# Data Collection and Manipulation

Samuel Sim, Lee Min Shuen

November 2021

This is the python code for getting and collecting data from LTA API. The guide can be found in: https://datamall.lta.gov.sg/content/dam/datamall/datasets/LTA_DataMall_API_User_Guide.pdf.

```python
# Getting Started
import requests

# Load unique access key
Key = "insert api key"

# Returns number of trips by weekdays and weekends for individual train stations
# Update Freq By 15th of every month, only up to last three months
data = requests.get(
    "http://datamall2.mytransport.sg/ltaodataservice/PV/Train",
    params = {"Date": 202109},
    headers = {"AccountKey": Key}).json()

data2 = requests.get(
    "http://datamall2.mytransport.sg/ltaodataservice/PCDForecast",
    params = {"TrainLine": "EWL"},
    headers = {"AccountKey": Key}).json()

print(data2)
data2
```

R code for data wrangling in order to estimate certain input parameters for our model.

```r
library(tidyverse)
library("ggpubr")

peak <- c(5, 7, 8, 17, 18, 19)
morningpeak <- c(5, 7, 8)
eveningpeak <- c(17, 18, 19)

nonpeak <- seq(5, 23)
nonpeak <- subset(x = nonpeak, !nonpeak %in% peak)
nonpeak <- c(nonpeak, 0)

data1 <- read.csv("transport_node_train_202107.csv")
data2 <- read.csv("transport_node_train_202108.csv")
data3 <- read.csv("transport_node_train_202109.csv")
```

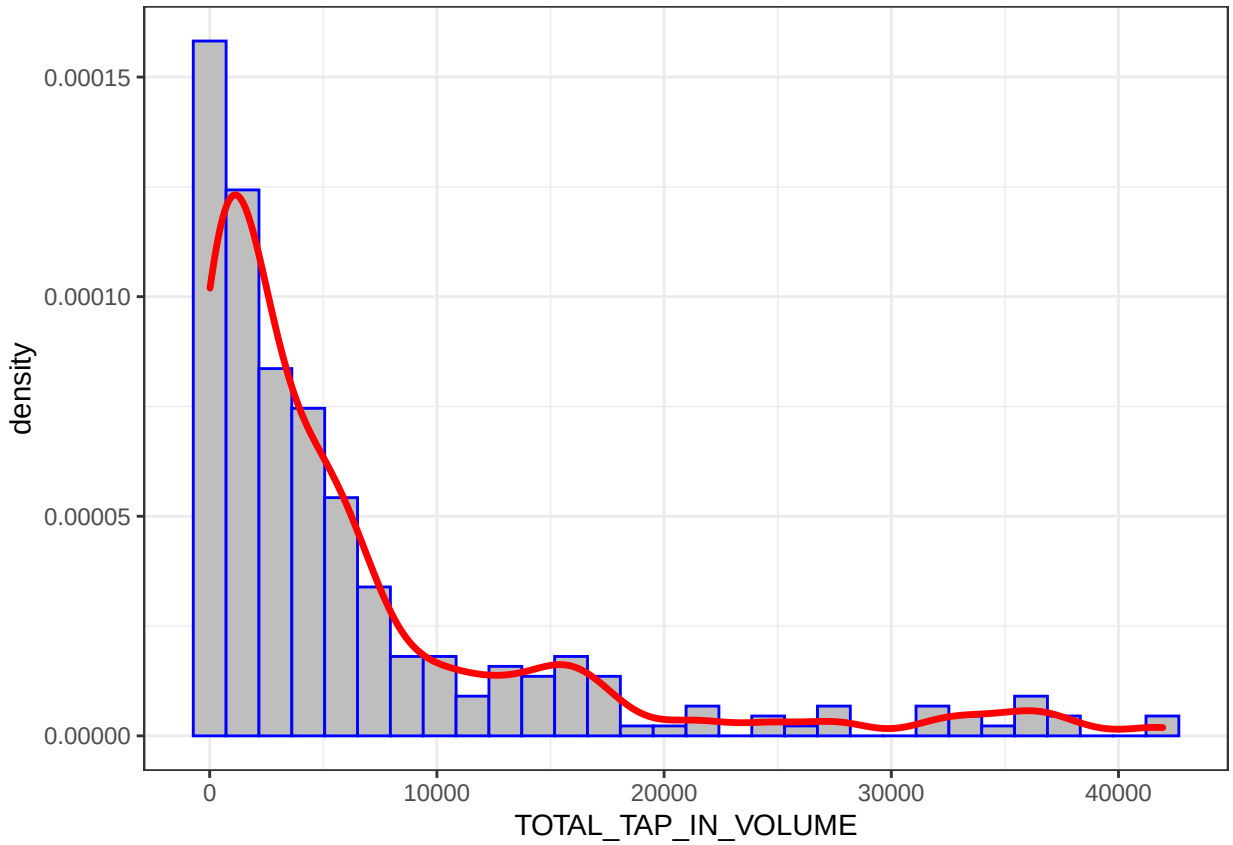
Data preprocessing stage:

```
combineddata <- rbind(data1, data2, data3)
combineddata$YEAR_MONTH <- as.factor(combineddata$YEAR_MONTH)

combineddata$PEAK <- 0
combinedpeakid <- combineddata$TIME_PER_HOUR %in% peak
combineddata$PEAK[combinedpeakid] <- ifelse(combineddata[combinedpeakid,
    ]$TIME_PER_HOUR %in% morningpeak, 1, 2)
combineddata$PEAK <- as.factor(combineddata$PEAK)

combineddata$Connected <- str_count(combineddata$PT_CODE, "/") +
    1
combineddata$TOTAL_TAP_IN_VOLUME <- combineddata$TOTAL_TAP_IN_VOLUME/combineddata$Connected
combineddata$IS_DT <- as.numeric(grepl("DT", combineddata$PT_CODE))
```

Visualization Plots of split datasets:

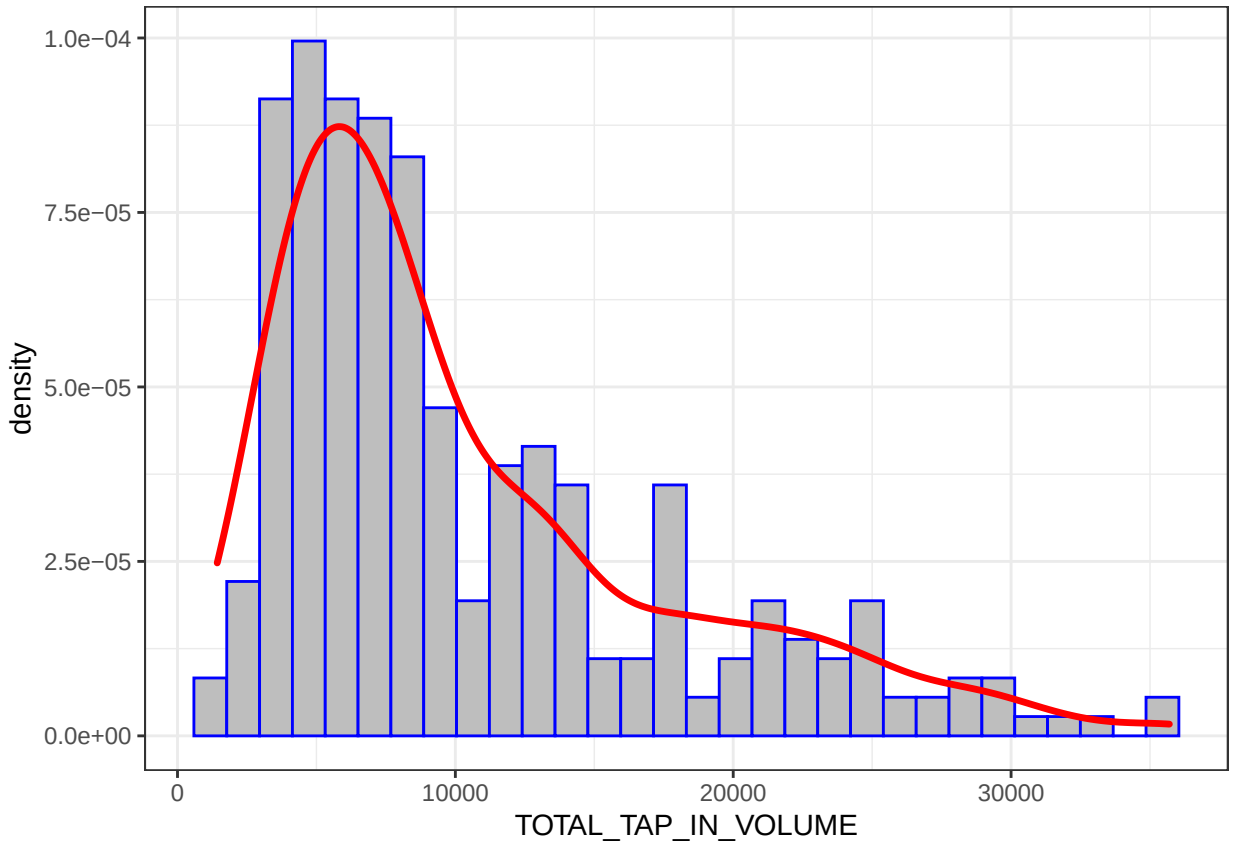
```
## Exploratory plots ##
combineddata_set1 <- combineddata[combineddata$DAY_TYPE == "WEEKDAY" &
    (combineddata$PEAK == 1 | combineddata$PEAK == 2) & combineddata$IS_DT ==
    1, ]
combineddata_set2 <- combineddata[combineddata$DAY_TYPE != "WEEKDAY" &
    (combineddata$PEAK == 1 | combineddata$PEAK == 2) & combineddata$IS_DT ==
    1, ]
combineddata_set3 <- combineddata[combineddata$DAY_TYPE == "WEEKDAY" &
    combineddata$PEAK == 0 & combineddata$IS_DT == 1, ]
combineddata_set4 <- combineddata[combineddata$DAY_TYPE != "WEEKDAY" &
    combineddata$PEAK == 0 & combineddata$IS_DT == 1, ]

ggplot(combineddata_set1[combineddata_set1$PEAK == 1, ], aes(x = TOTAL_TAP_IN_VOLUME)) +
    geom_histogram(aes(y = ..density..), color = "blue", fill = "gray") +
    geom_density(color = "red", lwd = 1.2) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



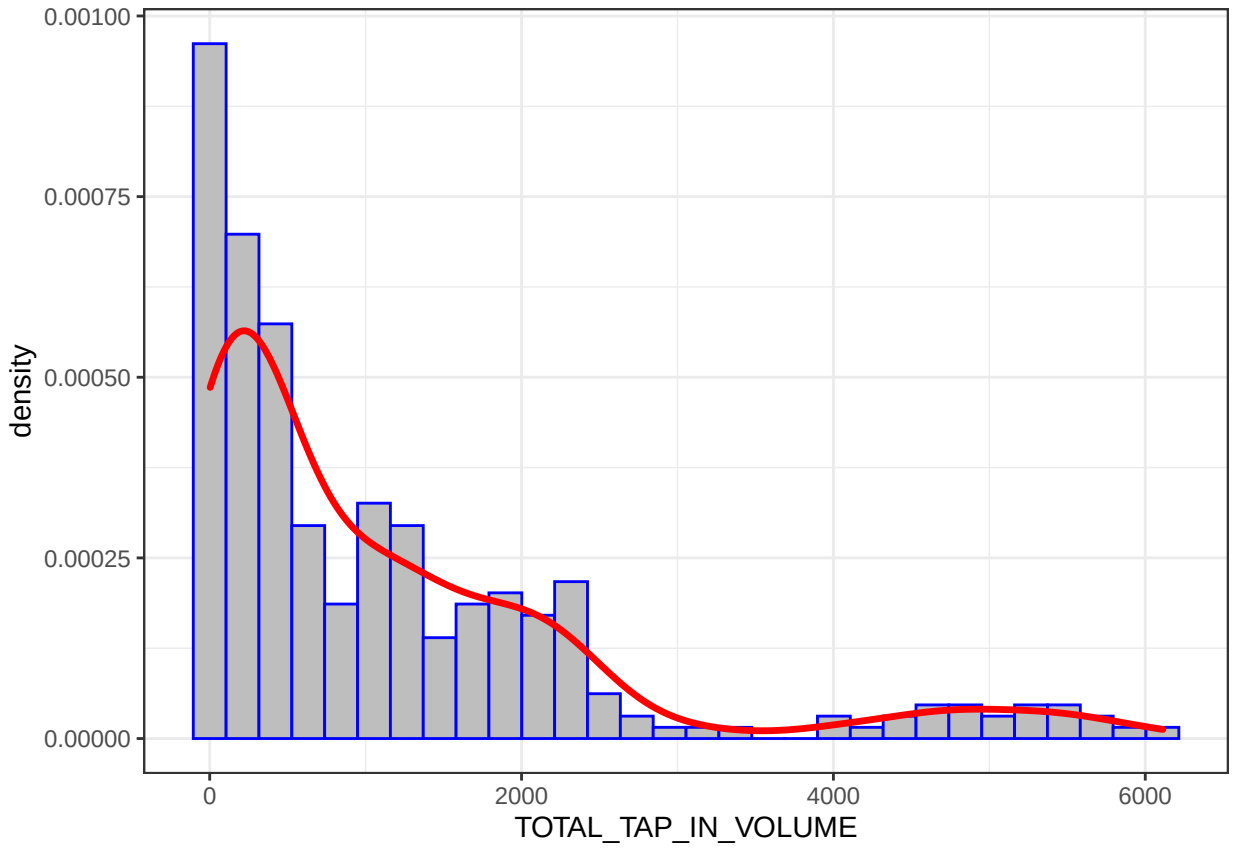

```
ggplot(combineddata_set1[combineddata_set1$PEAK == 2, ], aes(x = TOTAL_TAP_IN_VOLUME)) +
    geom_histogram(aes(y = ..density..), color = "blue", fill = "gray") +
    geom_density(color = "red", lwd = 1.2) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



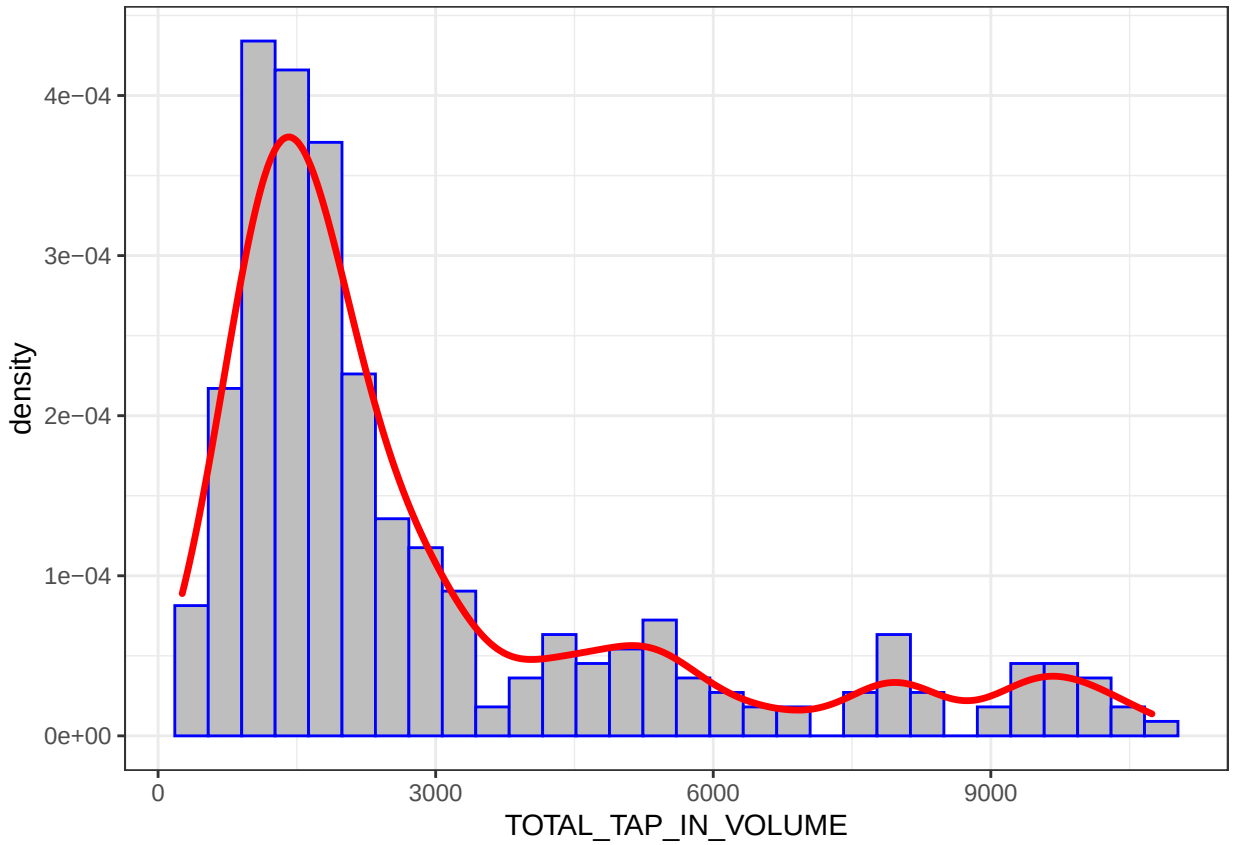

```
ggplot(combineddata_set2[combineddata_set2$PEAK == 1, ], aes(x = TOTAL_TAP_IN_VOLUME)) +
    geom_histogram(aes(y = ..density..), color = "blue", fill = "gray") +
    geom_density(color = "red", lwd = 1.2) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



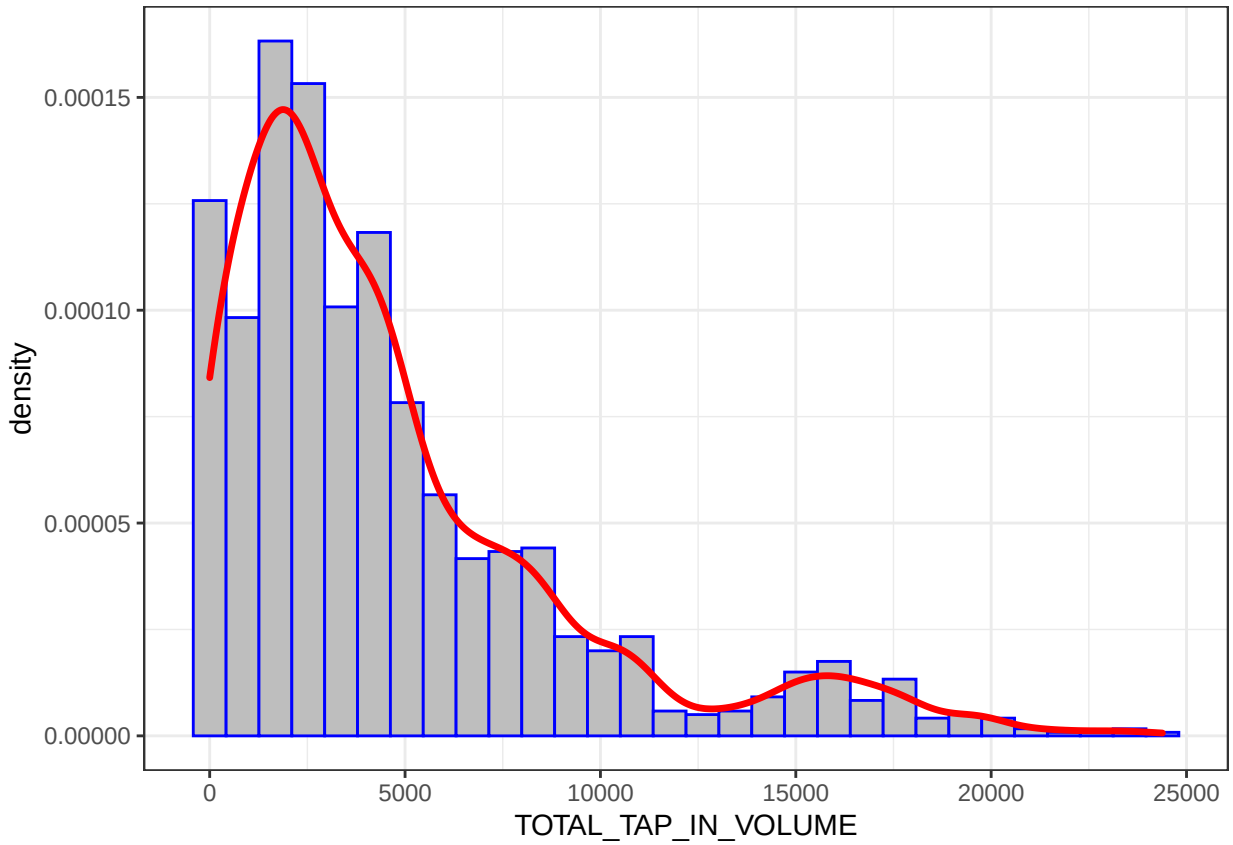

```
ggplot(combineddata_set2[combineddata_set2$PEAK == 2, ], aes(x = TOTAL_TAP_IN_VOLUME)) +
    geom_histogram(aes(y = ..density..), color = "blue", fill = "gray") +
    geom_density(color = "red", lwd = 1.2) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



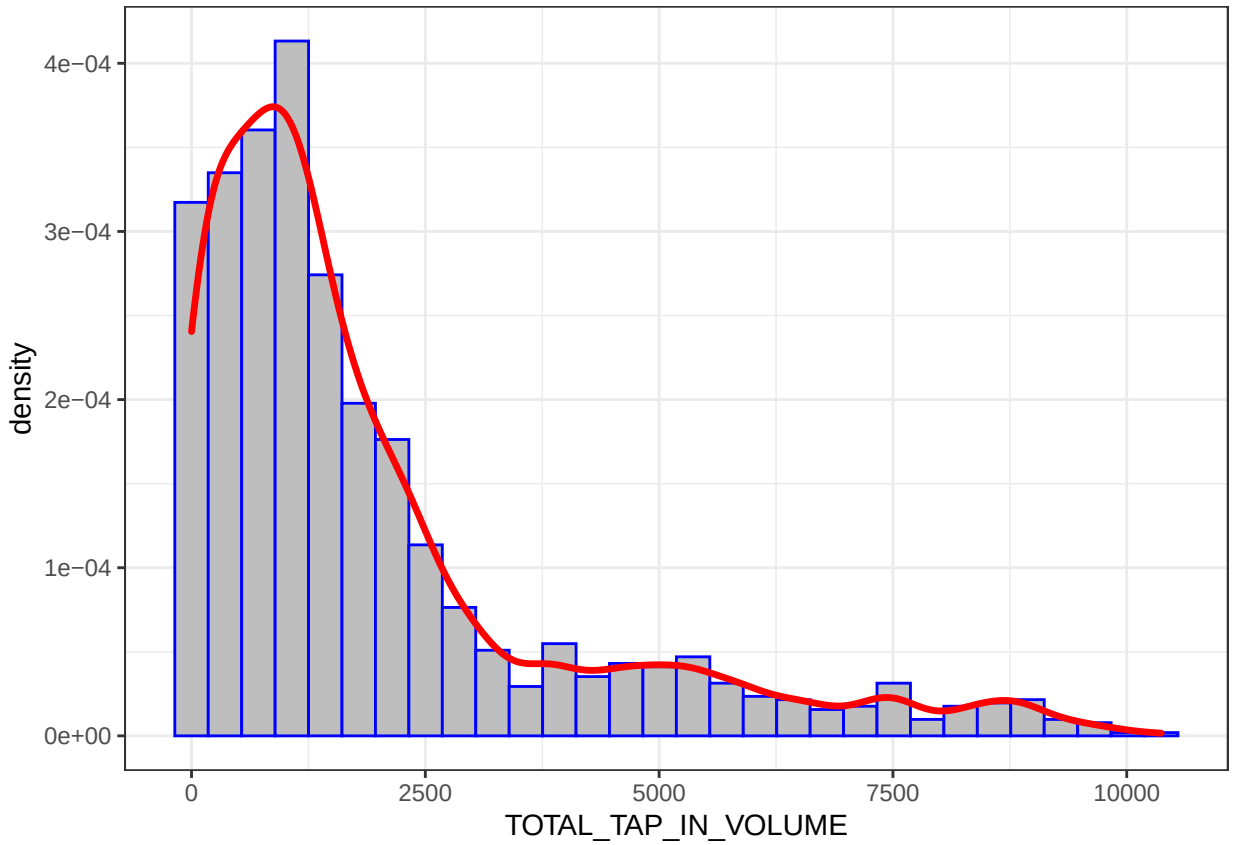

```
ggplot(combineddata_set3, aes(x = TOTAL_TAP_IN_VOLUME)) + geom_histogram(aes(y = ..density..),
    color = "blue", fill = "gray") + geom_density(color = "red",
    lwd = 1.2) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(combineddata_set4, aes(x = TOTAL_TAP_IN_VOLUME)) + geom_histogram(aes(y = ..density..),
    color = "blue", fill = "gray") + geom_density(color = "red",
    lwd = 1.2) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




Statistical Tests:

```
# Check if amount of passenger arrivals are different
# between morning peak, evening peak, and non-peak

# Weekday
kruskal.test(TOTAL_TAP_IN_VOLUME ~ PEAK, data = combineddata[combineddata$DAY_TYPE ==
    "WEEKDAY" & combineddata$YEAR_MONTH == "2021-07", ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by PEAK
## Kruskal-Wallis chi-squared = 182.74, df = 2, p-value < 2.2e-16
```

```
kruskal.test(TOTAL_TAP_IN_VOLUME ~ PEAK, data = combineddata[combineddata$DAY_TYPE ==
    "WEEKDAY" & combineddata$YEAR_MONTH == "2021-08", ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by PEAK
## Kruskal-Wallis chi-squared = 177.68, df = 2, p-value < 2.2e-16
```

```r
kruskal.test(TOTAL_TAP_IN_VOLUME ~ PEAK, data = combineddata[combineddata$DAY_TYPE ==
    "WEEKDAY" & combineddata$YEAR_MONTH == "2021-09", ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by PEAK
## Kruskal-Wallis chi-squared = 187.58, df = 2, p-value < 2.2e-16
```

```r
# Weekend
kruskal.test(TOTAL_TAP_IN_VOLUME ~ PEAK, data = combineddata[combineddata$DAY_TYPE !=
    "WEEKDAY" & combineddata$YEAR_MONTH == "2021-07", ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by PEAK
## Kruskal-Wallis chi-squared = 118.92, df = 2, p-value < 2.2e-16
```

```r
kruskal.test(TOTAL_TAP_IN_VOLUME ~ PEAK, data = combineddata[combineddata$DAY_TYPE !=
    "WEEKDAY" & combineddata$YEAR_MONTH == "2021-08", ])
```

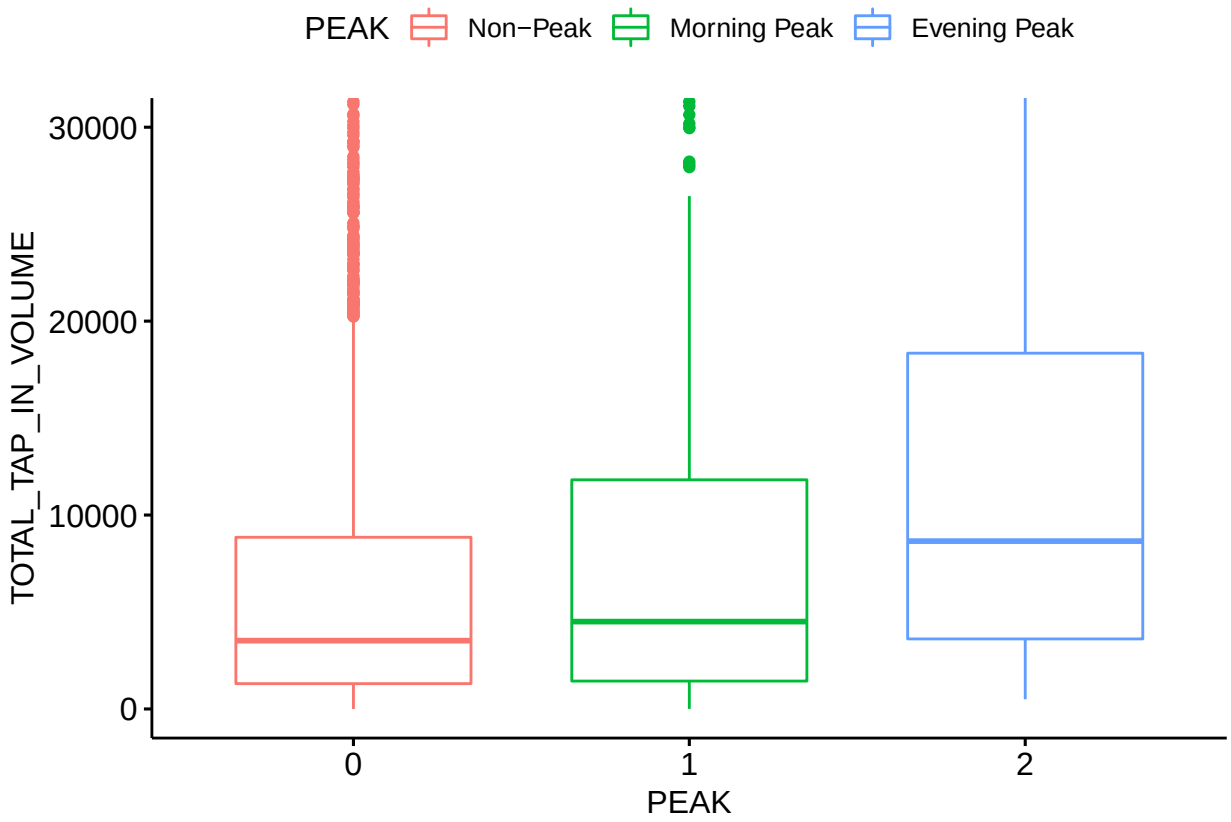
```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by PEAK
## Kruskal-Wallis chi-squared = 125.6, df = 2, p-value < 2.2e-16
```

```r
kruskal.test(TOTAL_TAP_IN_VOLUME ~ PEAK, data = combineddata[combineddata$DAY_TYPE !=
    "WEEKDAY" & combineddata$YEAR_MONTH == "2021-09", ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by PEAK
## Kruskal-Wallis chi-squared = 132.91, df = 2, p-value < 2.2e-16
```

```r
ggboxplot(combineddata[combineddata$DAY_TYPE == "WEEKDAY" & combineddata$YEAR_MONTH ==
    "2021-07", ], x = "PEAK", y = "TOTAL_TAP_IN_VOLUME", color = "PEAK",
    palette = c("#00AFBB", "#E7B800", "#FC4E07"), ylim = c(0,
        30000), ylab = "TOTAL_TAP_IN_VOLUME", xlab = "PEAK") +
    scale_color_hue(labels = c("Non-Peak", "Morning Peak", "Evening Peak"))
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```




```
ggsave("myplot.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
# Check if amount of passenger arrivals are different
# between months for morning/evening peak, non-peak ##

kruskal.test(TOTAL_TAP_IN_VOLUME ~ YEAR_MONTH, data = combineddata_set1[combineddata_set1$PEAK ==
    1, ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by YEAR_MONTH
## Kruskal-Wallis chi-squared = 0.30967, df = 2, p-value = 0.8566
```

```
kruskal.test(TOTAL_TAP_IN_VOLUME ~ YEAR_MONTH, data = combineddata_set1[combineddata_set1$PEAK ==
    2, ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by YEAR_MONTH
## Kruskal-Wallis chi-squared = 1.4017, df = 2, p-value = 0.4962
```

```
kruskal.test(TOTAL_TAP_IN_VOLUME ~ YEAR_MONTH, data = combineddata_set2[combineddata_set2$PEAK ==
    1, ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by YEAR_MONTH
## Kruskal-Wallis chi-squared = 0.69464, df = 2, p-value = 0.7066
```

```
kruskal.test(TOTAL_TAP_IN_VOLUME ~ YEAR_MONTH, data = combineddata_set2[combineddata_set2$PEAK ==
    2, ])
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by YEAR_MONTH
## Kruskal-Wallis chi-squared = 2.248, df = 2, p-value = 0.325
```

```
kruskal.test(TOTAL_TAP_IN_VOLUME ~ YEAR_MONTH, data = combineddata_set3)
```

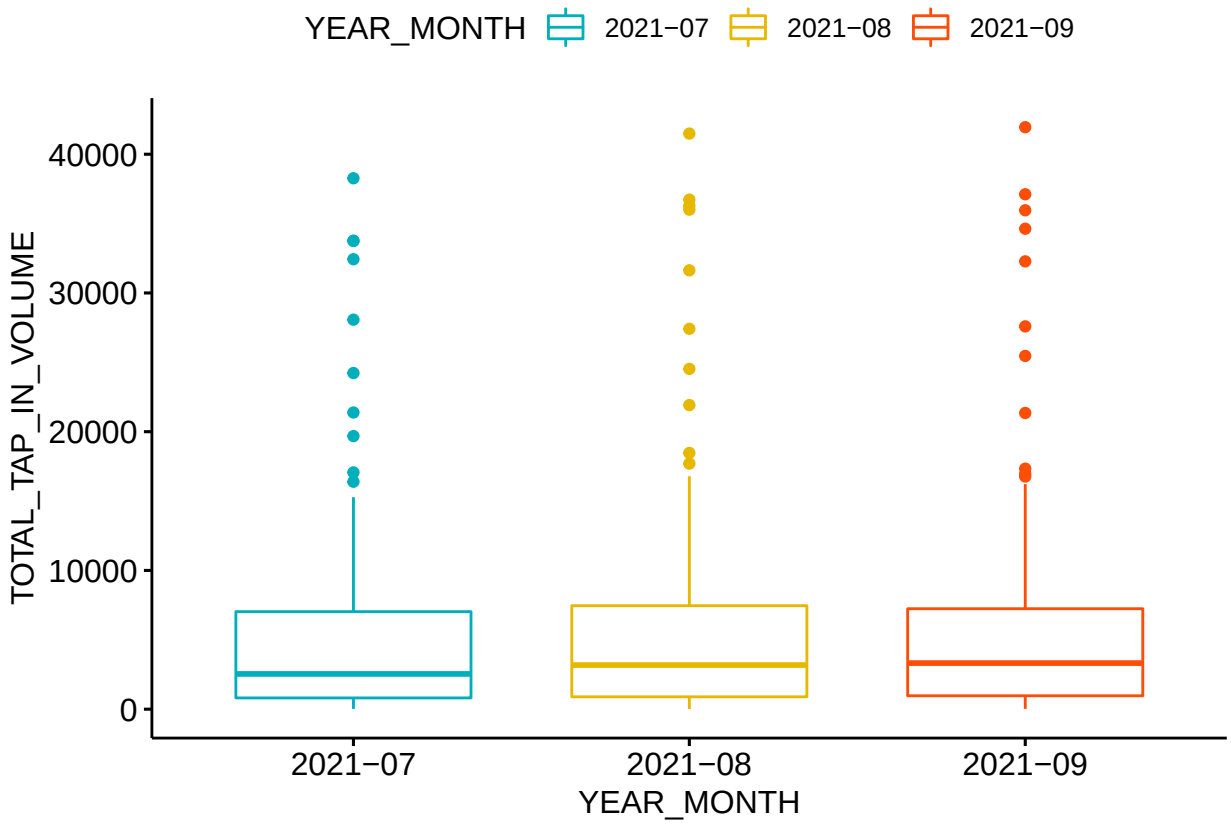
```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by YEAR_MONTH
## Kruskal-Wallis chi-squared = 1.8691, df = 2, p-value = 0.3928
```

```
kruskal.test(TOTAL_TAP_IN_VOLUME ~ YEAR_MONTH, data = combineddata_set4)
```

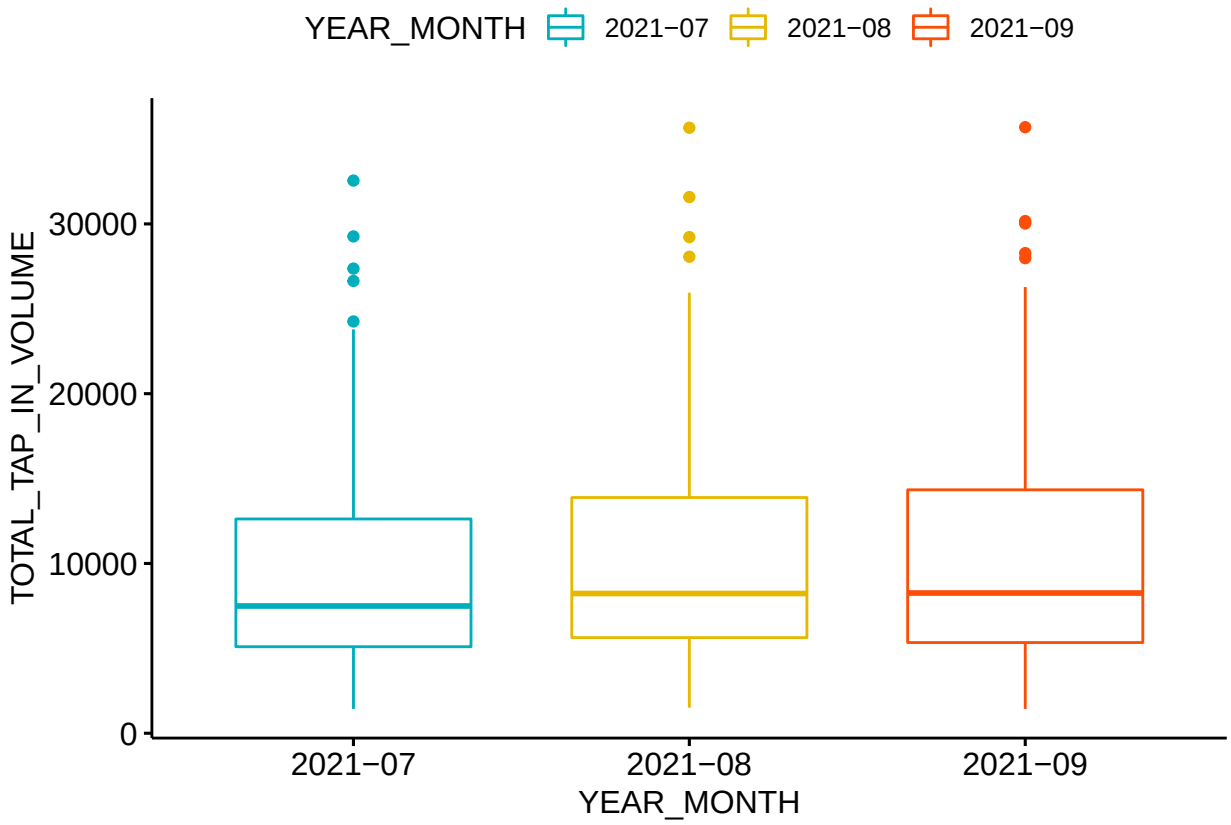
```
##
##  Kruskal-Wallis rank sum test
##
## data:  TOTAL_TAP_IN_VOLUME by YEAR_MONTH
## Kruskal-Wallis chi-squared = 4.3279, df = 2, p-value = 0.1149
```

Visualization using boxplots:

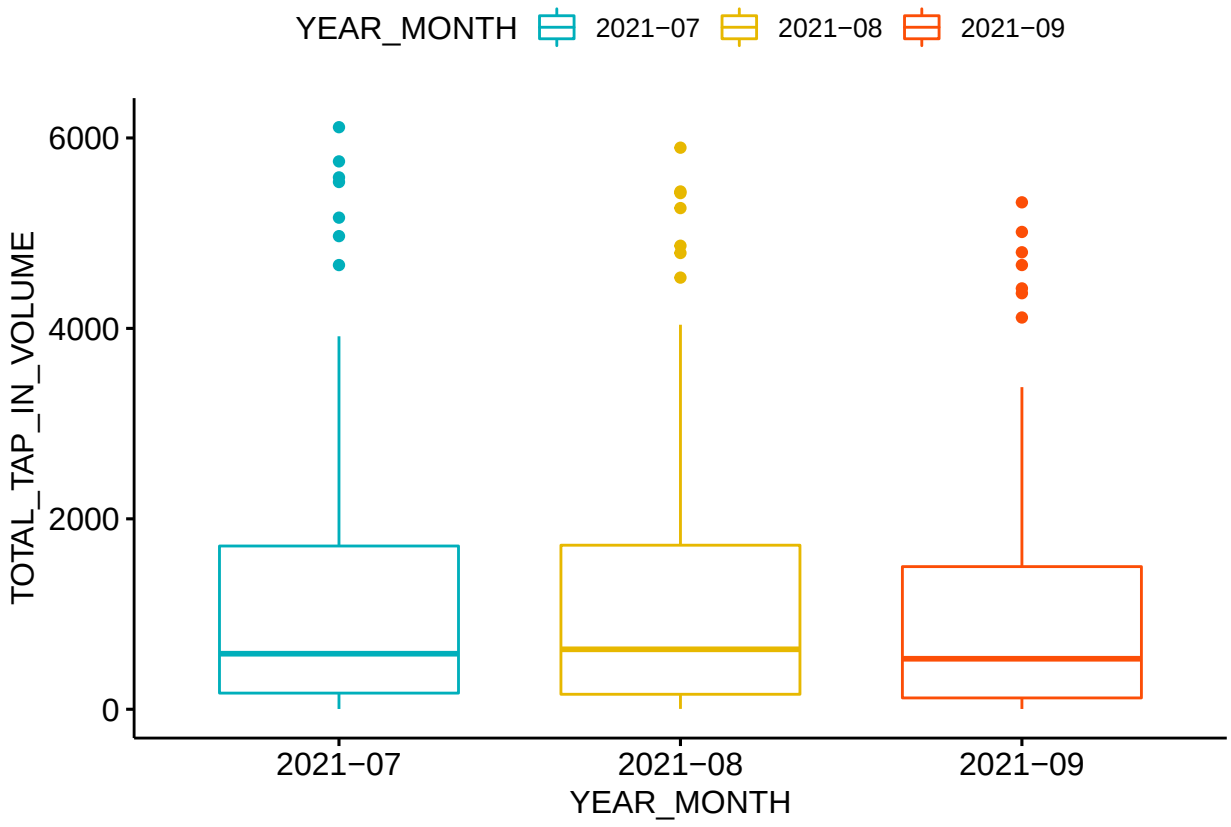
```
ggboxplot(combineddata_set1[combineddata_set1$PEAK == 1, ], x = "YEAR_MONTH",
    y = "TOTAL_TAP_IN_VOLUME", color = "YEAR_MONTH", palette = c("#00AFBB",
        "#E7B800", "#FC4E07"), ylab = "TOTAL_TAP_IN_VOLUME",
    xlab = "YEAR_MONTH")
```



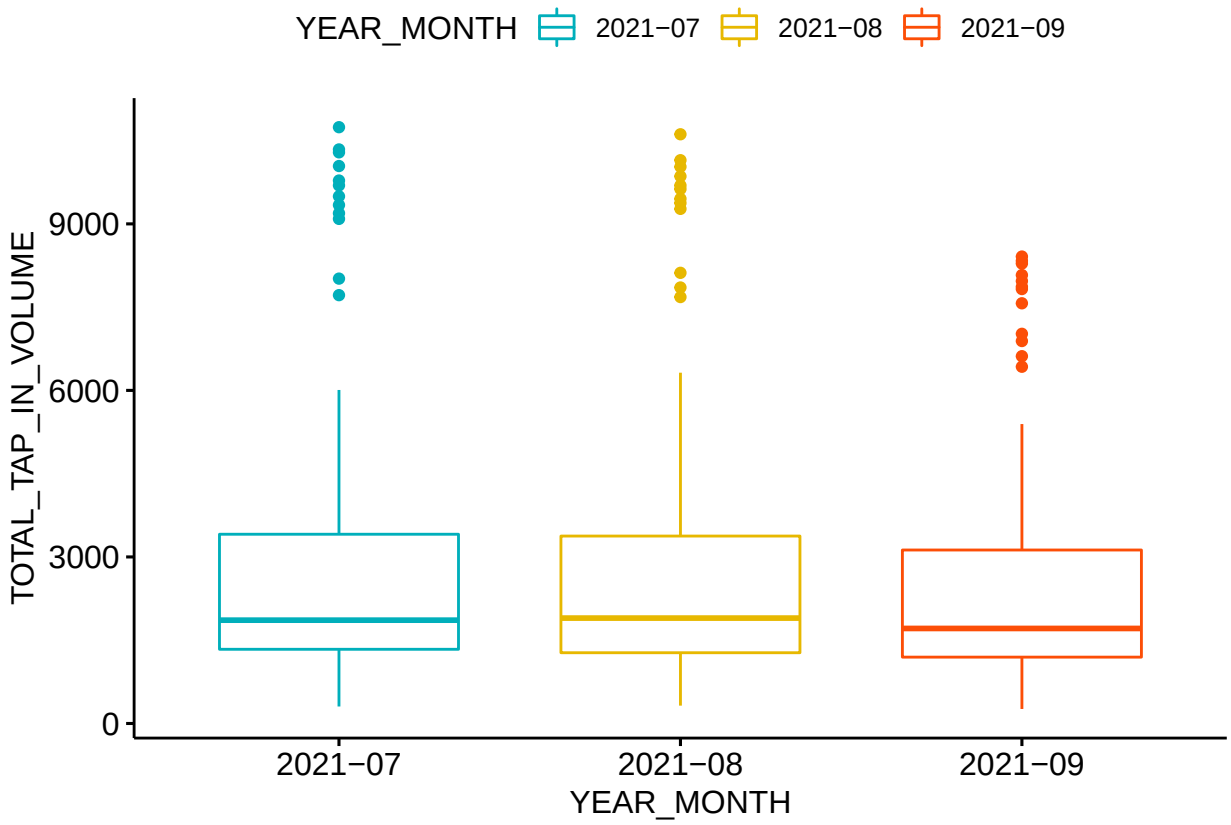

```
ggboxplot(combineddata_set1[combineddata_set1$PEAK == 2, ], x = "YEAR_MONTH",
    y = "TOTAL_TAP_IN_VOLUME", color = "YEAR_MONTH", palette = c("#00AFBB",
        "#E7B800", "#FC4E07"), ylab = "TOTAL_TAP_IN_VOLUME",
    xlab = "YEAR_MONTH")
```



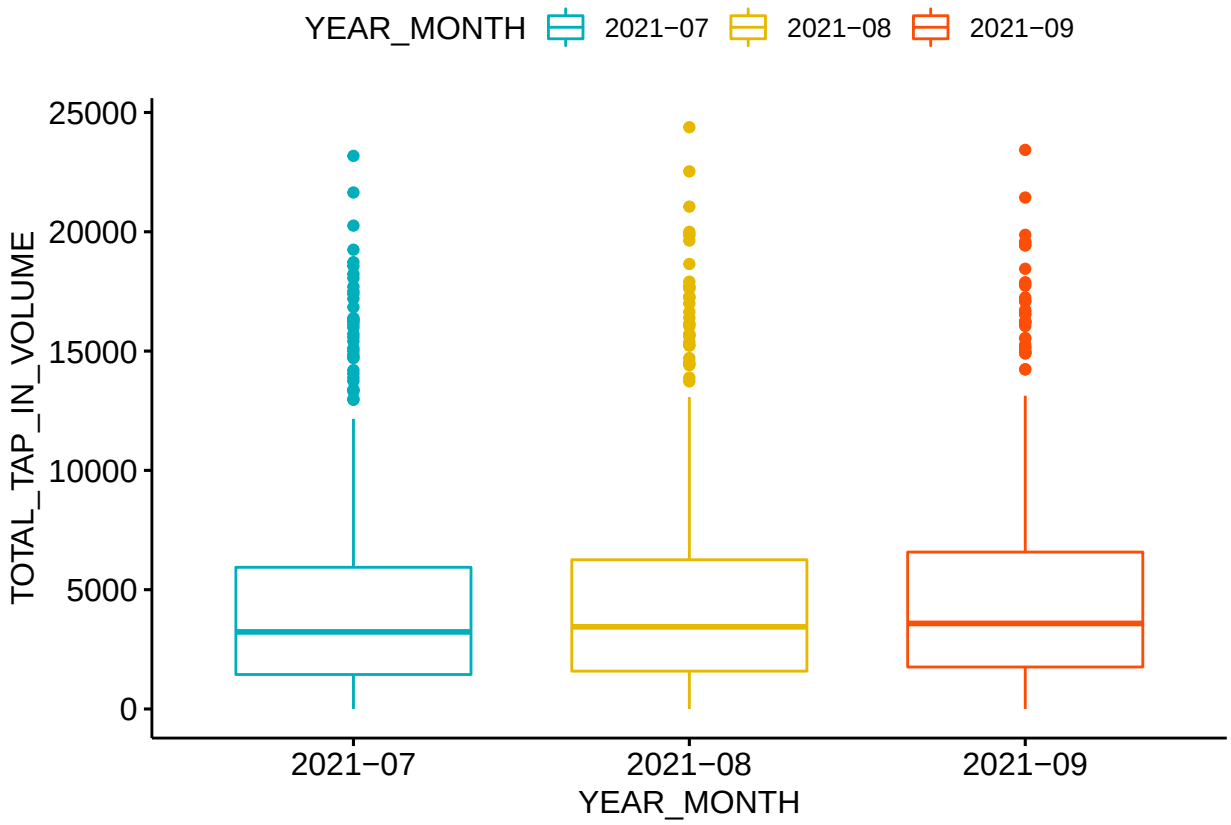

```
ggboxplot(combineddata_set2[combineddata_set2$PEAK == 1, ], x = "YEAR_MONTH",
    y = "TOTAL_TAP_IN_VOLUME", color = "YEAR_MONTH", palette = c("#00AFBB",
        "#E7B800", "#FC4E07"), ylab = "TOTAL_TAP_IN_VOLUME",
    xlab = "YEAR_MONTH")
```



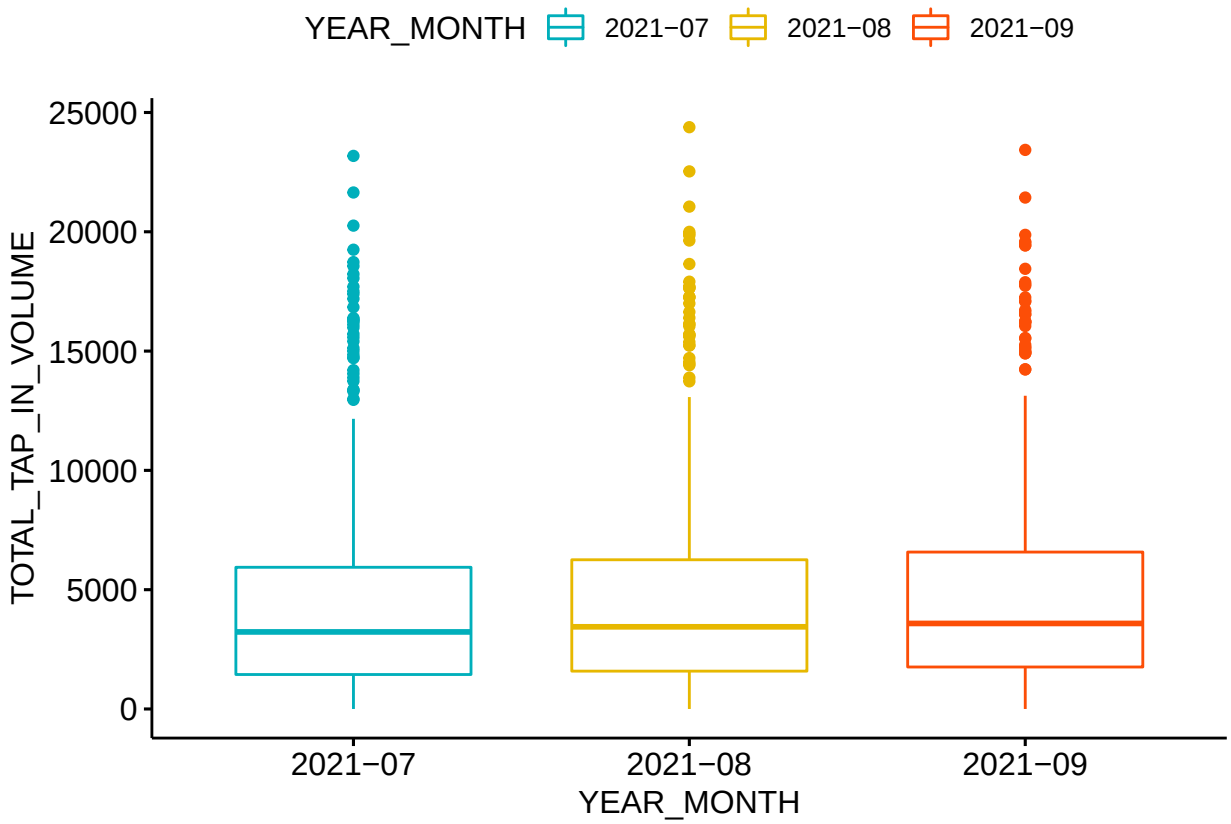

```
ggboxplot(combineddata_set2[combineddata_set2$PEAK == 2, ], x = "YEAR_MONTH",
    y = "TOTAL_TAP_IN_VOLUME", color = "YEAR_MONTH", palette = c("#00AFBB",
        "#E7B800", "#FC4E07"), ylab = "TOTAL_TAP_IN_VOLUME",
    xlab = "YEAR_MONTH")
```




```
ggboxplot(combineddata_set3, x = "YEAR_MONTH", y = "TOTAL_TAP_IN_VOLUME",
    color = "YEAR_MONTH", palette = c("#00AFBB", "#E7B800", "#FC4E07"),
    order = c("2021-07", "2021-08", "2021-09"), ylab = "TOTAL_TAP_IN_VOLUME",
    xlab = "YEAR_MONTH")
```




```
ggboxplot(combineddata_set3, x = "YEAR_MONTH", y = "TOTAL_TAP_IN_VOLUME",
    color = "YEAR_MONTH", palette = c("#00AFBB", "#E7B800", "#FC4E07"),
    order = c("2021-07", "2021-08", "2021-09"), ylab = "TOTAL_TAP_IN_VOLUME",
    xlab = "YEAR_MONTH")
```




```
# summary(aov(TOTAL_TAP_IN_VOLUME ~ YEAR_MONTH, data =
# combineddata_set1)) summary(aov(TOTAL_TAP_IN_VOLUME ~
# YEAR_MONTH, data = combineddata_set2))
# summary(aov(TOTAL_TAP_IN_VOLUME ~ YEAR_MONTH, data =
# combineddata_set3)) summary(aov(TOTAL_TAP_IN_VOLUME ~
# YEAR_MONTH, data = combineddata_set4))
```

Creating parameters values:

```
create_table <- function(data) {
    data$PEAK <- 0
    datapeakid <- data$TIME_PER_HOUR %in% peak
    data$PEAK[datapeakid] <- ifelse(data[datapeakid, ]$TIME_PER_HOUR %in%
        morningpeak, 1, 2)
    data$PEAK <- as.factor(data$PEAK)
    data$Connected <- str_count(data$PT_CODE, "/") + 1
    data$TOTAL_TAP_IN_VOLUME <- data$TOTAL_TAP_IN_VOLUME/data$Connected
    data$IS_DT <- as.numeric(grepl("DT", data$PT_CODE))

    weekday_morningpeak_arrival_frequency <- data$TOTAL_TAP_IN_VOLUME[data$DAY_TYPE ==
        "WEEKDAY" & data$PEAK == 1 & data$IS_DT == 1]
    weekday_eveningpeak_arrival_frequency <- data$TOTAL_TAP_IN_VOLUME[data$DAY_TYPE ==
        "WEEKDAY" & data$PEAK == 2 & data$IS_DT == 1]
    weekday_nonpeak_arrival_frequency <- data$TOTAL_TAP_IN_VOLUME[data$DAY_TYPE ==
        "WEEKDAY" & data$PEAK == 0 & data$IS_DT == 1]

    mean_weekday_morningpeak_arrival_frequency <- mean(weekday_morningpeak_arrival_frequency)
```

```
    mean_weekday_eveningpeak_arrival_frequency <- mean(weekday_eveningpeak_arrival_frequency)
    mean_weekday_nonpeak_arrival_frequency <- mean(weekday_nonpeak_arrival_frequency)

    weekend_morningpeak_arrival_frequency <- data$TOTAL_TAP_IN_VOLUME[data$DAY_TYPE !=
        "WEEKDAY" & data$PEAK == 1 & data$IS_DT == 1]
    weekend_eveningpeak_arrival_frequency <- data$TOTAL_TAP_IN_VOLUME[data$DAY_TYPE !=
        "WEEKDAY" & data$PEAK == 2 & data$IS_DT == 1]
    weekend_nonpeak_arrival_frequency <- data$TOTAL_TAP_IN_VOLUME[data$DAY_TYPE !=
        "WEEKDAY" & data$PEAK == 0 & data$IS_DT == 1]

    mean_weekend_morningpeak_arrival_frequency <- mean(weekend_morningpeak_arrival_frequency)
    mean_weekend_eveningpeak_arrival_frequency <- mean(weekend_eveningpeak_arrival_frequency)
    mean_weekend_nonpeak_arrival_frequency <- mean(weekend_nonpeak_arrival_frequency)

    frequencytable <- matrix(c(mean_weekday_morningpeak_arrival_frequency,
        mean_weekday_eveningpeak_arrival_frequency, mean_weekday_nonpeak_arrival_frequency,
        mean_weekend_morningpeak_arrival_frequency, mean_weekend_eveningpeak_arrival_frequency,
        mean_weekend_nonpeak_arrival_frequency), 2, 3, byrow = T)

    rownames(frequencytable) <- c("Weekday", "Weekend")
    colnames(frequencytable) <- c("MorningPeak", "EveningPeak",
        "NonPeak")

    return(frequencytable/60)
}

floor(create_table(combineddata))
```

```
##         MorningPeak EveningPeak NonPeak
## Weekday          85         133      58
## Weekend          14          33      23
```

```
round(1/floor(create_table(combineddata)), 4)
```

```
##         MorningPeak EveningPeak NonPeak
## Weekday      0.0118      0.0075  0.0172
## Weekend      0.0714      0.0303  0.0435
```