##     07     08     09 
## 692193 674301 621012
```

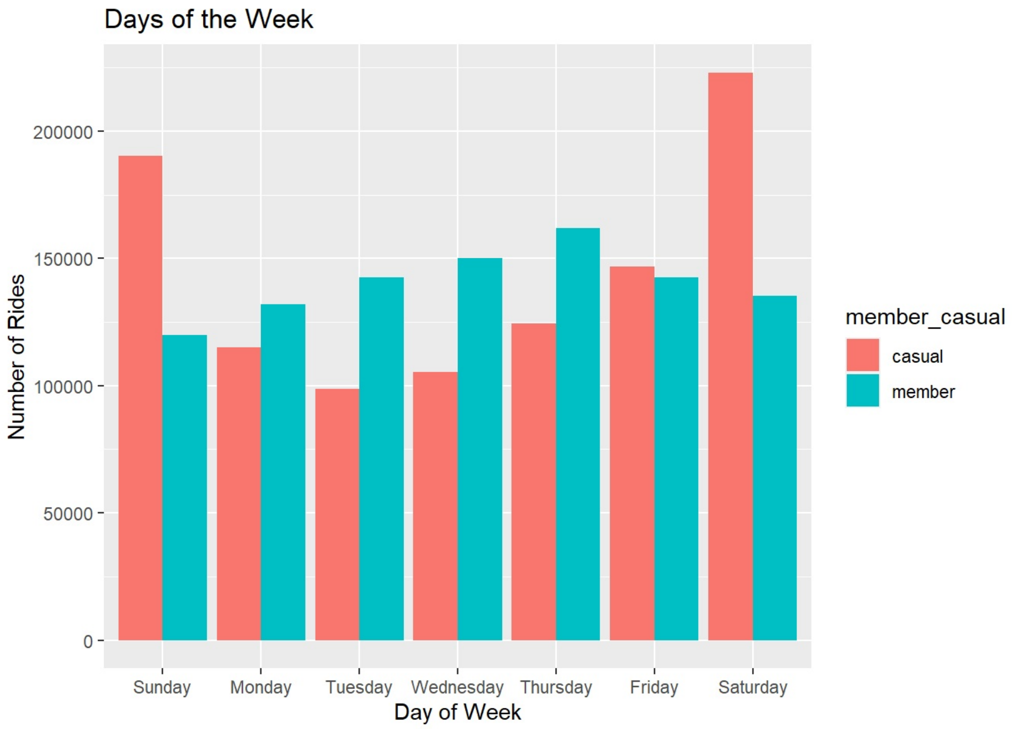
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

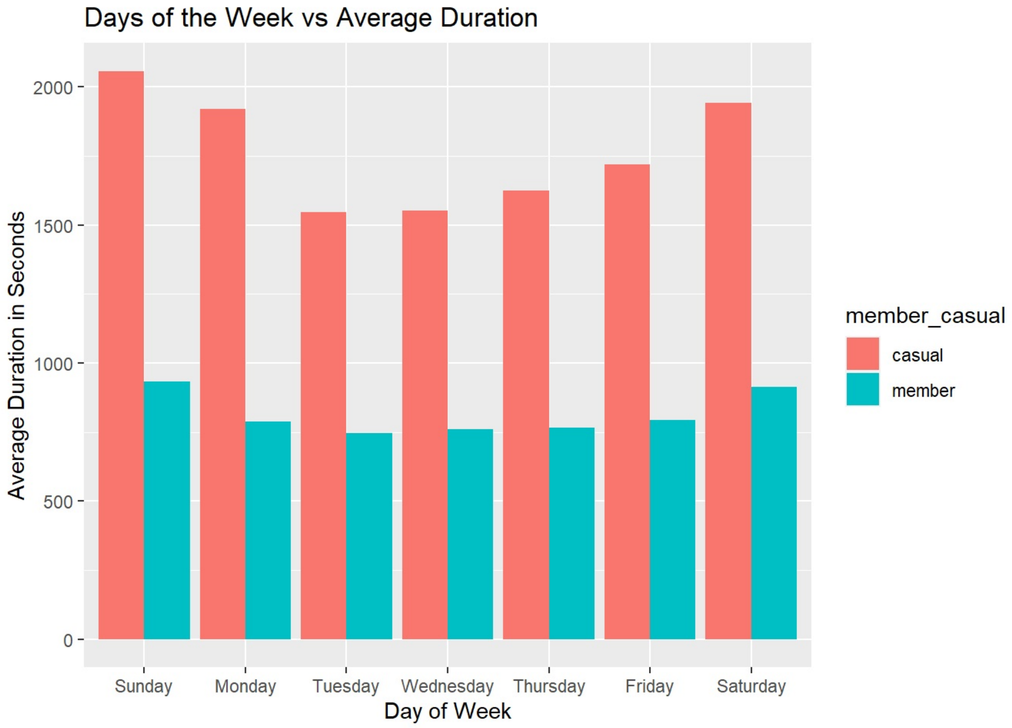
Plot the number of rides by user type during the week.

```
q3_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.

```
q3_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(q3_2021$day_of_week,q3_2021$member_casual))
```

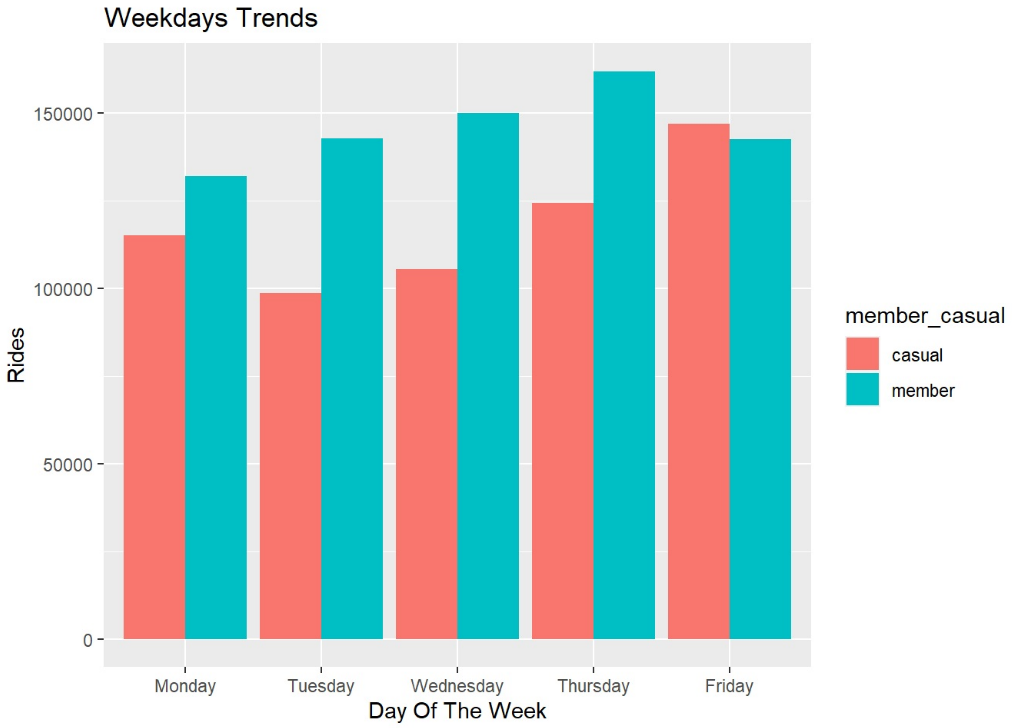
Rename columns

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

```
##   day_of_week member_casual   Freq
## 1      Sunday        casual 190367
## 2      Monday        casual 114976
## 3     Tuesday        casual  98677
## 4   Wednesday        casual 105457
## 5    Thursday        casual 124370
## 6      Friday        casual 146797
```

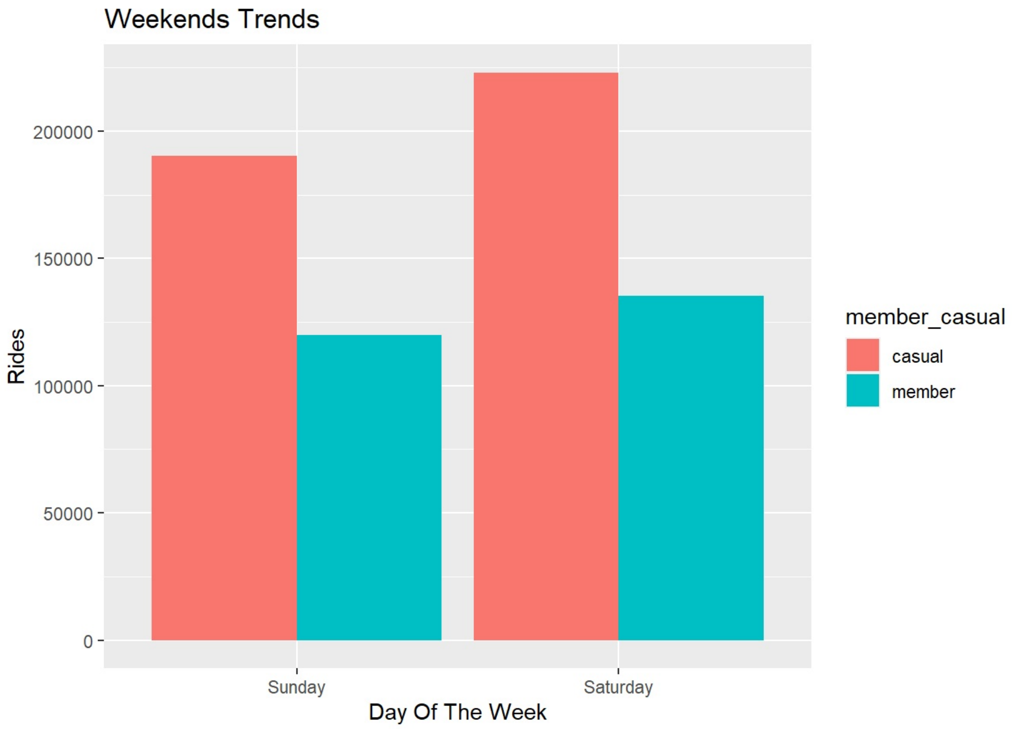
Weekday trends (Monday through Friday).

```
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(q3_2021$rideable_type,q3_2021$member_casual))
```

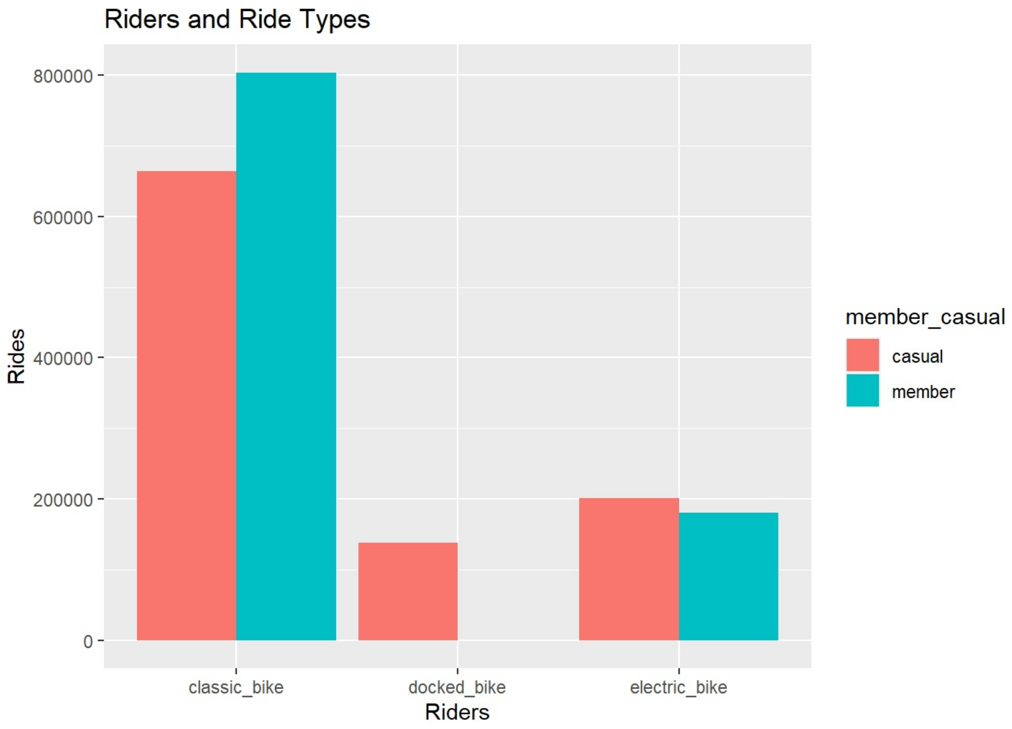
Rename columns.

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

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual 664374
## 2   docked_bike        casual 138096
## 3 electric_bike        casual 201221
## 4  classic_bike        member 803775
## 5   docked_bike        member      0
## 6 electric_bike        member 180040
```

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(q3_2021, “q3_2021.csv”)