Cyclistic Case Study 2021 All Trips

Hezar K

2022-11-29

This analysis is for Cyclistic Case Study for Google Data Analytics Course. This is an analysis for the year of 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)

library(lubridate)

library(data.table)

library(ggplot2)

library(anytime)
```

Import data from local drive.

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()**.(To reduce clutter I have removed colnames output from Feb21-Dec21, because all tables have the same column names.

```
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)

colnames(Mar21)

colnames(Apr21)

colnames(May21)

colnames(Jun21)

colnames(Jul21)

colnames(Aug21)

colnames(Sep21)

colnames(Oct21)

colnames(Nov21)

colnames(Dec21)
```

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

```
all_trips <- bind_rows(Jan21, Feb21, Mar21, Apr21, May21, Jun21, Jul21,
Aug21, Sep21, Oct21, Nov21, Dec21)
```

View(all_trips)

```
nrow(all_trips)
```

```
## [1] 5595063
```

```
dim(all_trips)
```

```
## [1] 5595063      13
```

```
head(all_trips)
```

```
## # 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(all_trips)
```

```
## # 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(all_trips)
```

```
##    ride_id          rideable_type       started_at                    
##  Length:5595063     Length:5595063     Min.   :2021-01-01 00:02:05.00  
##  Class :character   Class :character   1st Qu.:2021-06-06 23:52:40.00  
##  Mode  :character   Mode  :character   Median :2021-08-01 01:52:11.00  
##                                        Mean   :2021-07-29 07:41:02.63  
##                                        3rd Qu.:2021-09-24 16:36:16.00  
##                                        Max.   :2021-12-31 23:59:48.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2021-01-01 00:08:39.00   Length:5595063     Length:5595063    
##  1st Qu.:2021-06-07 00:44:21.00   Class :character   Class :character  
##  Median :2021-08-01 02:21:55.00   Mode  :character   Mode  :character  
##  Mean   :2021-07-29 08:02:58.75                                        
##  3rd Qu.:2021-09-24 16:54:05.50                                        
##  Max.   :2022-01-03 17:32:18.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:5595063     Length:5595063     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:5595063    
##  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   :4771    NA's   :4771
```

```
str(all_trips)
```

```
## spc_tbl_ [5,595,063 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)

##  $ ride_id           : chr [1:5595063] "E19E6F1B8D4C42ED"
"DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA453A75AE377DB" ...

##  $ rideable_type     : chr [1:5595063] "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...

##  $ started_at        : POSIXct[1:5595063], format: "2021-01-23 16:14:19"
"2021-01-27 18:43:08" ...

##  $ ended_at          : POSIXct[1:5595063], format: "2021-01-23 16:24:44"
"2021-01-27 18:47:12" ...

##  $ start_station_name: chr [1:5595063] "California Ave & Cortez St"
"California Ave & Cortez St" "California Ave & Cortez St" "California Ave &
Cortez St" ...

##  $ start_station_id  : chr [1:5595063] "17660" "17660" "17660" "17660" ...

##  $ end_station_name  : chr [1:5595063] NA NA NA NA ...

##  $ end_station_id    : chr [1:5595063] NA NA NA NA ...

##  $ start_lat         : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...

##  $ start_lng         : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...

##  $ end_lat           : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...

##  $ end_lng           : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...

##  $ member_casual     : chr [1:5595063] "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.

```
all_trips$date <- as.Date(all_trips$started_at)
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$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(all_trips$ride_length)
## [1] FALSE
```

Recheck *ride_length* data type.

```
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
is.numeric(all_trips$ride_length)
## [1] TRUE
```

STEP THREE: CLEAN DATA

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

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

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.

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

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

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

STEP FOUR: ANALYZE DATA

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

```
mean(all_trips$ride_length)
## [1] 1308.878
median(all_trips$ride_length)
## [1] 732
max(all_trips$ride_length)
## [1] 3356649
min(all_trips$ride_length)
## [1] 2
```

Run a statistical summary of the *ride_length*.

```
summary(all_trips$ride_length)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       2     417     732    1309    1327 3356649
```

Compare the members and casual users

```
aggregate(all_trips$ride_length ~ all_trips$member_casual, FUN = mean)
##   all_trips$member_casual all_trips$ride_length
## 1                  casual             1950.7623
## 2                  member              791.1678
aggregate(all_trips$ride_length ~ all_trips$member_casual, FUN = median)
```

```
##   all_trips$member_casual all_trips$ride_length
## 1                  casual                  1000
## 2                  member                   583
aggregate(all_trips$ride_length ~ all_trips$member_casual, FUN = max)
##   all_trips$member_casual all_trips$ride_length
## 1                  casual               3356649
## 2                  member                 89738
aggregate(all_trips$ride_length ~ all_trips$member_casual, FUN = min)
##   all_trips$member_casual all_trips$ride_length
## 1                  casual                     2
## 2                  member                     2
```

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

```
aggregate(all_trips$ride_length ~ all_trips$member_casual +
all_trips$day_of_week, FUN = mean)
##    all_trips$member_casual all_trips$day_of_week all_trips$ride_length
## 1                   casual                Friday             1855.2912
## 2                   member                Friday              767.6783
## 3                   casual                Monday             1958.3713
## 4                   member                Monday              763.6223
## 5                   casual              Saturday             2092.5141
## 6                   member              Saturday              889.3566
## 7                   casual                Sunday             2256.3895
## 8                   member                Sunday              911.8428
## 9                   casual              Thursday             1680.8100
## 10                  member              Thursday              741.3387
## 11                  casual               Tuesday             1728.3218
## 12                  member               Tuesday              743.3806
## 13                  casual             Wednesday             1696.3180
## 14                  member             Wednesday              747.6129
```

Sort the days of the week in order.

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


head(x)
```

```
##   all_trips$member_casual all_trips$day_of_week all_trips$ride_length
## 1                  casual                Sunday             2256.3895
## 2                  member                Sunday              911.8428
## 3                  casual                Monday             1958.3713
## 4                  member                Monday              763.6223
## 5                  casual               Tuesday             1728.3218
## 6                  member               Tuesday              743.3806
```

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

```
y <- all_trips %>%
  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          403723            2256.
## 2 casual              2          228915            1958.
## 3 casual              3          214917            1728.
## 4 casual              4          218111            1696.
## 5 casual              5          224184            1681.
## 6 casual              6          290011             1855.
```

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

```
table(all_trips$member_casual)
##
##  casual  member
## 2048141 2539393
table(all_trips$rideable_type)
##
##  classic_bike   docked_bike electric_bike
##       3241407        312036       1034091
table(all_trips$day_of_week)
##
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##    714879    575341    602985    615726    597594    655727    825282
table(all_trips$month)
##
##     01     02     03     04     05     06     07     08     09     10
11
##  83498  42986 205674 298169 450906 608694 692193 674301 621012 477914
255841
##     12
## 176346
```

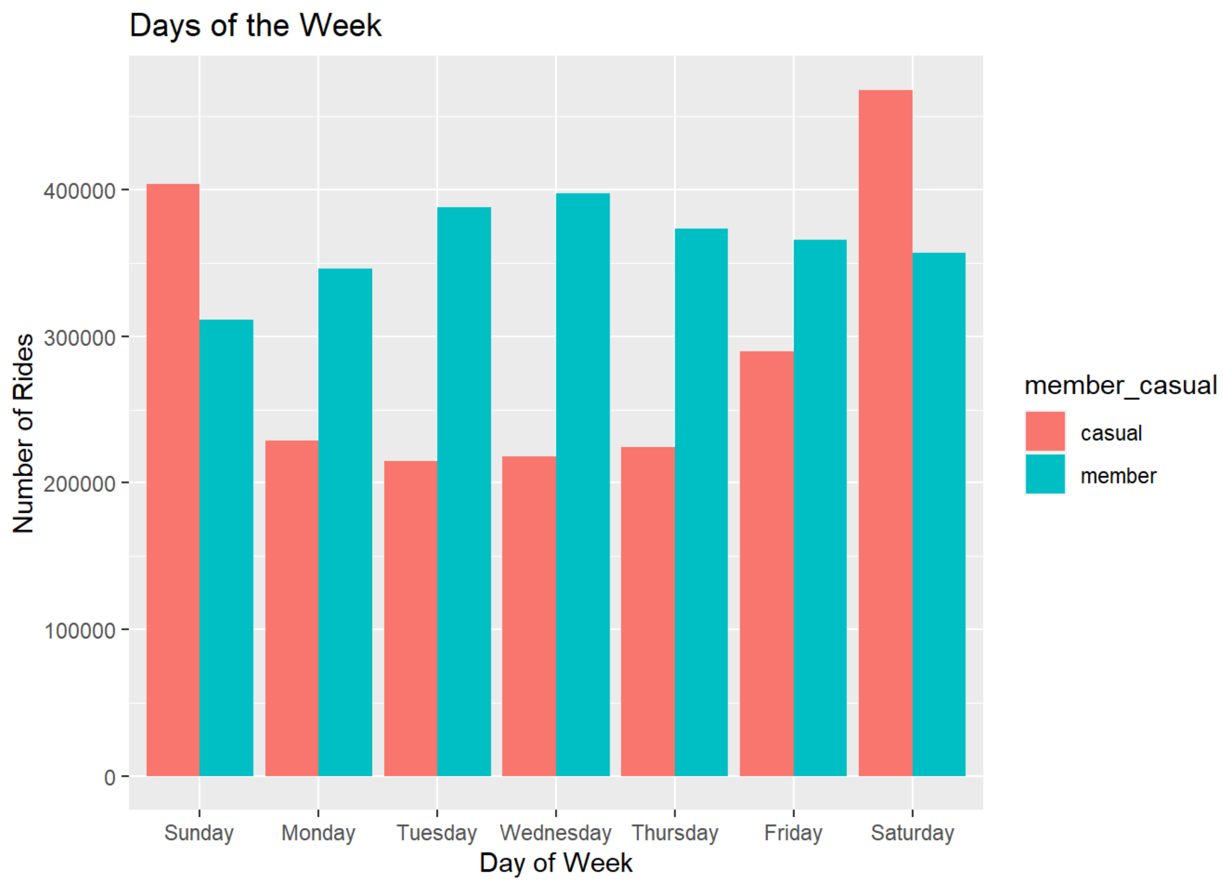
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

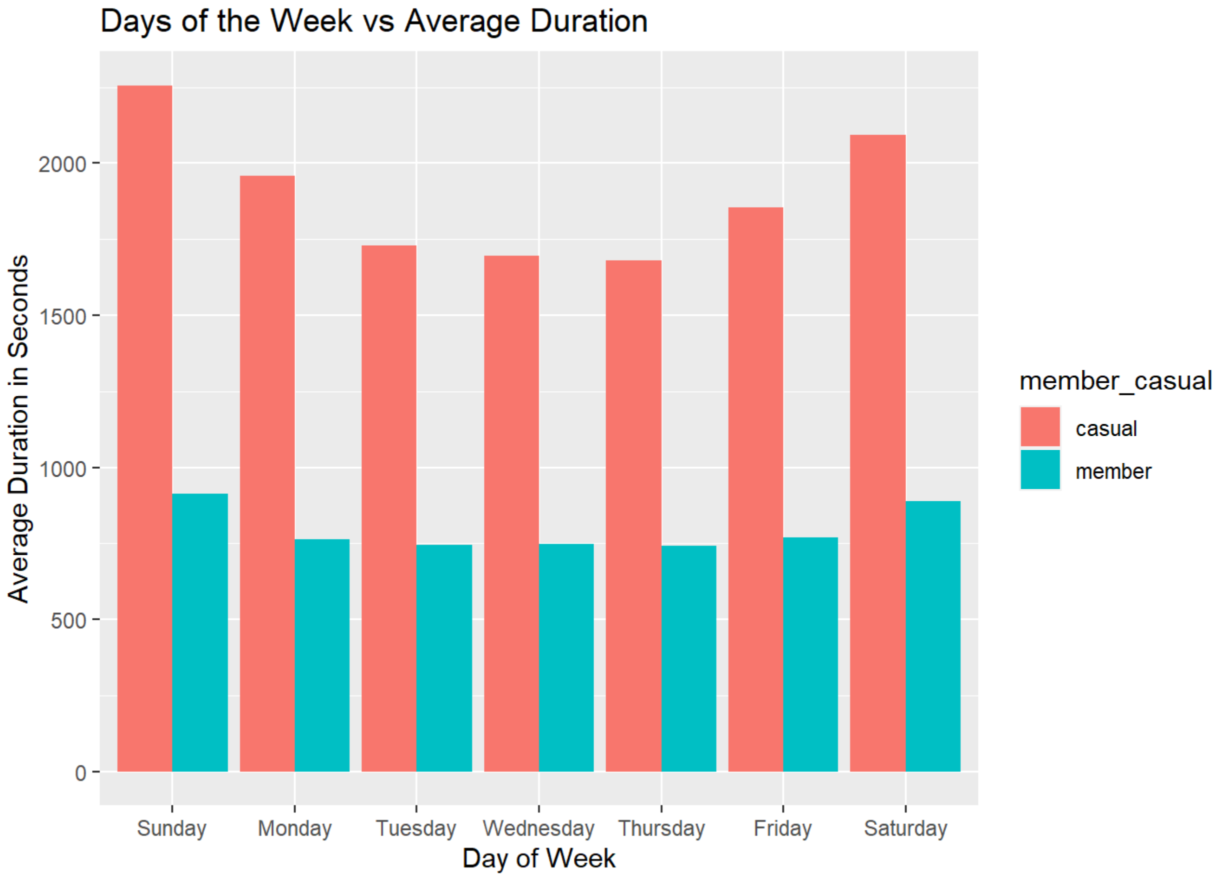
Plot the number of rides by user type during the week.

```
all_trips %>%
  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.

```
all_trips %>%

  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.

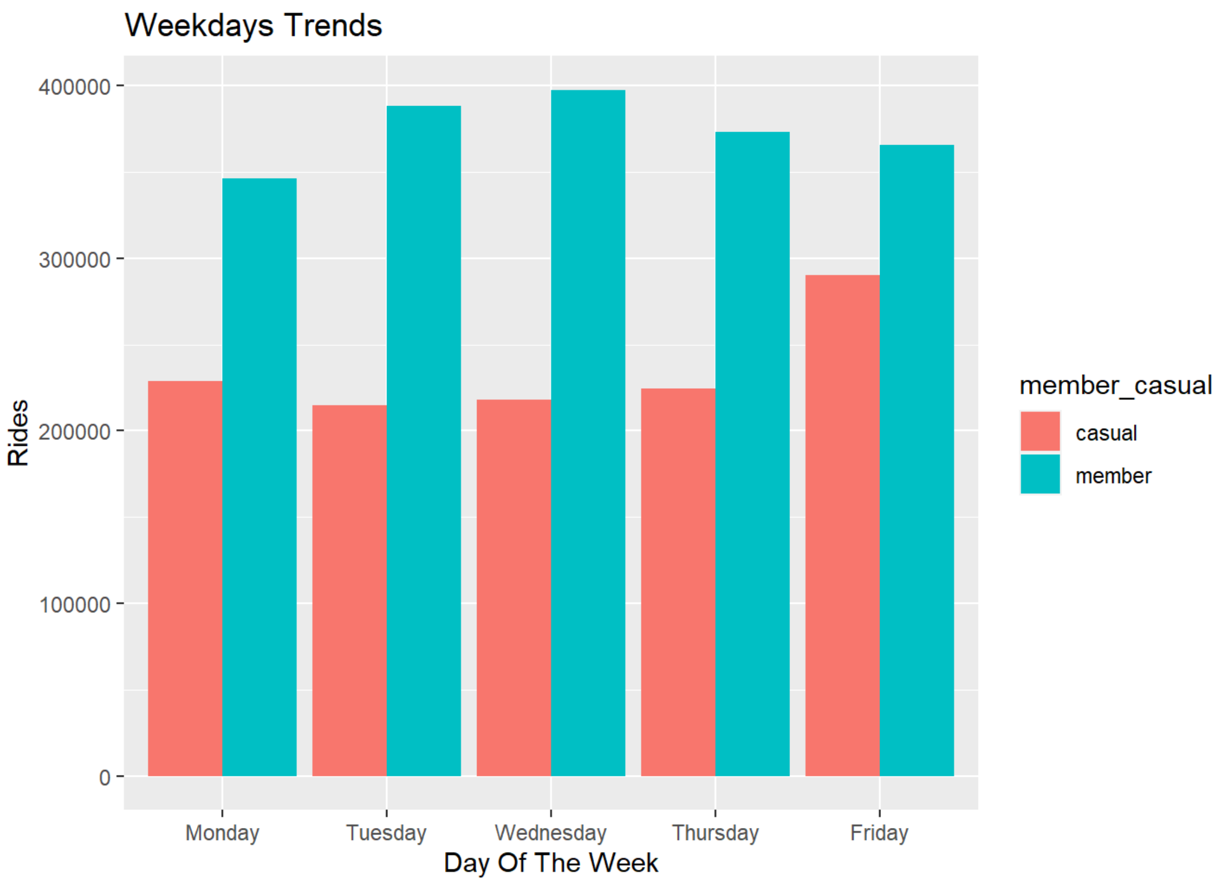
```
mc<- as.data.frame(table(all_trips$day_of_week,all_trips$member_casual))
```

Rename columns

```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)
head(mc)
##   day_of_week member_casual   Freq
## 1      Sunday        casual 403723
## 2      Monday        casual 228915
## 3     Tuesday        casual 214917
## 4   Wednesday        casual 218111
## 5    Thursday        casual 224184
## 6      Friday        casual 290011
```

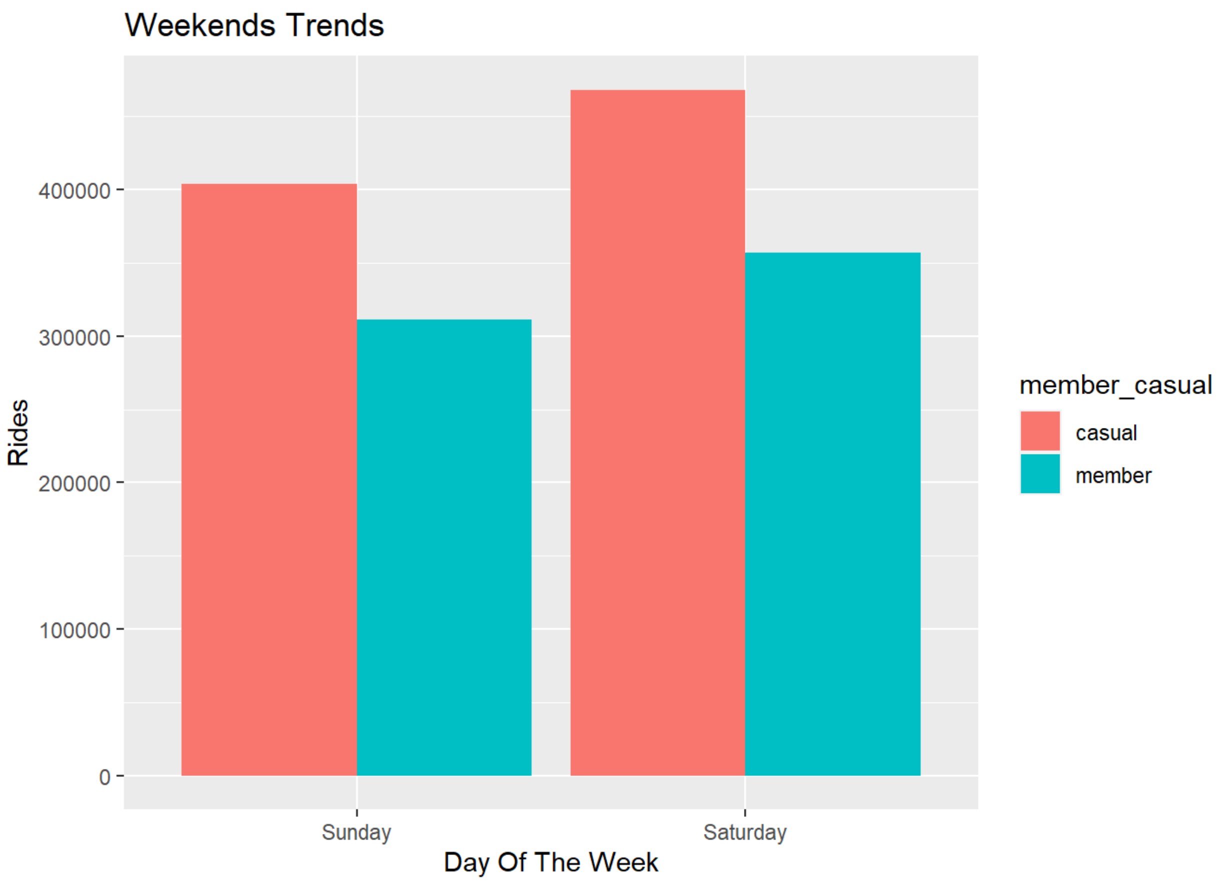
Weekday trends (Monday through Friday).

```
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

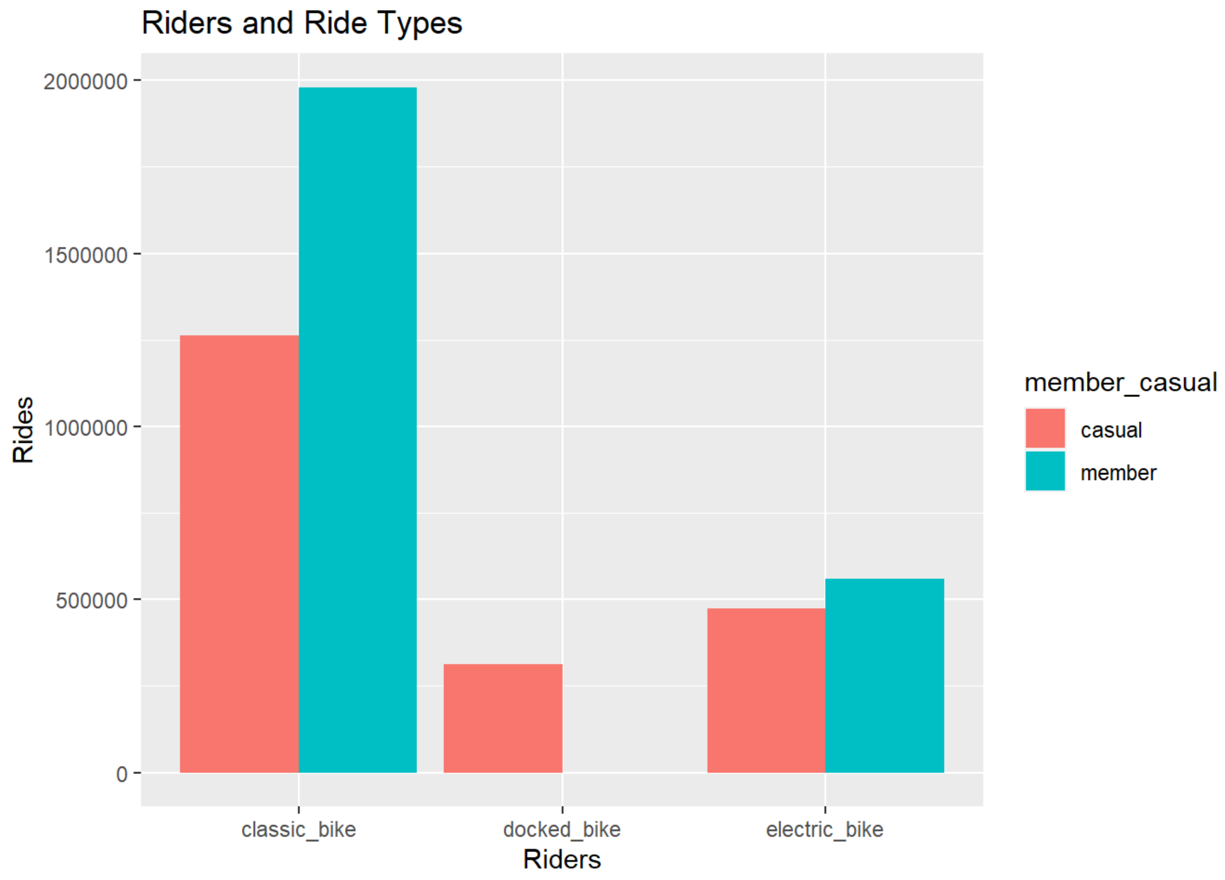
```
rt<- as.data.frame(table(all_trips$rideable_type,all_trips$member_casual))
```

Rename columns.

```
rt<-rename(rt, rideable_type = Var1, member_casual = Var2)
head(rt)
##   rideable_type member_casual    Freq
## 1  classic_bike        casual 1261409
## 2   docked_bike        casual  312035
## 3 electric_bike        casual  474697
## 4  classic_bike        member 1979998
## 5   docked_bike        member       1
## 6 electric_bike        member  559394
```

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(all_trips, “all_trips.csv”)