Cyclistic Case Study Q4_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 fourth 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)
library(lubridate)
library(data.table)
library(ggplot2)
library(anytime)
```

Import data from local drive.

```
Oct21 <- read_csv("202110-divvy-tripdata.csv")
Nov21 <- read_csv("202111-divvy-tripdata.csv")
Dec21 <- read_csv("202112-divvy-tripdata.csv")
```

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(Oct21)
```

```
##  [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(Nov21)
```

```
##  [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(Dec21)
```

```
##  [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.

```
q4_2021 <- bind_rows(Oct21, Nov21, Dec21)
```

```
View(q4_2021)
```

```
nrow(q4_2021)
```

```
## [1] 1238744
```

```
dim(q4_2021)
```

```
## [1] 1238744      13
```

```
head(q4_2021)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 620BC6107255B… electr… 2021-10-22 12:46:42 2021-10-22 12:49:50 Kingsb… KA1503…
## 2 4471C70731AB2… electr… 2021-10-21 09:12:37 2021-10-21 09:14:14 <NA>    <NA>
## 3 26CA69D43D15E… electr… 2021-10-16 16:28:39 2021-10-16 16:36:26 <NA>    <NA>
## 4 362947F0437E1… electr… 2021-10-16 16:17:48 2021-10-16 16:19:03 <NA>    <NA>
## 5 BB731DE2F2EC5… electr… 2021-10-20 23:17:54 2021-10-20 23:26:10 <NA>    <NA>
## 6 7176307BBC097… electr… 2021-10-21 16:57:37 2021-10-21 17:11:58 <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(q4_2021)
```

```
## # A tibble: 6 × 13
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 92BBAB97D1683… electr… 2021-12-24 15:42:09 2021-12-24 19:29:35 Canal … 13341
## 2 847431F3D5353… electr… 2021-12-12 13:36:55 2021-12-12 13:56:08 Canal … 13341
## 3 CF407BBC3B9FA… electr… 2021-12-06 19:37:50 2021-12-06 19:44:51 Canal … 13341
## 4 60BB69EBF5440… electr… 2021-12-02 08:57:04 2021-12-02 09:05:21 Canal … 13341
## 5 C414F654A2863… electr… 2021-12-13 09:00:26 2021-12-13 09:14:39 Lawnda… 362.0
## 6 37AC57E34B2E7… classi… 2021-12-13 08:45:32 2021-12-13 08:49:09 Michig… TA1309…
## # … 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(q4_2021)
```

```
##    ride_id          rideable_type        started_at
##  Length:1238744     Length:1238744     Min.   :2021-10-01 00:00:09.00
##  Class :character   Class :character   1st Qu.:2021-10-14 18:54:24.25
##  Mode  :character   Mode  :character   Median :2021-10-31 11:33:29.50
##                                        Mean   :2021-11-04 20:35:06.04
##                                        3rd Qu.:2021-11-22 21:52:28.25
##                                        Max.   :2021-12-31 23:59:48.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2021-10-01 00:03:11.00   Length:1238744     Length:1238744
##  1st Qu.:2021-10-14 19:08:30.75   Class :character   Class :character
##  Median :2021-10-31 11:51:44.50   Mode  :character   Mode  :character
##  Mean   :2021-11-04 20:52:02.20
##  3rd Qu.:2021-11-22 22:09:58.75
##  Max.   :2022-01-03 17:32:18.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:1238744     Length:1238744     Min.   :41.64   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.   :41.39   Min.   :-88.97   Length:1238744
##  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.   :42.13   Max.   :-87.52
##  NA's   :819     NA's   :819
```

```
str(q4_2021)
```

```
## spc_tbl_ [1,238,744 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1238744] "620BC6107255BF4C" "4471C70731AB2E45" "26CA69D43D15EE14" "362947F0437E1
514" ...
##  $ rideable_type     : chr [1:1238744] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:1238744], format: "2021-10-22 12:46:42" "2021-10-21 09:12:37" ...
##  $ ended_at          : POSIXct[1:1238744], format: "2021-10-22 12:49:50" "2021-10-21 09:14:14" ...
##  $ start_station_name: chr [1:1238744] "Kingsbury St & Kinzie St" NA NA NA ...
##  $ start_station_id  : chr [1:1238744] "KA1503000043" NA NA NA ...
##  $ end_station_name  : chr [1:1238744] NA NA NA NA ...
##  $ end_station_id    : chr [1:1238744] NA NA NA NA ...
##  $ start_lat         : num [1:1238744] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:1238744] -87.6 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:1238744] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:1238744] -87.6 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:1238744] "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.

```
q4_2021$date <- as.Date(q4_2021$started_at)
q4_2021$month <- format(as.Date(q4_2021$date), "%m")
q4_2021$month <- month.name[as.numeric(q4_2021$month)]
q4_2021$day <- format(as.Date(q4_2021$date), "%d")
q4_2021$year <- format(as.Date(q4_2021$date), "%Y")
q4_2021$day_of_week <- format(as.Date(q4_2021$date), "%A")
q4_2021$ride_length <- difftime(q4_2021$ended_at,q4_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(q4_2021$ride_length)
```

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

Recheck *ride_length* data type.

```
q4_2021$ride_length <- as.numeric(as.character(q4_2021$ride_length))
is.numeric(q4_2021$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
q4_2021 <- na.omit(q4_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.

```
q4_2021 <- subset(q4_2021, nchar(as.character(ride_id)) == 16)
```

Remove rows with the *ride_length* less than 60 seconds 1 minute.

```
q4_2021 <- subset (q4_2021, ride_length > 59)
```

STEP FOUR: ANALYZE DATA

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

```
mean(q4_2021$ride_length)
```

```
## [1] 973.1662
```

```
median(q4_2021$ride_length)
```

```
## [1] 588
```

```
max(q4_2021$ride_length)
```

```
## [1] 2442301
```

```
min(q4_2021$ride_length)
```

```
## [1] 60
```

Run a statistical summary of the *ride_length*.

```
summary(q4_2021$ride_length)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
##      60.0     349.0     588.0     973.2    1026.0 2442301.0
```

Compare the members and casual users

```
aggregate(q4_2021$ride_length ~ q4_2021$member_casual, FUN = mean)
```

```
##   q4_2021$member_casual q4_2021$ride_length
## 1                casual           1527.0706
## 2                member            693.3502
```

```
aggregate(q4_2021$ride_length ~ q4_2021$member_casual, FUN = median)
```

```
##   q4_2021$member_casual q4_2021$ride_length
## 1                casual                 806
## 2                member                 508
```

```
aggregate(q4_2021$ride_length ~ q4_2021$member_casual, FUN = max)
```

```
##   q4_2021$member_casual q4_2021$ride_length
## 1                casual             2442301
## 2                member               87634
```

```
aggregate(q4_2021$ride_length ~ q4_2021$member_casual, FUN = min)
```

```
##   q4_2021$member_casual q4_2021$ride_length
## 1                casual                  60
## 2                member                  60
```

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

```
aggregate(q4_2021$ride_length ~ q4_2021$member_casual + q4_2021$day_of_week, FUN = mean)
```

```
##    q4_2021$member_casual q4_2021$day_of_week q4_2021$ride_length
## 1                 casual              Friday           1427.9125
## 2                 member              Friday            680.4321
## 3                 casual              Monday           1476.6850
## 4                 member              Monday            659.2894
## 5                 casual            Saturday           1686.1116
## 6                 member            Saturday            785.6137
## 7                 casual              Sunday           1851.4760
## 8                 member              Sunday            786.2086
## 9                 casual            Thursday           1267.0465
## 10                member            Thursday            648.8842
## 11                casual             Tuesday           1280.5640
## 12                member             Tuesday            658.3114
## 13                casual           Wednesday           1273.8111
## 14                member           Wednesday            667.3871
```

Sort the days of the week in order.

```
q4_2021$day_of_week <- ordered(q4_2021$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(q4_2021$ride_length ~ q4_2021$member_casual + q4_2021$day_of_week, FUN = mean)

head(x)
```

```
##   q4_2021$member_casual q4_2021$day_of_week q4_2021$ride_length
## 1                casual              Sunday           1851.4760
## 2                member              Sunday            786.2086
## 3                casual              Monday           1476.6850
## 4                member              Monday            659.2894
## 5                casual             Tuesday           1280.5640
## 6                member             Tuesday            658.3114
```

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

```
y <- q4_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           57480            1851.
## 2 casual              2           29595            1477.
## 3 casual              3           32745            1281.
## 4 casual              4           33677            1274.
## 5 casual              5           29866            1267.
## 6 casual              6           44701            1428.
```

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

```
table(q4_2021$member_casual)
```

```
## 
## casual member 
## 301275 596383
```

```
table(q4_2021$rideable_type)
```

```
## 
##  classic_bike   docked_bike electric_bike 
##        562130         34909        300619
```

```
table(q4_2021$day_of_week)
```

```
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##    123369    111319    131671    131883    112884    132024    154508
```

```
table(q4_2021$month)
```

```
##
## December November  October
##   174005   252189   471464
```

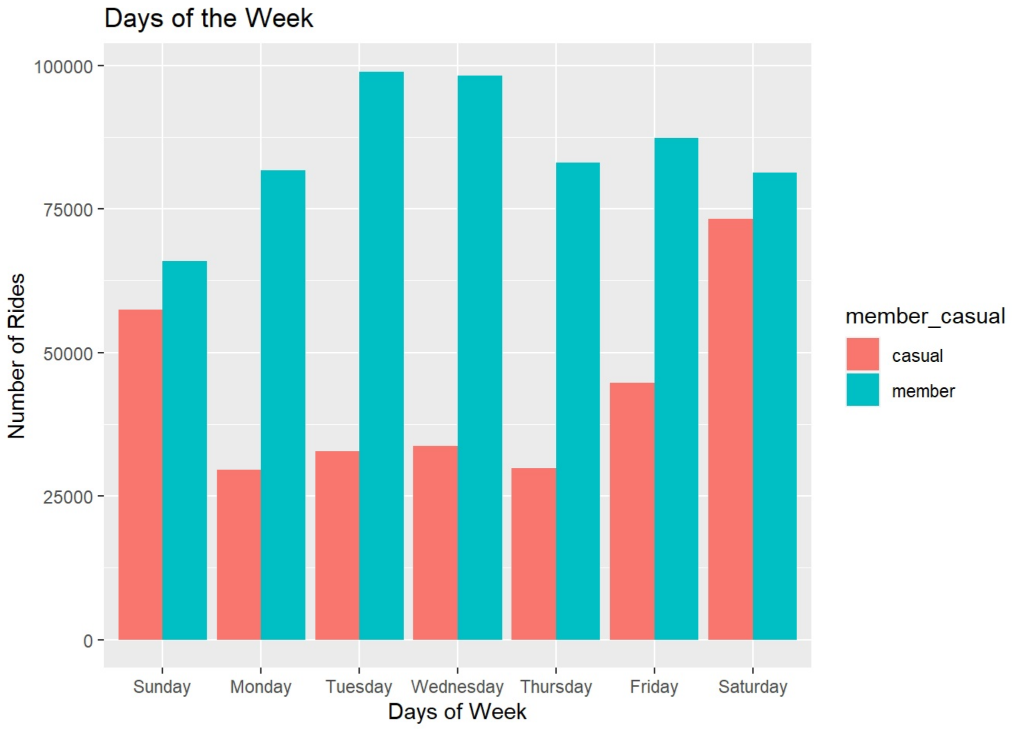
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

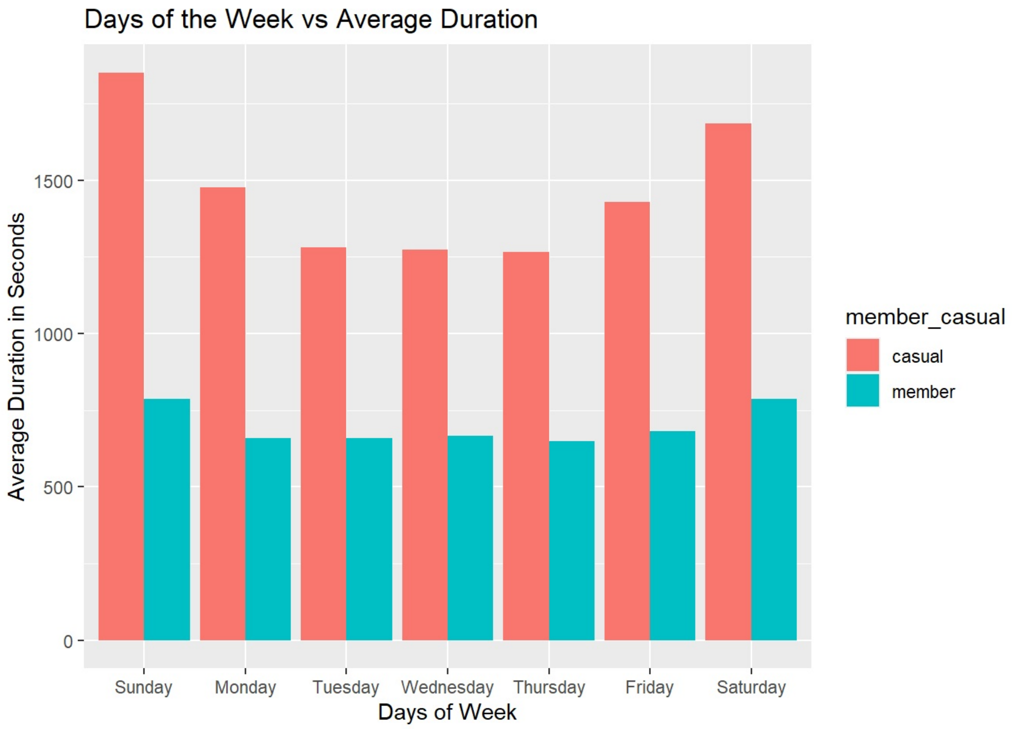
Plot the number of rides by user type during the week.

```
q4_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 = "Days of Week",
     y= "Number of Rides",
     title= "Days of the Week")
```



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

```
q4_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 = "Days 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(q4_2021$day_of_week,q4_2021$member_casual))
```

Rename columns

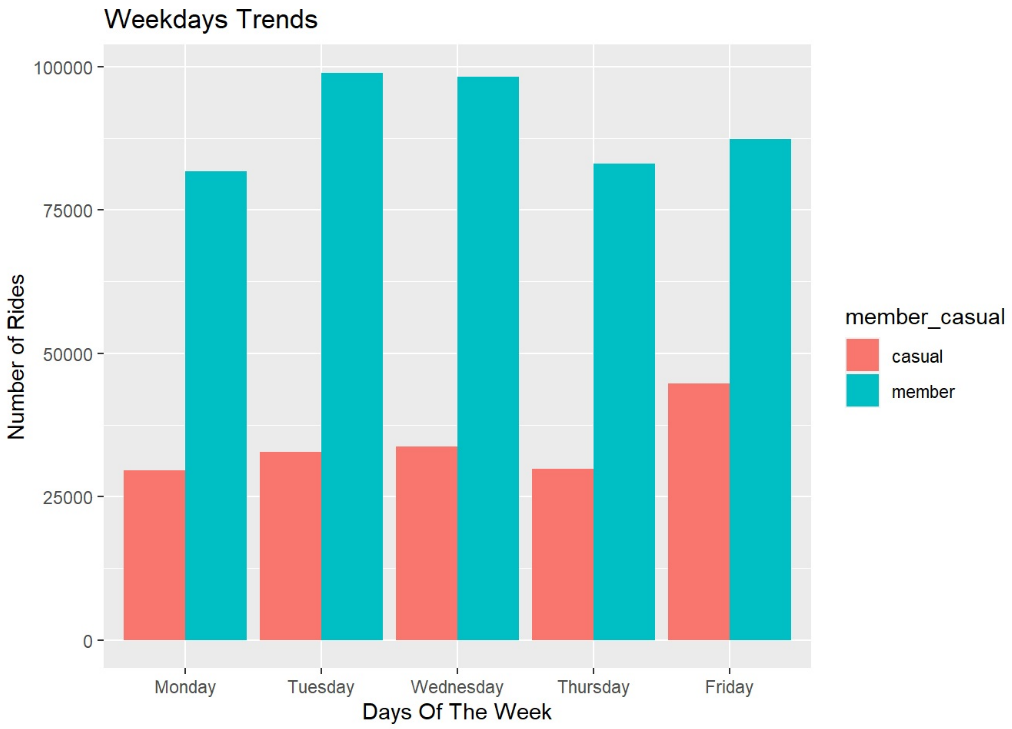
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)

head(mc)
```

```
##   day_of_week member_casual  Freq
## 1      Sunday        casual 57480
## 2      Monday        casual 29595
## 3     Tuesday        casual 32745
## 4   Wednesday        casual 33677
## 5    Thursday        casual 29866
## 6      Friday        casual 44701
```

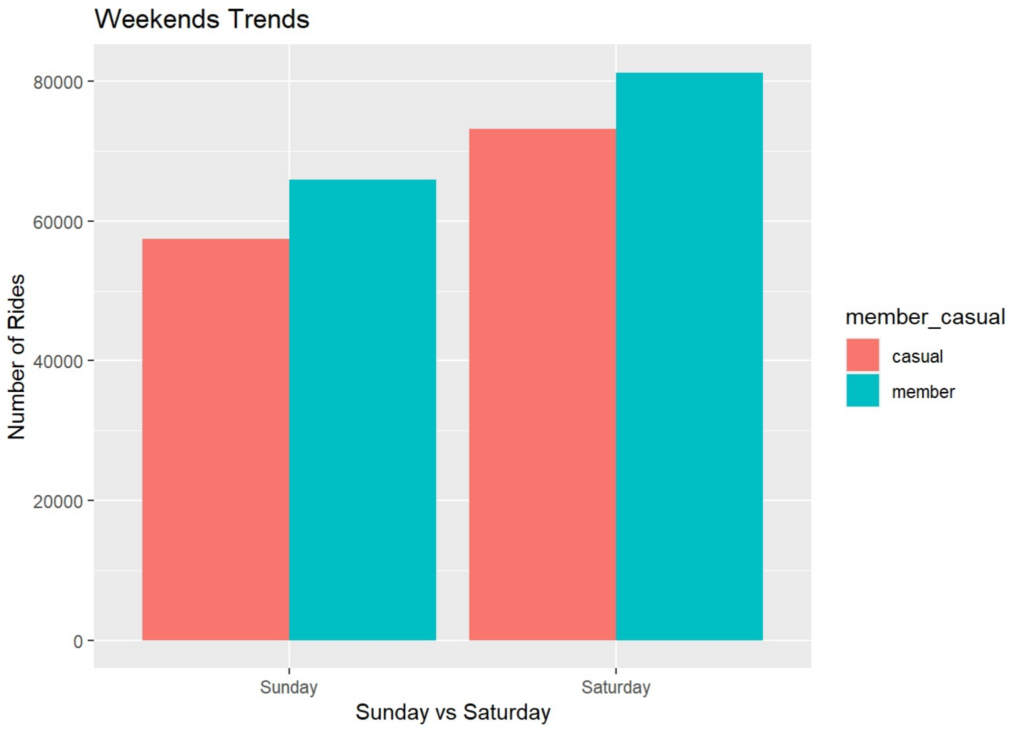
Weekday trends (Monday through Friday).

```
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= "Days Of The Week",
       y = "Number of 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= "Sunday vs Saturday",
       y = "Number of Rides")
```



Create dataframe for member and casual riders vs ride type

```
rt<- as.data.frame(table(q4_2021$rideable_type,q4_2021$member_casual))
```

Rename columns.

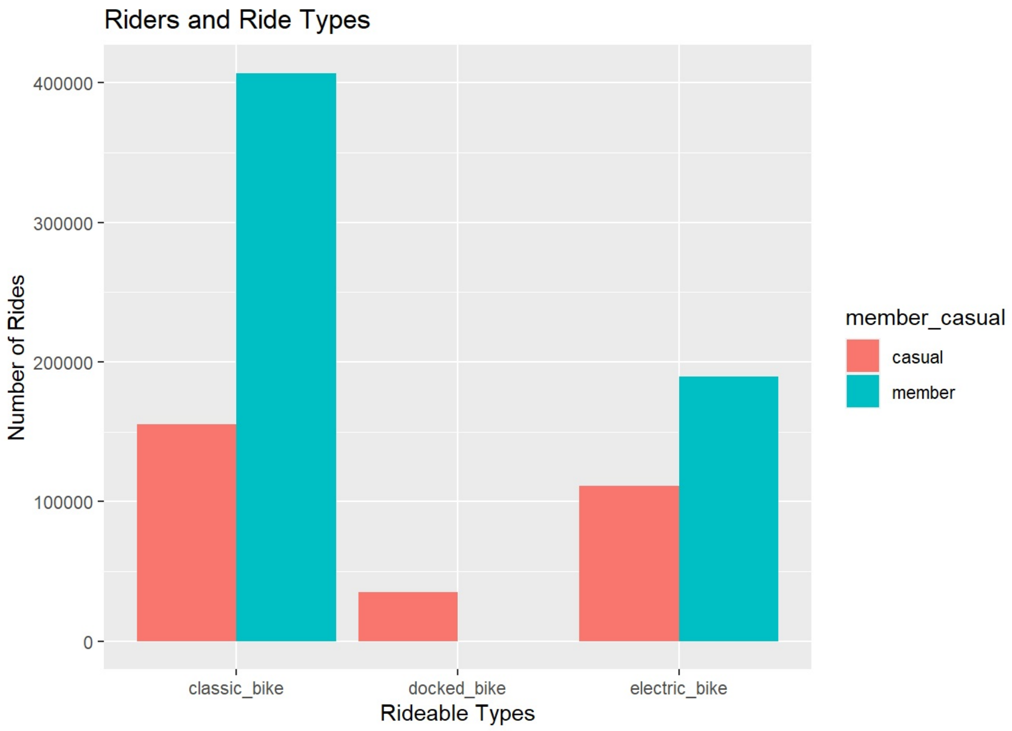
```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)

head(rt)
```

```
##   rideable_type member_casual   Freq
## 1  classic_bike        casual 155180
## 2   docked_bike        casual  34909
## 3 electric_bike        casual 111186
## 4  classic_bike        member 406950
## 5   docked_bike        member      0
## 6 electric_bike        member 189433
```

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= "Rideable Types",
       y = "Number of Rides")
```



Create vector of month names for Q4 2021

```
q4_months <- c("October", "November", "December")
```

Subset month.name to include only Q4 2021 months

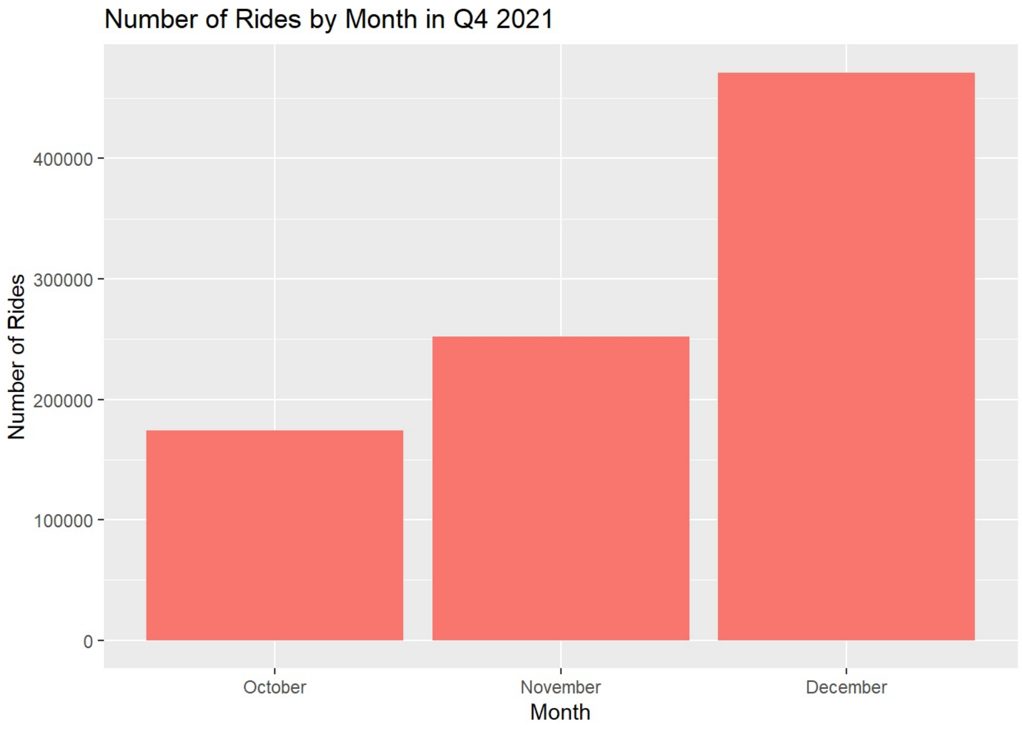
```
q4_month_names <- month.name[match(q4_months, month.name)]
```

Create trips_by_month dataframe with only Q4 2021 months

```
trips_by_month <- data.frame(month = q4_month_names, count = table(q4_2021$month))
```

Set the levels of the month variable in the trips_by_month dataframe

```
trips_by_month$month <- factor(trips_by_month$month, levels = c("October", "November", "December"))
ggplot(trips_by_month, aes(x = month, y = count.Freq)) +
geom_bar(stat = "identity", fill = "#F8766D") +
  labs(x = "Month", y = "Number of Rides", title = "Number of Rides by Month in Q4 2021")
```



Number of Rides by Month in Q4 2021

STEP SIX: EXPORT ANALYZED DATA

Save the analyzed data as a new file. fwrite(q4_2021, “q4_2021.csv”)