

Cyclistic Case Study Jun21

Hezar K

2022-11-29

This is an analysis for Cyclistic Case Study for Google Data Analytics Course. This is an analysis for June 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)
```

```
## — 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.

```
Jun21 <- read_csv("C:/Users/theby/Documents/202106-divvy-tripdata.csv")
```

```
## Rows: 729595 Columns: 13
## — Column specification —
## Delimiter: ","
## chr (9): ride_id, rideable_type, started_at, ended_at, start_station_name, s...
## dbl (4): start_lat, start_lng, end_lat, end_lng
##
## 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()`, 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(Jun21)

```
colnames(Jun21)
```

```
## [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(Jun21)
```

```
## [1] 729595
```

```
dim(Jun21)
```

```
## [1] 729595      13
```

```
head(Jun21)
```

```
## # A tibble: 6 × 13
##   ride_id      ridea...1 start...2 ended...3 start...4 start...5 end_s...6 end_s...7 start...8
##   <chr>         <chr>    <chr>    <chr>    <chr>    <chr>    <chr>    <chr>    <dbl>
## 1 99FEC93BA843F... electr... 6/13/2... 6/13/2... <NA>    <NA>    <NA>    <NA>    41.8
## 2 06048DCFC8520... electr... 6/4/20... 6/4/20... <NA>    <NA>    <NA>    <NA>    41.8
## 3 9598066F68045... electr... 6/4/20... 6/4/20... <NA>    <NA>    <NA>    <NA>    41.8
## 4 B03C0FE48C412... electr... 6/3/20... 6/3/20... <NA>    <NA>    <NA>    <NA>    41.8
## 5 B9EEA89F8FEE7... electr... 6/4/20... 6/4/20... <NA>    <NA>    <NA>    <NA>    41.8
## 6 62B943CEAAA42... electr... 6/3/20... 6/3/20... <NA>    <NA>    <NA>    <NA>    41.8
## # ... with 4 more variables: start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names 1rideable_type,
## #   2started_at, 3ended_at, 4start_station_name, 5start_station_id,
## #   6end_station_name, 7end_station_id, 8start_lat
```

```
tail(Jun21)
```

```
## # A tibble: 6 × 13
##   ride_id      ridea...1 start...2 ended...3 start...4 start...5 end_s...6 end_s...7 start...8
##   <chr>         <chr>    <chr>    <chr>    <chr>    <chr>    <chr>    <chr>    <dbl>
## 1 547E5403EE677... electr... 6/12/2... 6/12/2... Wells ... SL-011 <NA>    <NA>    41.9
## 2 CB282292CCFCE... electr... 6/14/2... 6/14/2... Wells ... SL-011 <NA>    <NA>    41.9
## 3 47BD346FAFB9B... classi... 6/30/2... 6/30/2... Clark ... 13303 Kingsb... KA1503... 41.9
## 4 52467C23D17C6... classi... 6/13/2... 6/13/2... Indian... TA1307... State ... SL-013 41.8
## 5 7DF6D74420D7D... electr... 6/8/20... 6/8/20... Clark ... 13303 <NA>    <NA>    41.9
## 6 0C01F8BA99E51... electr... 6/3/20... 6/3/20... Clark ... 13303 <NA>    <NA>    41.9
## # ... with 4 more variables: start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, and abbreviated variable names 1rideable_type,
## #   2started_at, 3ended_at, 4start_station_name, 5start_station_id,
## #   6end_station_name, 7end_station_id, 8start_lat
```

```
summary(Jun21)
```

```
##      ride_id      rideable_type      started_at      ended_at
## Length:729595      Length:729595      Length:729595      Length:729595
## Class :character      Class :character      Class :character      Class :character
## Mode :character      Mode :character      Mode :character      Mode :character
##
##
##
## start_station_name start_station_id      end_station_name      end_station_id
## Length:729595      Length:729595      Length:729595      Length:729595
## Class :character      Class :character      Class :character      Class :character
## Mode :character      Mode :character      Mode :character      Mode :character
##
##
##
##      start_lat      start_lng      end_lat      end_lng
## Min. :41.64      Min. : -87.78      Min. :41.51      Min. : -87.86
## 1st Qu.:41.88      1st Qu.: -87.66      1st Qu.:41.88      1st Qu.: -87.66
## Median :41.90      Median : -87.64      Median :41.90      Median : -87.64
## Mean :41.90      Mean : -87.64      Mean :41.90      Mean : -87.64
## 3rd Qu.:41.93      3rd Qu.: -87.63      3rd Qu.:41.93      3rd Qu.: -87.63
## Max. :42.07      Max. : -87.52      Max. :42.08      Max. : -87.49
##
##      NA's :717      NA's :717
## member_casual
## Length:729595
## Class :character
## Mode :character
##
##
##
##
```

```
str(Jun21)
```

```
## spc_tbl_ [729,595 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ ride_id      : chr [1:729595] "99FEC93BA843FB20" "06048DCFC8520CAF" "9598066F68045DF2" "B03C0FE48C4122
## 14" ...
## $ rideable_type      : chr [1:729595] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
## $ started_at      : chr [1:729595] "6/13/2021 14:31" "6/4/2021 11:18" "6/4/2021 9:49" "6/3/2021 19:56" ...
## $ ended_at      : chr [1:729595] "6/13/2021 14:34" "6/4/2021 11:24" "6/4/2021 9:55" "6/3/2021 20:21" ...
## $ start_station_name: chr [1:729595] NA NA NA NA ...
## $ start_station_id  : chr [1:729595] NA NA NA NA ...
## $ end_station_name  : chr [1:729595] NA NA NA NA ...
## $ end_station_id    : chr [1:729595] NA NA NA NA ...
## $ start_lat      : num [1:729595] 41.8 41.8 41.8 41.8 41.8 ...
## $ start_lng      : num [1:729595] -87.6 -87.6 -87.6 -87.6 -87.6 ...
## $ end_lat      : num [1:729595] 41.8 41.8 41.8 41.8 41.8 ...
## $ end_lng      : num [1:729595] -87.6 -87.6 -87.6 -87.6 -87.6 ...
## $ member_casual    : chr [1:729595] "member" "member" "member" "member" ...
## - attr(*, "spec")=
## .. cols(
## ..   ride_id = col_character(),
## ..   rideable_type = col_character(),
## ..   started_at = col_character(),
## ..   ended_at = col_character(),
## ..   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>
```

Columns `started_at` and `ended_at` need to be convert from character data type to date data type. **Str()** syntax confirms changes.

```
Jun21$started_at <- mdy_hm(Jun21$started_at)
Jun21$ended_at <- mdy_hm(Jun21$ended_at)
str(Jun21)
```

```
## spc_tbl_ [729,595 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ ride_id      : chr [1:729595] "99FEC93BA843FB20" "06048DCFC8520CAF" "9598066F68045DF2" "B03C0FE48C4122
14" ...
## $ rideable_type : chr [1:729595] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
## $ started_at   : POSIXct[1:729595], format: "2021-06-13 14:31:00" "2021-06-04 11:18:00" ...
## $ ended_at     : POSIXct[1:729595], format: "2021-06-13 14:34:00" "2021-06-04 11:24:00" ...
## $ start_station_name: chr [1:729595] NA NA NA NA ...
## $ start_station_id  : chr [1:729595] NA NA NA NA ...
## $ end_station_name  : chr [1:729595] NA NA NA NA ...
## $ end_station_id    : chr [1:729595] NA NA NA NA ...
## $ start_lat        : num [1:729595] 41.8 41.8 41.8 41.8 41.8 ...
## $ start_lng        : num [1:729595] -87.6 -87.6 -87.6 -87.6 -87.6 ...
## $ end_lat          : num [1:729595] 41.8 41.8 41.8 41.8 41.8 ...
## $ end_lng          : num [1:729595] -87.6 -87.6 -87.6 -87.6 -87.6 ...
## $ member_casual    : chr [1:729595] "member" "member" "member" "member" ...
## - attr(*, "spec")=
## .. cols(
## ..   ride_id = col_character(),
## ..   rideable_type = col_character(),
## ..   started_at = col_character(),
## ..   ended_at = col_character(),
## ..   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.

```
Jun21$date <- as.Date(Jun21$started_at)
Jun21$month <- format(as.Date(Jun21$date), "%m")
Jun21$day <- format(as.Date(Jun21$date), "%d")
Jun21$year <- format(as.Date(Jun21$date), "%Y")
Jun21$day_of_week <- format(as.Date(Jun21$date), "%A")
Jun21$ride_length <- difftime(Jun21$ended_at, Jun21$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(Jun21$ride_length)
```

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

Recheck *ride_length* data type.

```
Jun21$ride_length <- as.numeric(as.character(Jun21$ride_length))
is.numeric(Jun21$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
Jun21 <- na.omit(Jun21)
```

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.

```
Jun21 <- subset(Jun21, nchar(as.character(ride_id)) == 16)
```

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

```
Jun21 <- subset (Jun21, ride_length > "1")
```

STEP FOUR: ANALYZE DATA

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

```
mean(Jun21$ride_length)
```

```
## [1] 1591.727
```

```
median(Jun21$ride_length)
```

```
## [1] 840
```

```
max(Jun21$ride_length)
```

```
## [1] 3356640
```

```
min(Jun21$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(Jun21$ride_length)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       60     480     840    1592    1500 3356640
```

Compare the members and casual users

```
aggregate(Jun21$ride_length ~ Jun21$member_casual, FUN = mean)
```

```
##      Jun21$member_casual Jun21$ride_length
## 1                casual      2325.7656
## 2                member      856.3459
```

```
aggregate(Jun21$ride_length ~ Jun21$member_casual, FUN = median)
```

```
##      Jun21$member_casual Jun21$ride_length
## 1                casual      1080
## 2                member      660
```

```
aggregate(Jun21$ride_length ~ Jun21$member_casual, FUN = max)
```

```
##      Jun21$member_casual Jun21$ride_length
## 1                casual    3356640
## 2                member     89700
```

```
aggregate(Jun21$ride_length ~ Jun21$member_casual, FUN = min)
```

```
##      Jun21$member_casual Jun21$ride_length
## 1                casual      60
## 2                member      60
```

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

```
aggregate(Jun21$ride_length ~ Jun21$member_casual + Jun21$day_of_week, FUN = mean)
```

```
##      Jun21$member_casual Jun21$day_of_week Jun21$ride_length
## 1          casual      Friday      2277.8661
## 2          member      Friday       843.9492
## 3          casual      Monday      1893.3033
## 4          member      Monday       793.9335
## 5          casual      Saturday     2602.6440
## 6          member      Saturday     936.9463
## 7          casual      Sunday      2593.7640
## 8          member      Sunday       974.9384
## 9          casual      Thursday     2195.8357
## 10         member      Thursday      822.5388
## 11         casual      Tuesday     2086.7342
## 12         member      Tuesday      828.9955
## 13         casual      Wednesday    2184.8598
## 14         member      Wednesday     821.1432
```

Sort the days of the week in order.

```
Jun21$day_of_week <- ordered(Jun21$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(Jun21$ride_length ~ Jun21$member_casual + Jun21$day_of_week, FUN = mean)

head(x)
```

```
##      Jun21$member_casual Jun21$day_of_week Jun21$ride_length
## 1          casual      Sunday      2593.7640
## 2          member      Sunday       974.9384
## 3          casual      Monday      1893.3033
## 4          member      Monday       793.9335
## 5          casual      Tuesday     2086.7342
## 6          member      Tuesday      828.9955
```

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

```
y <- Jun21 %>%
  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        58448          2594.
## 2 casual            2        27936          1893.
## 3 casual            3        38386          2087.
## 4 casual            4        38870          2185.
## 5 casual            5        32932          2196.
## 6 casual            6        42824          2278.
```

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

```
table(Jun21$member_casual)
```

```
##
## casual member
## 302239 301687
```

```
table(Jun21$rideable_type)
```

```
##
## classic_bike  docked_bike electric_bike
##      430159      51508      122259
```

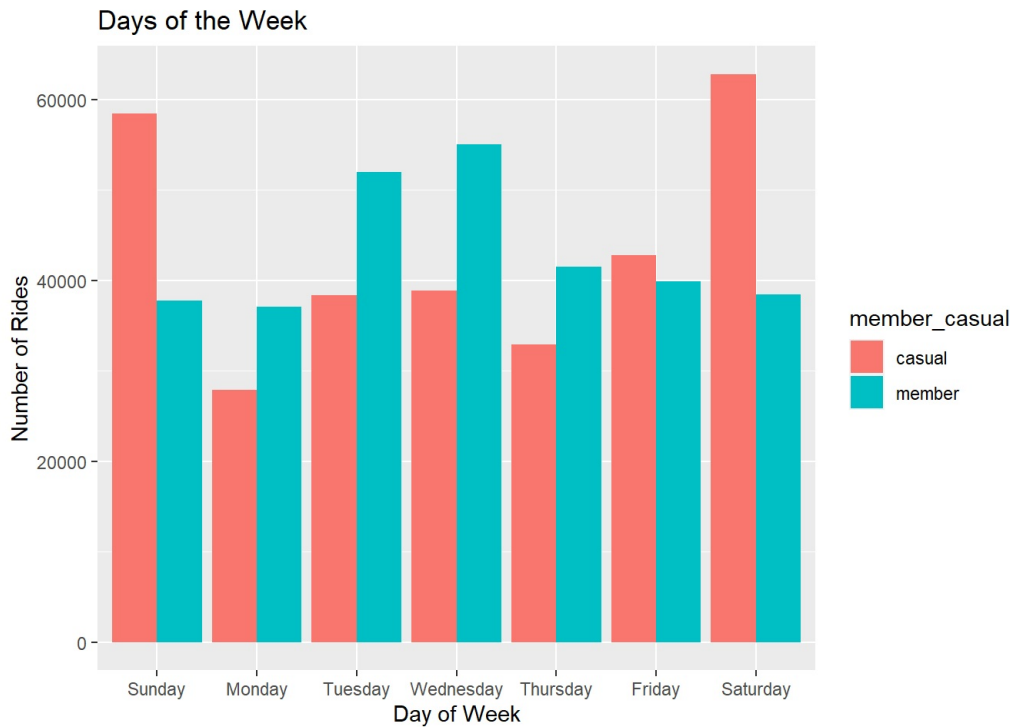
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

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

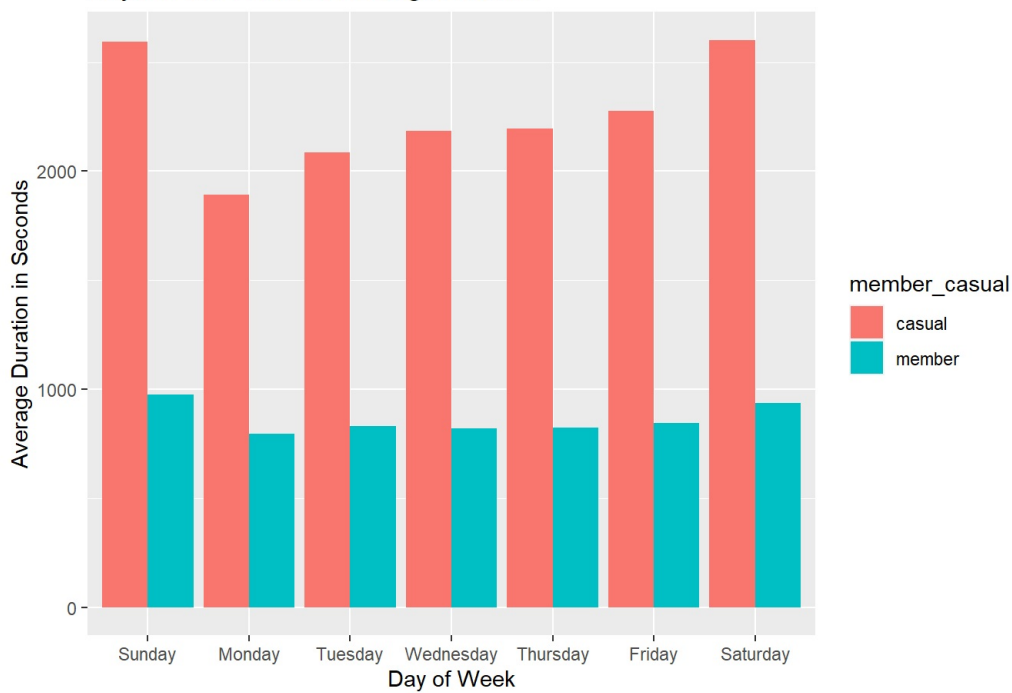
```
Jun21 %>%
  mutate(day_of_week) %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "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.

```
Jun21 %>%
  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")
```

Days of the Week vs Average Duration



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

```
mc<- as.data.frame(table(Jun21$day_of_week,Jun21$member_casual))
```

Rename columns

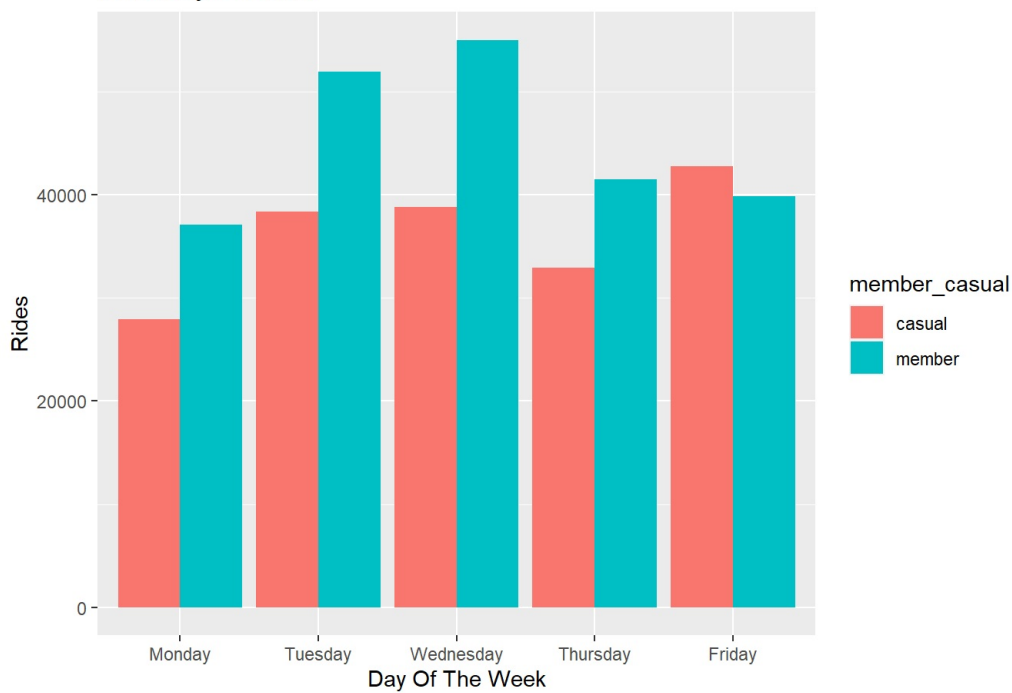
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)
head(mc)
```

```
##   day_of_week member_casual  Freq
## 1    Sunday          casual 58448
## 2    Monday          casual 27936
## 3   Tuesday          casual 38386
## 4  Wednesday          casual 38870
## 5   Thursday          casual 32932
## 6    Friday          casual 42824
```

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")
```

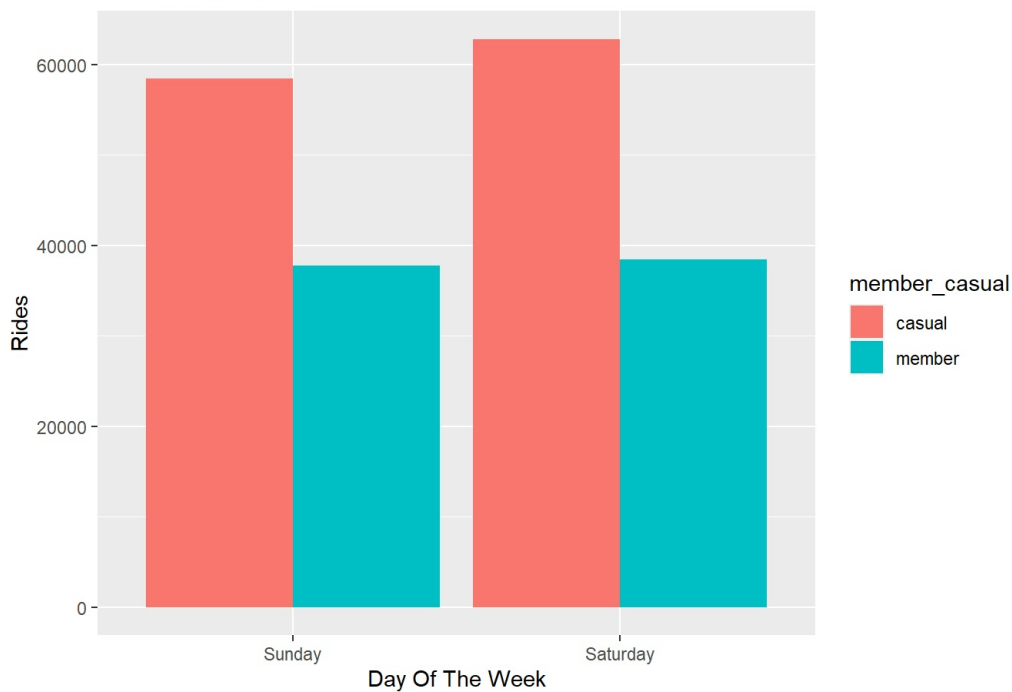

Weekdays Trends



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 = "Day Of The Week",
       y = "Rides")
```

Weekends Trends



Create dataframe for member and casual riders vs ride type

```
rt<- as.data.frame(table(Jun21$rideable_type,Jun21$member_casual))
```

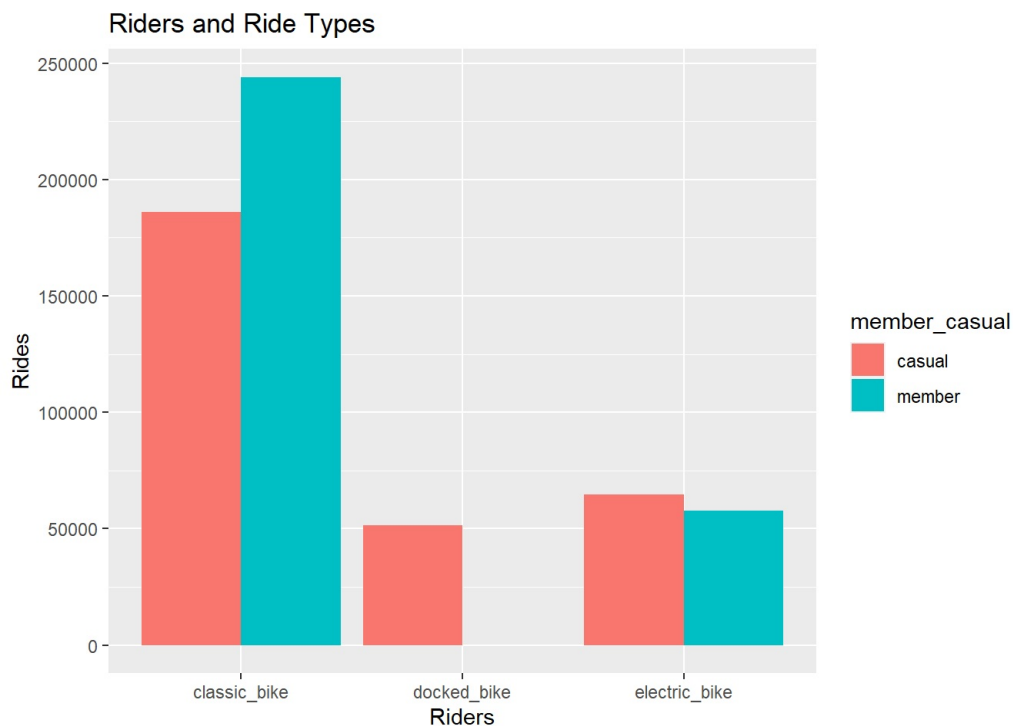
Rename columns.

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

```
##   rideable_type member_casual   Freq
## 1 classic_bike      casual 186131
## 2 docked_bike       casual  51508
## 3 electric_bike     casual  64600
## 4 classic_bike     member 244028
## 5 docked_bike      member    0
## 6 electric_bike    member  57659
```

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(Jun21, "Jun21.csv")`