

# Cyclistic Case Study Feb21

Hezar K

2022-11-29

This is an analysis for Cyclistic Case Study for Google Data Analytics Course. This is an analysis for February 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.

```
Feb21 <- read_csv("C:/Users/theby/Documents/202102-divvy-tripdata.csv")
```

## 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(Feb21)

```
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"
```

```
nrow(Feb21)
```

```
## [1] 49622
```

```
dim(Feb21)
```

```
## [1] 49622    13
```

```
head(Feb21)
```

```
## # A tibble: 6 × 13
##   ride_id      ridea...1 started_at      ended_at      start...2 start...3
##   <chr>        <chr>   <dtm>         <dtm>         <chr>      <chr>
## 1 89E7AA6C29227... classi... 2021-02-12 16:14:56 2021-02-12 16:21:43 Glenwo... 525
## 2 0FEFDE2603568... classi... 2021-02-14 17:52:38 2021-02-14 18:12:09 Glenwo... 525
## 3 E6159D746B2DB... electr... 2021-02-09 19:10:18 2021-02-09 19:19:10 Clark ... KA1503...
## 4 B32D3199F1C2E... classi... 2021-02-02 17:49:41 2021-02-02 17:54:06 Wood S... 637
## 5 83E463F23575F... electr... 2021-02-23 15:07:23 2021-02-23 15:22:37 State ... 13216
## 6 BDAA7E3494E8D... electr... 2021-02-24 15:43:33 2021-02-24 15:49:05 Fairba... 18003
## # ... 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(Feb21)
```

```
## # A tibble: 6 × 13
##   ride_id      ridea...1 started_at      ended_at      start...2 start...3
##   <chr>      <chr>   <dtm>      <dtm>      <chr>   <chr>
## 1 F1E4C456F8F88... electr... 2021-02-12 12:47:42 2021-02-12 13:23:32 Burnha... 15545
## 2 7ED482EE6C9F5... classi... 2021-02-20 15:25:32 2021-02-20 15:59:45 Wester... TA1307...
## 3 203DF22F090C1... classi... 2021-02-09 08:54:38 2021-02-09 09:08:19 Frankl... 13017
## 4 940161523673F... docked... 2021-02-27 14:46:06 2021-02-27 15:00:49 Frankl... 13017
## 5 C5538FFA492A7... classi... 2021-02-09 11:44:17 2021-02-09 11:46:13 Frankl... 13017
## 6 EB4CA525B953E... electr... 2021-02-04 10:26:44 2021-02-04 10:31:21 Frankl... 13017
## # ... 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(Feb21)
```

```
##   ride_id      rideable_type      started_at
## Length:49622 Length:49622 Min. :2021-02-01 00:55:44.00
## Class :character Class :character 1st Qu.:2021-02-09 08:20:56.25
## Mode :character Mode :character Median :2021-02-22 13:17:53.00
## Mean :2021-02-18 01:16:52.85
## 3rd Qu.:2021-02-26 16:02:13.50
## Max. :2021-02-28 23:59:41.00
##
## ended_at      start_station_name start_station_id
## Min. :2021-02-01 01:22:48.00 Length:49622 Length:49622
## 1st Qu.:2021-02-09 08:36:02.50 Class :character Class :character
## Median :2021-02-22 13:39:20.50 Mode :character Mode :character
## Mean :2021-02-18 01:41:18.23
## 3rd Qu.:2021-02-26 16:19:32.75
## Max. :2021-03-05 15:11:45.00
##
## end_station_name end_station_id start_lat start_lng
## Length:49622 Length:49622 Min. :41.65 Min. : -87.77
## 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 : -87.64
## 3rd Qu.:41.93 3rd Qu.: -87.63
## Max. :42.06 Max. : -87.53
##
## end_lat end_lng member_casual
## Min. :41.54 Min. : -87.77 Length:49622
## 1st Qu.:41.88 1st Qu.: -87.66 Class :character
## Median :41.90 Median : -87.64 Mode :character
## Mean :41.90 Mean : -87.64
## 3rd Qu.:41.93 3rd Qu.: -87.63
## Max. :42.07 Max. : -87.53
## NA's :214 NA's :214
```

```
str(Feb21)
```

```
## spec_tbl_ [49,622 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ ride_id      : chr [1:49622] "89E7AA6C29227EFF" "0FEFDE2603568365" "E6159D746B2CDBB91" "B32D3199F1C2E75
B" ...
## $ rideable_type : chr [1:49622] "classic_bike" "classic_bike" "electric_bike" "classic_bike" ...
## $ started_at   : POSIXct[1:49622], format: "2021-02-12 16:14:56" "2021-02-14 17:52:38" ...
## $ ended_at     : POSIXct[1:49622], format: "2021-02-12 16:21:43" "2021-02-14 18:12:09" ...
## $ start_station_name: chr [1:49622] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Clark St & Lake St
" "Wood St & Chicago Ave" ...
## $ start_station_id : chr [1:49622] "525" "525" "KA1503000012" "637" ...
## $ end_station_name : chr [1:49622] "Sheridan Rd & Columbia Ave" "Bosworth Ave & Howard St" "State St & Rando
lph St" "Honore St & Division St" ...
## $ end_station_id   : chr [1:49622] "660" "16806" "TA1305000029" "TA1305000034" ...
## $ start_lat        : num [1:49622] 42 42 41.9 41.9 41.8 ...
## $ start_lng        : num [1:49622] -87.7 -87.7 -87.6 -87.7 -87.6 ...
## $ end_lat          : num [1:49622] 42 42 41.9 41.9 41.8 ...
## $ end_lng          : num [1:49622] -87.7 -87.7 -87.6 -87.7 -87.6 ...
## $ member_casual    : chr [1:49622] "member" "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.

```
Feb21$date <- as.Date(Feb21$started_at)
Feb21$month <- format(as.Date(Feb21$date), "%m")
Feb21$month <- month.name[as.numeric(Feb21$month)]
Feb21$day <- format(as.Date(Feb21$date), "%d")
Feb21$year <- format(as.Date(Feb21$date), "%Y")
Feb21$day_of_week <- format(as.Date(Feb21$date), "%A")
Feb21$ride_length <- difftime(Feb21$ended_at, Feb21$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(Feb21$ride_length)
```

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

Recheck *ride\_length* data type.

```
Feb21$ride_length <- as.numeric(as.character(Feb21$ride_length))
is.numeric(Feb21$ride_length)
```

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

### STEP THREE: CLEAN DATA

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

```
Feb21 <- na.omit(Feb21)
```

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.

```
Feb21 <- subset(Feb21, nchar(as.character(ride_id)) == 16)
```

Remove rows with the *ride\_length* less than 60 seconds or 1 minute.

```
Feb21 <- subset (Feb21, ride_length > 59)
```

### STEP FOUR: ANALYZE DATA

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

```
mean(Feb21$ride_length)
```

```
## [1] 1296.328
```

```
median(Feb21$ride_length)
```

```
## [1] 675
```

```
max(Feb21$ride_length)
```

```
## [1] 1807754
```

```
min(Feb21$ride_length)
```

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

Run a statistical summary of the *ride\_length*.

```
summary(Feb21$ride_length)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##         60     407     675    1296    1191 1807754
```

Compare the members and casual users

```
aggregate(Feb21$ride_length ~ Feb21$member_casual, FUN = mean)
```

```
##      Feb21$member_casual Feb21$ride_length
## 1                    casual      2862.8077
## 2                    member       901.9381
```

```
aggregate(Feb21$ride_length ~ Feb21$member_casual, FUN = median)
```

```
##      Feb21$member_casual Feb21$ride_length
## 1                    casual          1016
## 2                    member           621
```

```
aggregate(Feb21$ride_length ~ Feb21$member_casual, FUN = max)
```

```
##      Feb21$member_casual Feb21$ride_length
## 1                    casual      1807754
## 2                    member       88461
```

```
aggregate(Feb21$ride_length ~ Feb21$member_casual, FUN = min)
```

```
##      Feb21$member_casual Feb21$ride_length
## 1                    casual           60
## 2                    member           60
```

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

```
aggregate(Feb21$ride_length ~ Feb21$member_casual + Feb21$day_of_week, FUN = mean)
```

```
##      Feb21$member_casual Feb21$day_of_week Feb21$ride_length
## 1          casual      Friday      3772.8599
## 2          member      Friday       835.2875
## 3          casual      Monday      1917.9371
## 4          member      Monday       917.2421
## 5          casual      Saturday     3715.6154
## 6          member      Saturday     999.6027
## 7          casual      Sunday      2187.5168
## 8          member      Sunday     1009.9220
## 9          casual      Thursday     1341.7620
## 10         member      Thursday      813.5224
## 11         casual      Tuesday     2646.1375
## 12         member      Tuesday      913.2483
## 13         casual      Wednesday    1717.1872
## 14         member      Wednesday     867.2685
```

Sort the days of the week in order.

```
Feb21$day_of_week <- ordered(Feb21$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(Feb21$ride_length ~ Feb21$member_casual + Feb21$day_of_week, FUN = mean)

head(x)
```

```
##      Feb21$member_casual Feb21$day_of_week Feb21$ride_length
## 1          casual      Sunday      2187.5168
## 2          member      Sunday     1009.9220
## 3          casual      Monday      1917.9371
## 4          member      Monday       917.2421
## 5          casual      Tuesday     2646.1375
## 6          member      Tuesday      913.2483
```

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

```
y <- Feb21 %>%
  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            1194            2188.
## 2 casual            2             445            1918.
## 3 casual            3             822            2646.
## 4 casual            4             919            1717.
## 5 casual            5             836            1342.
## 6 casual            6            1206            3773.
```

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

```
table(Feb21$member_casual)
```

```
##
## casual member
##   8508  33793
```

```
table(Feb21$rideable_type)
```

```
##
## classic_bike  docked_bike electric_bike
##       34054       1259       6988
```

```
table(Feb21$day_of_week)
```

```
##
##   Sunday   Monday   Tuesday Wednesday Thursday   Friday   Saturday
##   4667     3873     5755     6622     5985     6775     8624
```

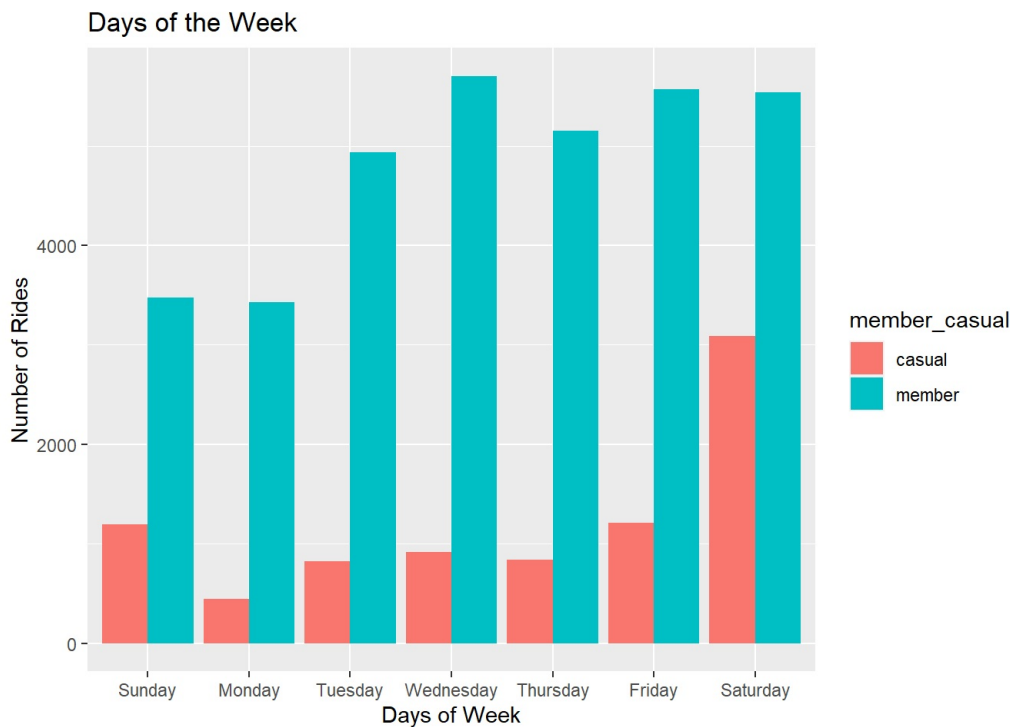
## STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

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

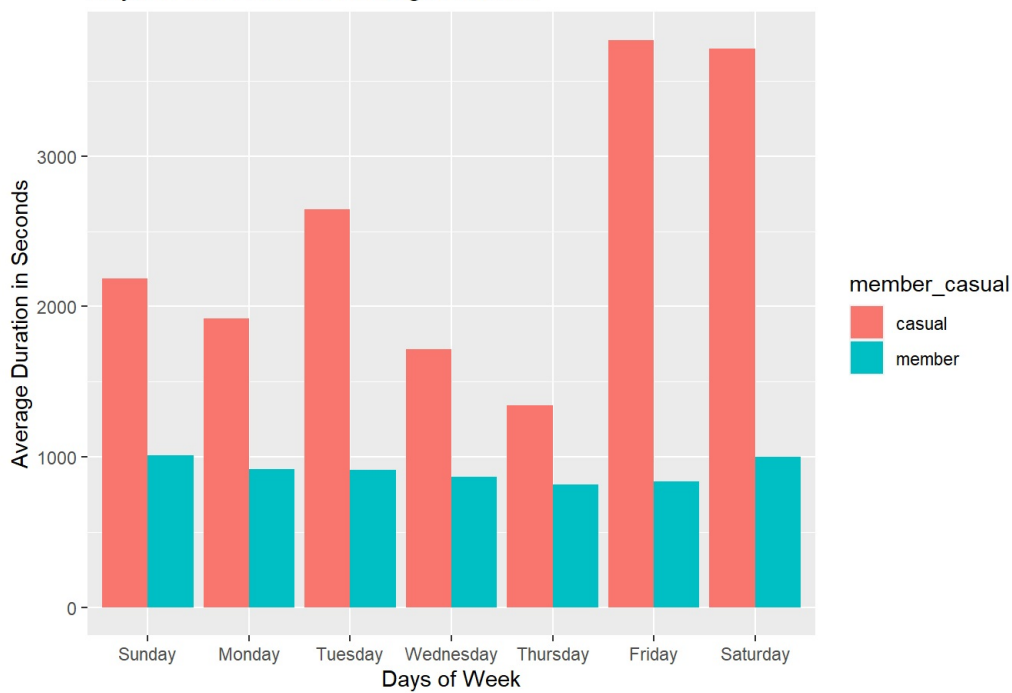
```
Feb21 %>%
  mutate(day_of_week) %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "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.

```
Feb21 %>%
  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")
```

Days of the Week vs Average Duration



Create new dataframe for plots for weekday trends vs weekend trends.

```
mc<- as.data.frame(table(Feb21$day_of_week, Feb21$member_casual))
```

Rename columns

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

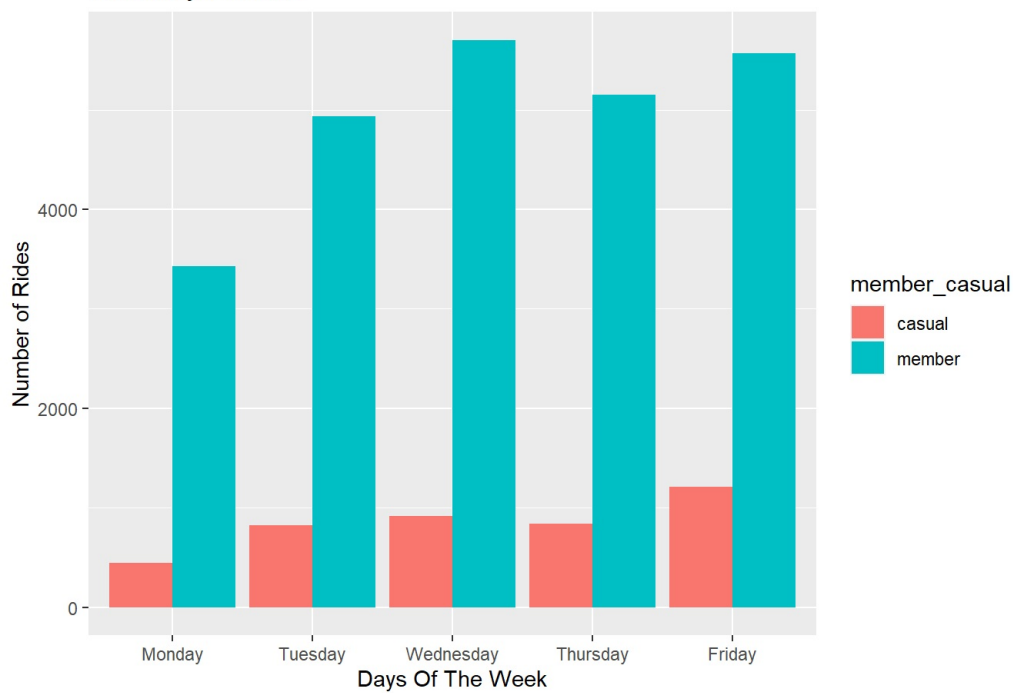
```
head(mc)
```

```
##   day_of_week member_casual Freq
## 1    Sunday          casual 1194
## 2    Monday          casual  445
## 3    Tuesday          casual  822
## 4   Wednesday          casual  919
## 5   Thursday          casual  836
## 6    Friday          casual 1206
```

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")
```

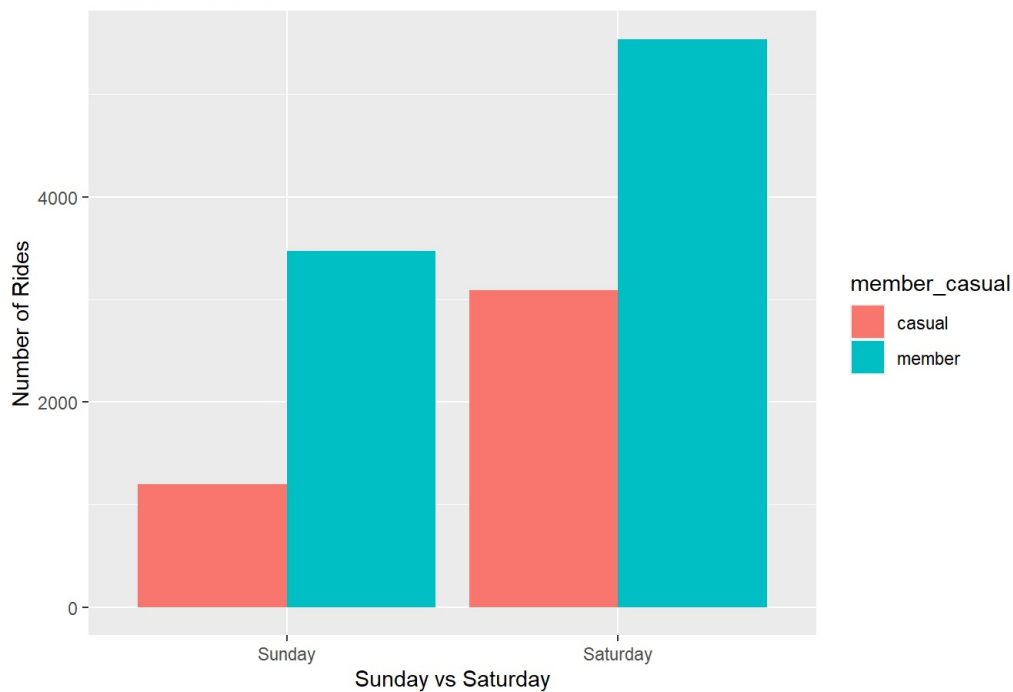
### Weekdays Trends



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")
```

### Weekends Trends



Create dataframe for member and casual riders vs ride type

```
rt<- as.data.frame(table(Feb21$rideable_type,Feb21$member_casual))
```

Rename columns.

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

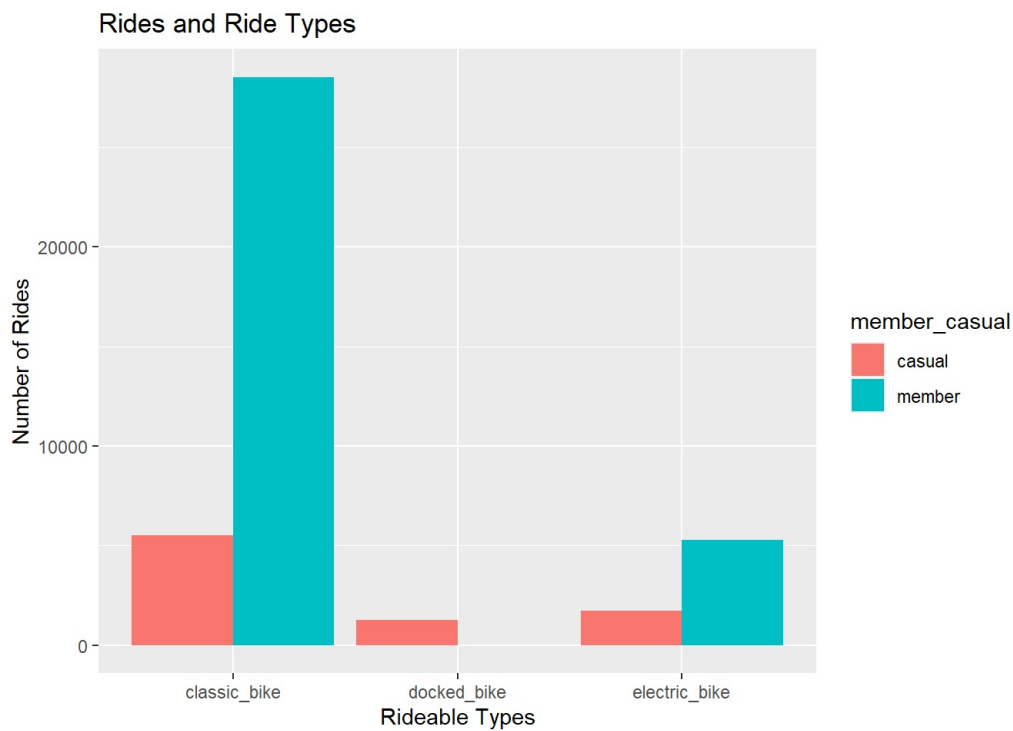
head(rt)
```



```
##   rideable_type member_casual Freq
## 1 classic_bike      casual  5525
## 2 docked_bike       casual  1259
## 3 electric_bike     casual  1724
## 4 classic_bike     member 28529
## 5 docked_bike      member    0
## 6 electric_bike    member  5264
```

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 = "Rides and Ride Types",
       x = "Rideable Types",
       y = "Number of Rides")
```



## STEP SIX: EXPORT ANALYZED DATA

Save the analyzed data as a new file. `fwrite(Feb21, "Feb21.csv")`