Google data Analytics Capstone Project
Part 2 (b): *2019 Q1 and 2020 Q1(Using R)*

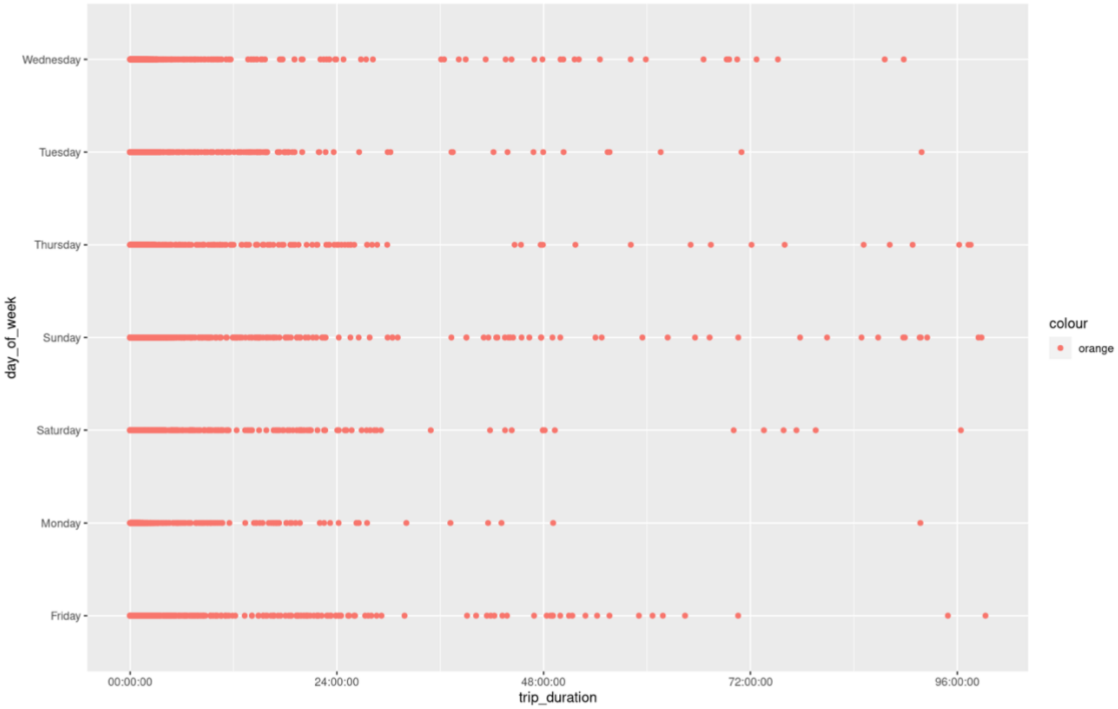
For this analyzation, R Posit Cloud was used for which data limits are quite less. Therefore, two data sets were used: Divvy_Trips_2019_Q1 and Divvy_Trips_2020_Q1. Using R Script, comparison plots of trip duration (length of a single trip) and day of week has been made for both 2019 and 2020.

Loading and Cleaning Divvy_Trips_2019_Q1

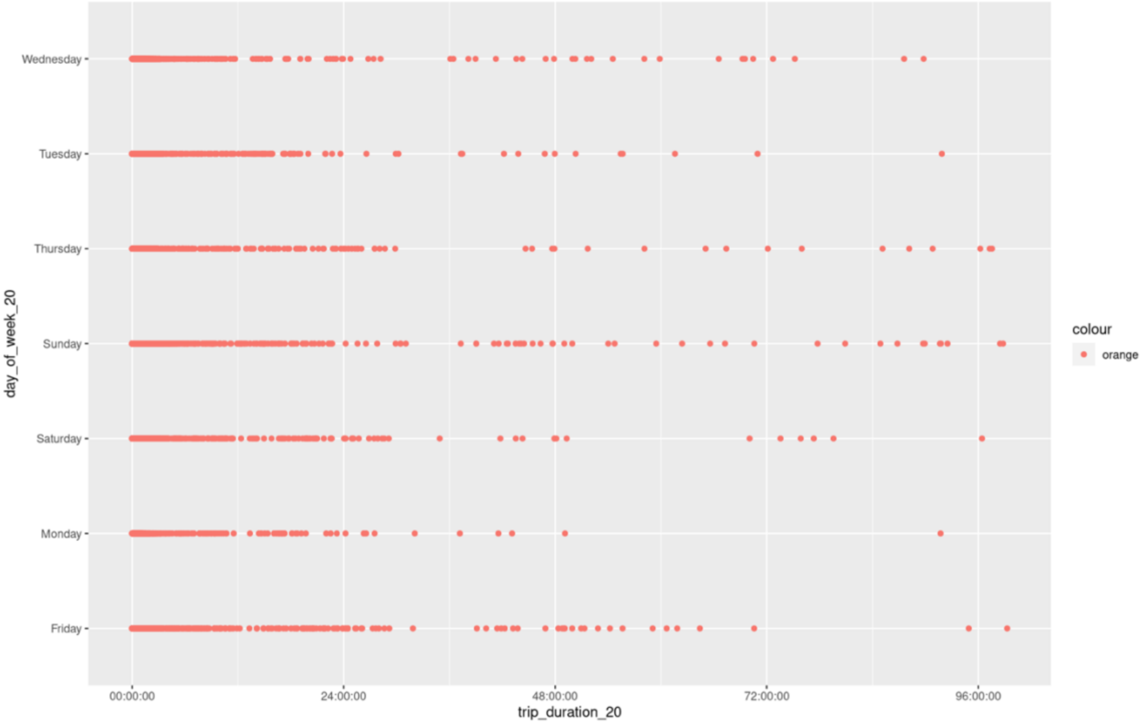
```
# Install tidyverse
install.packages("tidyverse")
library(tidyverse)

#Read the data frame Divvy_Trips_2019_Q1
  library(readr)
  Divvy_Trips_2019_Q1_Divvy_Trips_2019_Q1 <- read_csv("Divvy_Trips_2019_Q1 -
Divvy_Trips_2019_Q1.csv")

#Install tidyr, reader and dplyr packages
  install.packages("tidyr")
  library(tidyr)
  install.packages("readr")
  library(readr)
  install.packages("dplyr")
  library(dplyr)

# Create a table with unique station names and their frequencies in from_station_name
  freq_from <- Divvy_Trips_2019_Q1_Divvy_Trips_2019_Q1 % %
+     group_by(from_station_name) % %
+     summarise(freq_from = n())


# Create a table with unique station names and their frequencies in to_station_name
  freq_to <- Divvy_Trips_2019_Q1_Divvy_Trips_2019_Q1 % %
+     group_by(to_station_name) % %
+     summarise(freq_to = n())


# Merge the two tables based on station names
  station_frequency <- merge(freq_from, freq_to, by.x = "from_station_name", by.y =
"to_station_name", all = TRUE)
  View(station_frequency)
```

```
# Calculate total number of Subscribers and Customers
  total_users <- Divvy_Trips_2019_Q1_Divvy_Trips_2019_Q1 % %
+     group_by(usertype) % %
+     summarise(total_users = n())


# Calculate total number of Males and Females
  total_gender <- Divvy_Trips_2019_Q1_Divvy_Trips_2019_Q1 % %
+     group_by(gender) % %
+     summarise(total_gender = n())

  # Calculate total number of Male/Female Subscribers and Male/Female Customers
  total_users_gender <- Divvy_Trips_2019_Q1_Divvy_Trips_2019_Q1 % %
+     group_by(usertype, gender) % %
+     summarise(total_users_gender = n())
`summarise()` has grouped output by 'usertype'. You can override using the `.groups` argument.

# Display the results
 print("Total number of Subscribers and Customers:")
  print("\nTotal number of Males and Females:")
  print("\nTotal number of Male/Female Subscribers and Male/Female Customers:")


#Install ggplot2 package
  install.packages("ggplot2")
  library (ggpplot2)
data_2019$start_time <- as.POSIXct(data_2019$start_time, format = "%Y-%m-%d
%H:%M:%S")
data_2019$end_time <- as.POSIXct(data_2019$end_time, format = "%Y-%m-%d
%H:%M:%S")

# Calculate the time difference
time_diff <- as.numeric(difftime(data_2019$end_time, data_2019$start_time, units = "secs"))

# Create trip_duration column in hh:mm:ss format
data_2019$trip_duration <- sprintf("%02d:%02d:%02d",
                             time_diff %/% 3600,
                             (time_diff %% 3600) %/% 60,
                             time_diff %% 60)


data_2019$start_time <- as.POSIXct(data_2019$start_time, format = "%Y-%m-%d
%H:%M:%S")

# Create a new column for the day of the week
```

```
data_2019$day_of_week <- weekdays(data_2019$start_time)


view(data_2020)
data_2020$started_at <- as.POSIXct(data_2020$started_at, format = "%Y-%m-%d %H:%M:%S
")
  data_2020$ended_at <- as.POSIXct(data_2020$ended_at, format = "%Y-%m-%d %H:%M:%S
")

  # Calculate the time difference
  time_diff <- as.numeric(difftime(data_2020$ended_at, data_2020$started_at, units = "secs"))

  # Create trip_duration column in hh:mm:ss format
  data_2020$trip_duration_20 <- sprintf("%02d:%02d:%02d",
+                                        time_diff %/% 3600,
+                                        (time_diff %% 3600) %/% 60,
+                                        time_diff %% 60)


  data_2020$started_at <- as.POSIXct(data_2020$started_at, format = "%Y-%m-%d %H:%M:%
S")

  # Create a new column for the day of the week
  data_2020$day_of_week_20 <- weekdays(data_2020$started_at)

# Create a new data frame data_2 with 426887 rows
data_2 <- data.frame(
+     trip_duration = rep(NA, 426887),
+     day_of_week = rep(NA, 426887),
+     trip_duration_20 = rep(NA, 426887),
+     day_of_week_20 = rep(NA, 426887)
+ )

  View(data_2)
  data_2$trip_duration_20 <- data_2020$trip_duration_20
  data_2$trip_duration <- data_2019$trip_duration
  data_2$day_of_week_20 <- data_2020$day_of_week_20
  data_2$day_of_week <- data_2019$day_of_week

  # Save the new data frame data_2 to a CSV file
  write.csv(data_2, "data_2.csv", row.names = FALSE)


#Create ggplot for 2019
ggplot(data = data_2) + geom_point(mapping = aes(x= trip_duration, y= day_of_week, color= "o
range"))
```



#Create ggplot for 2020
ggplot(data = data_2) + geom_point(mapping = aes(x= trip_duration_20, y= day_of_week_20, color= "orange"))


Wednesday
Tuesday
Thursday
Sunday
Saturday
Monday
Friday
day_of_week_20
00:00:00
24:00:00
48:00:00
72:00:00
96:00:00
trip_duration_20
colour
orange