**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= "orange"))

A graph with red dots

Description automatically generated

#Create ggplot for 2020

ggplot(data = data\_2) + geom\_point(mapping = aes(x= trip\_duration\_20, y= day\_of\_week\_20, color= "orange"))

A graph with red dots

Description automatically generated