

Cyclistic Case Study

Hezar K

2022-11-29

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

```
May21 <- read_csv("C:/Users/theby/Documents/202105-divvy-tripdata.csv")
```

```
## Rows: 531633 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(May21)

```
colnames(May21)
```

```
## [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(May21)
```

```
## [1] 531633
```

```
dim(May21)
```

```
## [1] 531633      13
```

```
head(May21)
```

```
## # 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 C809ED75D6160... electr... 5/30/2... 5/30/2... <NA>    <NA>    <NA>    <NA>    41.9
## 2 DD59FDCE0ACAC... electr... 5/30/2... 5/30/2... <NA>    <NA>    <NA>    <NA>    41.9
## 3 0AB83CB88C43E... electr... 5/30/2... 5/30/2... <NA>    <NA>    <NA>    <NA>    41.9
## 4 7881AC6D39110... electr... 5/30/2... 5/30/2... <NA>    <NA>    <NA>    <NA>    41.9
## 5 853FA701B4582... electr... 5/30/2... 5/30/2... <NA>    <NA>    <NA>    <NA>    41.9
## 6 F5E63DFD96B2A... electr... 5/30/2... 5/30/2... <NA>    <NA>    <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
```

```
tail(May21)
```

```
## # 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 D0B8E59E2B3C4... electr... 5/2/20... 5/2/20... Blacks... 13398    <NA>    <NA>    41.8
## 2 EF56D7D1D612A... electr... 5/20/2... 5/20/2... Blacks... 13398    <NA>    <NA>    41.8
## 3 745191CB9F21D... classi... 5/29/2... 5/29/2... Sherid... TA1307... Michig... 13042    42.0
## 4 428575BAA5356... electr... 5/31/2... 5/31/2... Sherid... TA1307... <NA>    <NA>    42.0
## 5 FC8A4A7AB7249... electr... 5/25/2... 5/25/2... Sherid... TA1307... <NA>    <NA>    42.0
## 6 E873B8AA3EE84... docked... 5/12/2... 5/12/2... Sherid... TA1307... Clark ... TA1307... 42.0
## # ... 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(May21)
```

```
##      ride_id      rideable_type      started_at      ended_at
## Length:531633      Length:531633      Length:531633      Length:531633
## 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:531633      Length:531633      Length:531633      Length:531633
## 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. :41.65      Min. : -87.78      Min. :41.56      Min. : -87.85
## 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. :42.09      Max. : -87.52
##
##                NA's :452      NA's :452
## member_casual
## Length:531633
## Class :character
## Mode :character
##
##
##
##
```

```
str(May21)
```

```
## spc_tbl_ [531,633 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ ride_id      : chr [1:531633] "C809ED75D6160B2A" "DD59FDCE0ACACAF3" "0AB83CB88C43EFC2" "7881AC6D39110C
## 60" ...
## $ rideable_type : chr [1:531633] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
## $ started_at    : chr [1:531633] "5/30/2021 11:58" "5/30/2021 11:29" "5/30/2021 14:24" "5/30/2021 14:25"
## ...
## $ ended_at      : chr [1:531633] "5/30/2021 12:10" "5/30/2021 12:14" "5/30/2021 14:25" "5/30/2021 14:41"
## ...
## $ start_station_name: chr [1:531633] NA NA NA NA ...
## $ start_station_id  : chr [1:531633] NA NA NA NA ...
## $ end_station_name  : chr [1:531633] NA NA NA NA ...
## $ end_station_id    : chr [1:531633] NA NA NA NA ...
## $ start_lat         : num [1:531633] 41.9 41.9 41.9 41.9 41.9 ...
## $ start_lng         : num [1:531633] -87.6 -87.6 -87.7 -87.7 -87.7 ...
## $ end_lat          : num [1:531633] 41.9 41.8 41.9 41.9 41.9 ...
## $ end_lng          : num [1:531633] -87.6 -87.6 -87.7 -87.7 -87.7 ...
## $ member_casual     : chr [1:531633] "casual" "casual" "casual" "casual" ...
## - 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.

```
May21$started_at <- mdy_hm(May21$started_at)
May21$ended_at <- mdy_hm(May21$ended_at)
str(May21)
```

```
## spc_tbl_ [531,633 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ ride_id      : chr [1:531633] "C809ED75D6160B2A" "DD59FDCE0ACACAF3" "0AB83CB88C43EFC2" "7881AC6D39110C60" ...
## $ rideable_type : chr [1:531633] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
## $ started_at   : POSIXct[1:531633], format: "2021-05-30 11:58:00" "2021-05-30 11:29:00" ...
## $ ended_at     : POSIXct[1:531633], format: "2021-05-30 12:10:00" "2021-05-30 12:14:00" ...
## $ start_station_name: chr [1:531633] NA NA NA NA ...
## $ start_station_id : chr [1:531633] NA NA NA NA ...
## $ end_station_name : chr [1:531633] NA NA NA NA ...
## $ end_station_id   : chr [1:531633] NA NA NA NA ...
## $ start_lat        : num [1:531633] 41.9 41.9 41.9 41.9 41.9 ...
## $ start_lng        : num [1:531633] -87.6 -87.6 -87.7 -87.7 -87.7 ...
## $ end_lat          : num [1:531633] 41.9 41.8 41.9 41.9 41.9 ...
## $ end_lng          : num [1:531633] -87.6 -87.6 -87.7 -87.7 -87.7 ...
## $ member_casual    : chr [1:531633] "casual" "casual" "casual" "casual" ...
## - 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.

```
May21$date <- as.Date(May21$started_at)
May21$month <- format(as.Date(May21$date), "%m")
May21$day <- format(as.Date(May21$date), "%d")
May21$year <- format(as.Date(May21$date), "%Y")
May21$day_of_week <- format(as.Date(May21$date), "%A")
May21$ride_length <- difftime(May21$ended_at, May21$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(May21$ride_length)
```

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

Recheck *ride_length* data type.

```
May21$ride_length <- as.numeric(as.character(May21$ride_length))
is.numeric(May21$ride_length)
```

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

STEP THREE: CLEAN DATA

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

```
May21 <- na.omit(May21)
```

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.

```
May21 <- subset(May21, nchar(as.character(ride_id)) == 16)
```

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

```
May21 <- subset (May21, ride_length > "1")
```

STEP FOUR: ANALYZE DATA

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

```
mean(May21$ride_length)
```

```
## [1] 1602.389
```

```
median(May21$ride_length)
```

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

```
max(May21$ride_length)
```

```
## [1] 3235260
```

```
min(May21$ride_length)
```

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

Run a statistical summary of the *ride_length*.

```
summary(May21$ride_length)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       60     480     840    1602    1560 3235260
```

Compare the members and casual users

```
aggregate(May21$ride_length ~ May21$member_casual, FUN = mean)
```

```
##      May21$member_casual May21$ride_length
## 1                casual      2390.6105
## 2                member       869.2645
```

```
aggregate(May21$ride_length ~ May21$member_casual, FUN = median)
```

```
##      May21$member_casual May21$ride_length
## 1                casual          1200
## 2                member           660
```

```
aggregate(May21$ride_length ~ May21$member_casual, FUN = max)
```

```
##      May21$member_casual May21$ride_length
## 1                casual      3235260
## 2                member       88020
```

```
aggregate(May21$ride_length ~ May21$member_casual, FUN = min)
```

```
##      May21$member_casual May21$ride_length
## 1                casual           60
## 2                member           60
```

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

```
aggregate(May21$ride_length ~ May21$member_casual + May21$day_of_week, FUN = mean)
```

```
##      May21$member_casual May21$day_of_week May21$ride_length
## 1          casual      Friday      2206.7040
## 2          member      Friday       820.7301
## 3          casual      Monday      2347.8264
## 4          member      Monday       841.3280
## 5          casual      Saturday     2411.8294
## 6          member      Saturday       974.1453
## 7          casual      Sunday      2896.2811
## 8          member      Sunday     1015.8313
## 9          casual      Thursday     2022.9587
## 10         member      Thursday       801.6089
## 11         casual      Tuesday     1909.6763
## 12         member      Tuesday       764.5276
## 13         casual      Wednesday    1932.2563
## 14         member      Wednesday     813.2399
```

Sort the days of the week in order.

```
May21$day_of_week <- ordered(May21$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(May21$ride_length ~ May21$member_casual + May21$day_of_week, FUN = mean)

head(x)
```

```
##      May21$member_casual May21$day_of_week May21$ride_length
## 1          casual      Sunday      2896.2811
## 2          member      Sunday     1015.8313
## 3          casual      Monday      2347.8264
## 4          member      Monday       841.3280
## 5          casual      Tuesday     1909.6763
## 6          member      Tuesday       764.5276
```

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

```
y <- May21 %>%
  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        53812          2896.
## 2 casual          2        28810          2348.
## 3 casual          3        14919          1910.
## 4 casual          4        18823          1932.
## 5 casual          5        18488          2023.
## 6 casual          6        24824          2207.
```

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

```
table(May21$member_casual)
```

```
##
## casual member
## 215596 231799
```

```
table(May21$rideable_type)
```

```
##
## classic_bike  docked_bike electric_bike
##      305709      43155      98531
```

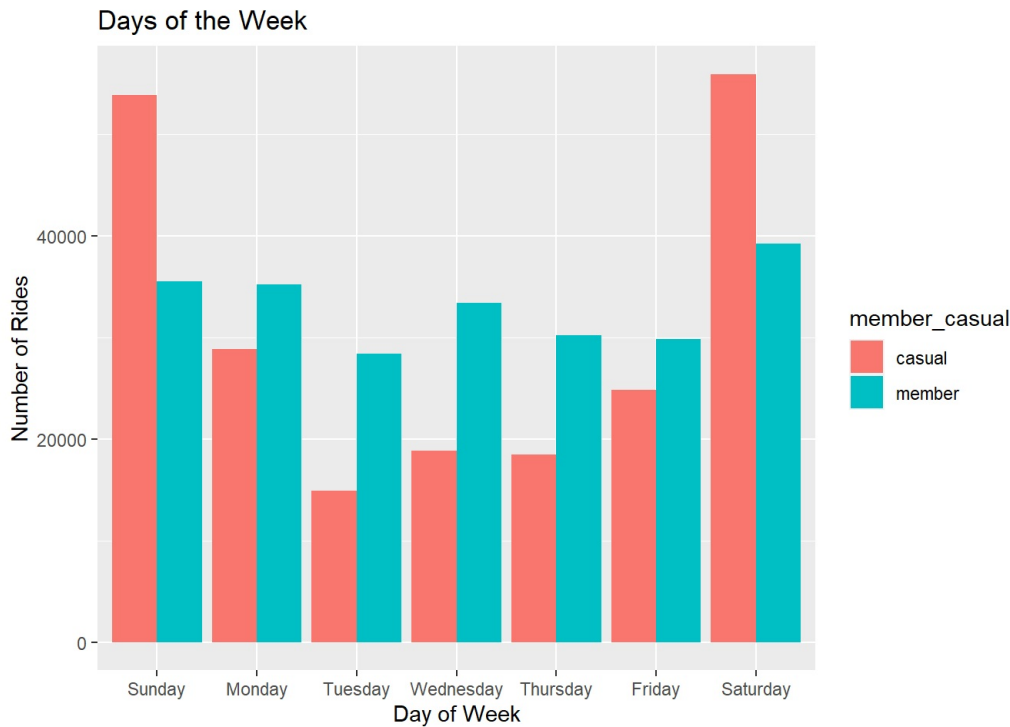
STEP FIVE: VISUALIZATION

Display full digits instead of scientific number.

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

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

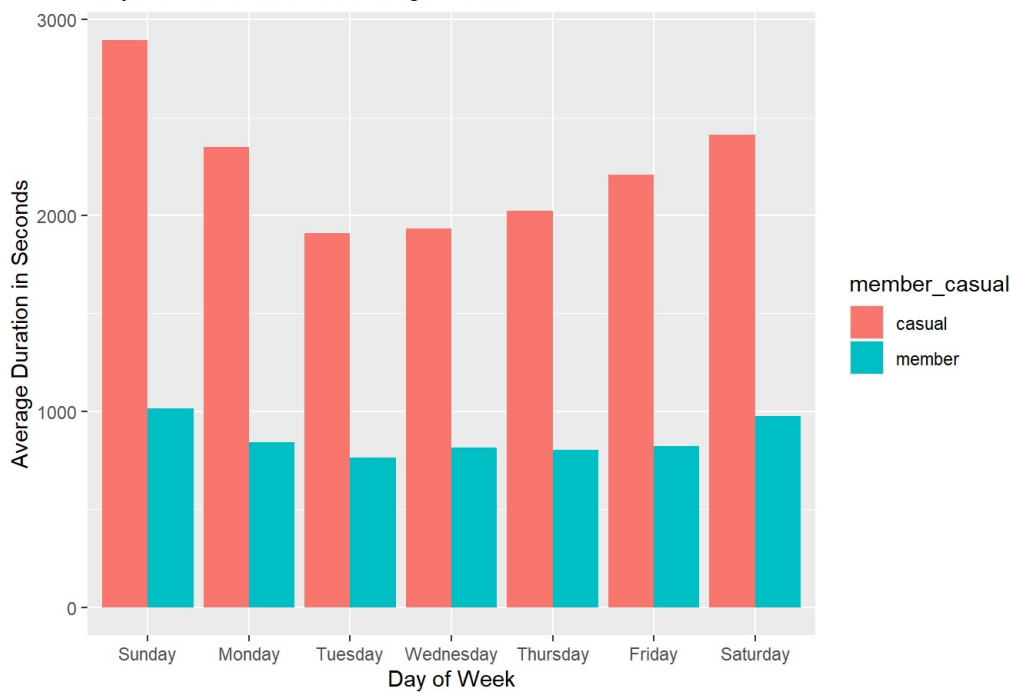
```
May21 %>%
  mutate(day_of_week) %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length), .groups = 'drop') %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "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.

```
May21 %>%
  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(May21$day_of_week,May21$member_casual))
```

Rename columns

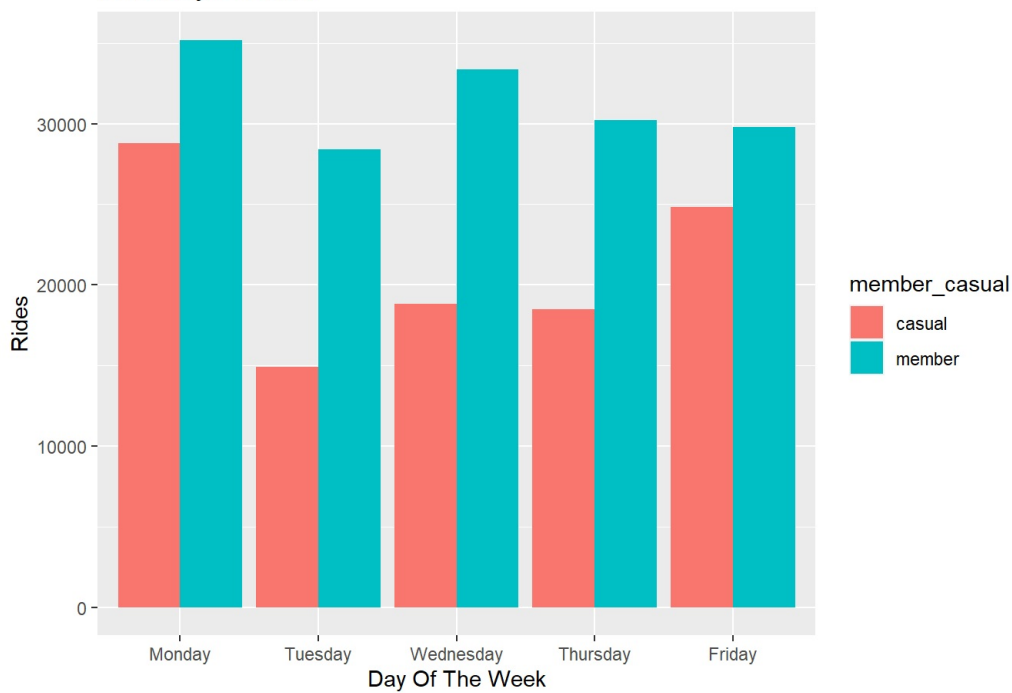
```
mc<-rename(mc, day_of_week = Var1, member_casual = Var2)
head(mc)
```

```
##   day_of_week member_casual  Freq
## 1    Sunday          casual 53812
## 2    Monday          casual 28810
## 3   Tuesday          casual 14919
## 4  Wednesday          casual 18823
## 5   Thursday          casual 18488
## 6    Friday          casual 24824
```

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")
```

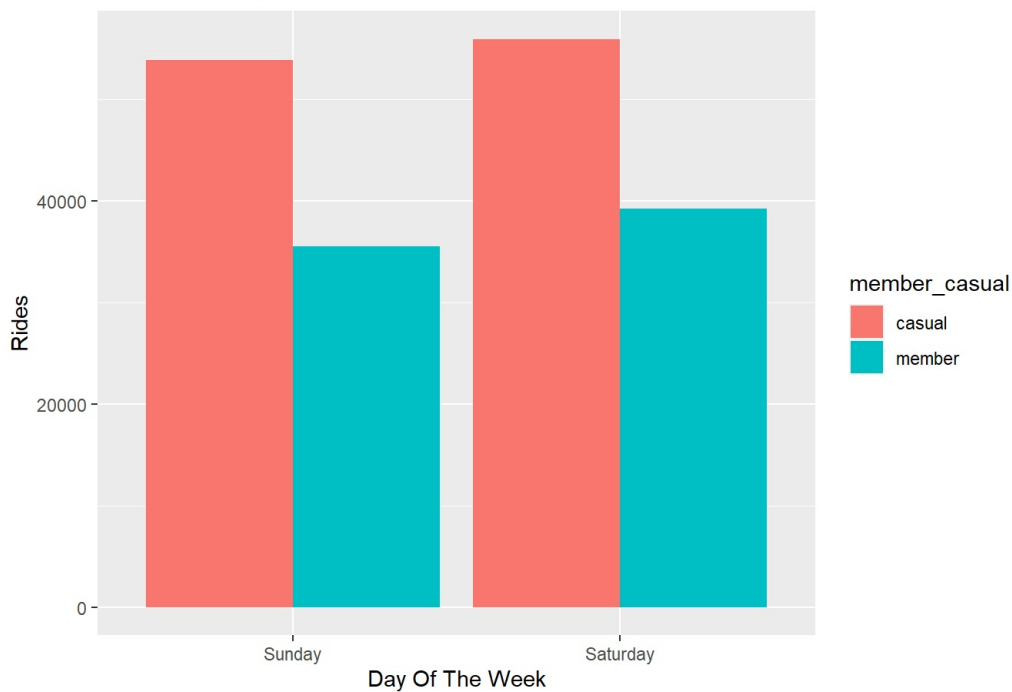

Weekdays Trends



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(May21$rideable_type,May21$member_casual))
```

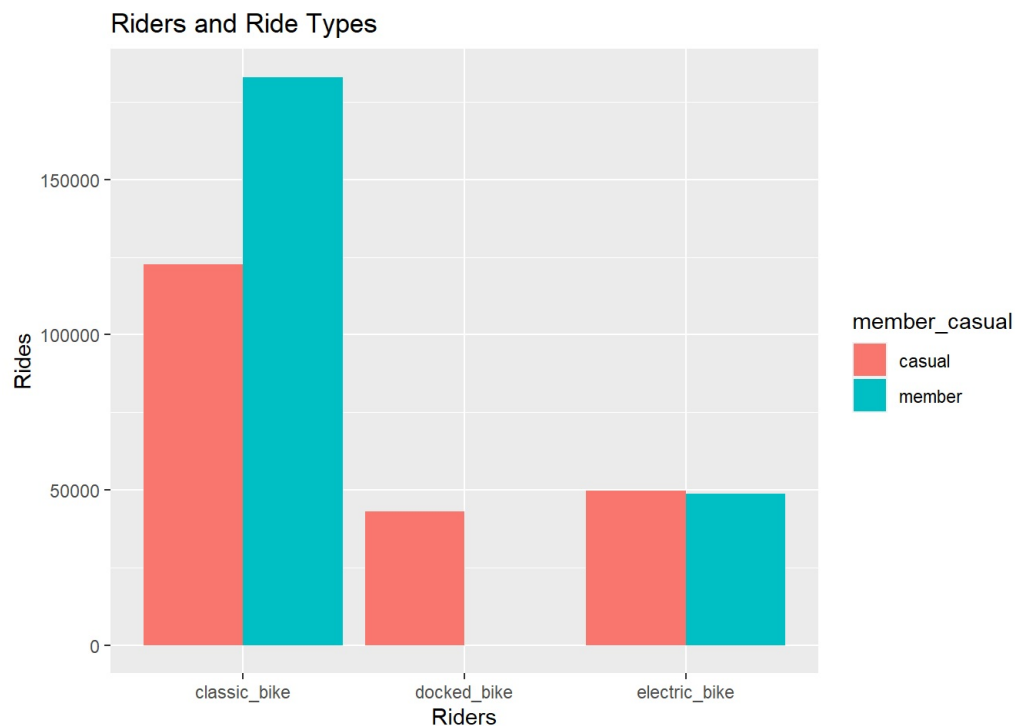
Rename columns.

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

```
##   rideable_type member_casual   Freq
## 1 classic_bike      casual 122706
## 2 docked_bike      casual  43155
## 3 electric_bike    casual  49735
## 4 classic_bike      member 183003
## 5 docked_bike      member    0
## 6 electric_bike    member  48796
```

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(May21, "May21.csv")`