EDA - Bike-Sharing in Boston

Soham Shinde

9/30/2021

Problem 1

Find a dataset that is personally interesting to you. It may be a publicly-available dataset, or a dataset for which you have permission to use and share results. There are many places online to find publicly-available dataset, and simply searching Google for your preferred topic plus “public dataset” may provide many hits.

Import the dataset into R, tidy the dataset (if necessary), and print the first several lines of the dataset.

Describe the dataset and its variables. Comment on whether you had to tidy the dataset, and how you tidied the data (if you did).

The Dataset: Bluebikes in Boston

BlueBikes is a bicycle sharing system which operates in the city of Boston and neighboring municipalities. The user needs to pick up a bike from a station, ride it for specific time and return the bike to any station. The BlueBikes app tracks the time duration for each trip.

There are two datasets: bluebikes_tripdata_2019.csv - (2.52 million trips) bluebikes_tripdata_2020.csv - (2 million trips)

Variables considered for analysis: 1. tripduration: duration of trip (seconds) 2. start station name: name of station at which the ride started 3. start station latitude: latitude of start station 4. start station longitude: longitude of start station 5. end station name: name of station at which the ride ended 6. end station latitude: latitude of end station 7. end station longitude: longitude of end station 8. bikeid: unique ID of bike used 9. usertype: type of user (Customer or Subscriber) 10. month: month of the trip 11. birth year: birth year 12. gender: gender

Missing Values: bluebikes_tripdata_2019.csv = 0 bluebikes_tripdata_2020.csv = 3.7 Million (approx.) postal code = 27.87% gender = 79.19% birth year = 79.19%

The dataset did not require tidying of data.

• Derive a new Column ‘age’ from Column ‘birth year’. • Add a new Column ‘distance’ by calculating distance between two coordinates using Haversine Formula.

```r
#import packages
library(readr)
library(tibble)

#Read the csv file into a tibble
data19 = tibble(read_csv("bluebikes_tripdata_2019.csv"))
```

```
## Rows: 2522771 Columns: 17
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (3): start station name, end station name, usertype
## dbl  (12): tripduration, start station id, start station latitude, start sta...
## dttm  (2): starttime, stoptime
```

```
##
## 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.
```

```
data20 = tibble(read_csv("bluebikes_tripdata_2020.csv"))
```

```
## Rows: 1999446 Columns: 18
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): start station name, end station name, usertype, postal code
## dbl  (12): tripduration, start station id, start station latitude, start sta...
## dttm  (2): starttime, stoptime
```

```
##
## 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.
```

```
head(data19)
```

```
## # A tibble: 6 x 17
##   tripduration starttime           stoptime            `start station id`
##          <dbl> <dttm>              <dttm>                           <dbl>
## 1          790 2019-12-01 00:01:25 2019-12-01 00:14:35                370
## 2          166 2019-12-01 00:05:42 2019-12-01 00:08:29                 80
## 3          323 2019-12-01 00:08:28 2019-12-01 00:13:52                381
## 4          709 2019-12-01 00:08:38 2019-12-01 00:20:27                185
## 5          332 2019-12-01 00:10:08 2019-12-01 00:15:41                221
## 6          507 2019-12-01 00:14:26 2019-12-01 00:22:53                177
## # ... with 13 more variables: start station name <chr>,
## #   start station latitude <dbl>, start station longitude <dbl>,
## #   end station id <dbl>, end station name <chr>, end station latitude <dbl>,
## #   end station longitude <dbl>, bikeid <dbl>, usertype <chr>,
## #   birth year <dbl>, gender <dbl>, year <dbl>, month <dbl>
```

Problem 2 Useggplot2to create visualizations to identify interesting or unexpected relationships in the dataset. After performing your analysis, present your results by creating an attractive “Miniposter” slide using PowerPoint, Keynote, or similar program. Submit your slide to the “Miniposter” assignment on Canvas. In your homework solutions, reproduce the plots from your “Miniposter” figures, and provide your interpretations of them.

```
#Apply aggregate function
sub2 = aggregate(data.frame(value = data19$`start station name`),
          list(StartStationName = data19$`start station name`),
          length)
sub2n = aggregate(data.frame(value = data20$`start station name`),
          list(StartStationName = data20$`start station name`),
          length)

#386 stations

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

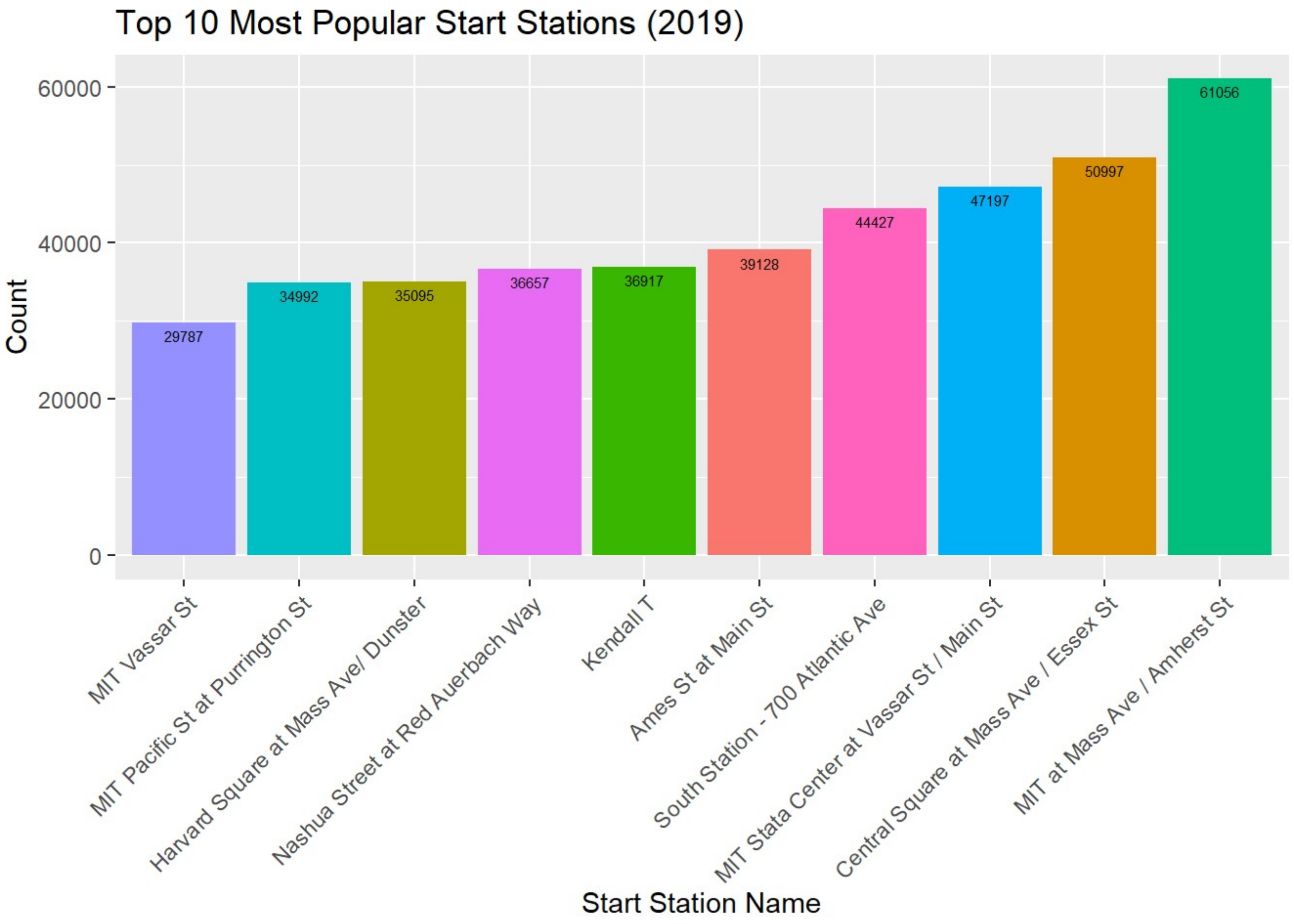
```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

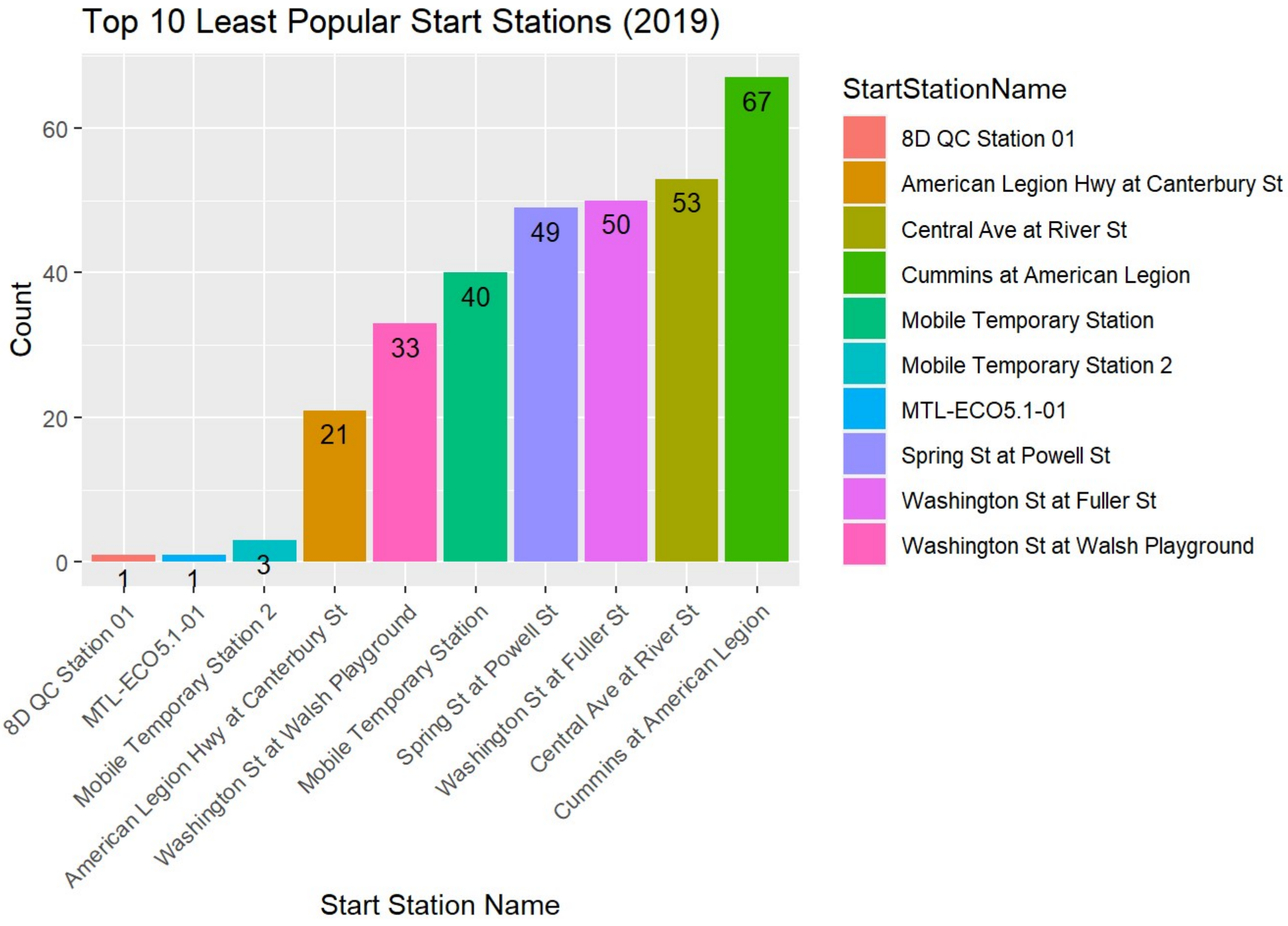
```
## v tidyr   1.1.3     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

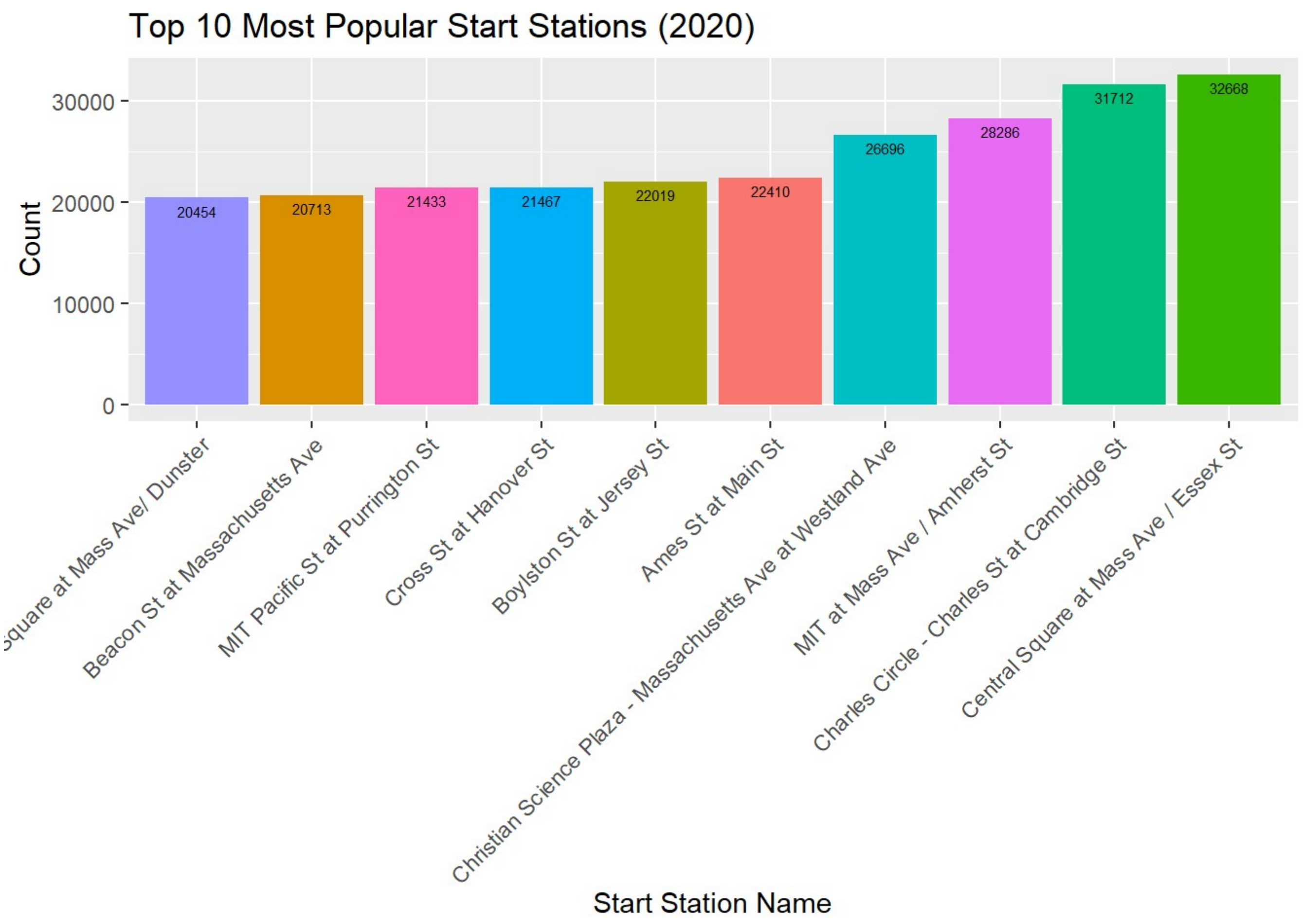
```
#arrange in order
sub3 = head(arrange(sub2,desc(value)),10)
sub3a = head(arrange(sub2,value),10)

#use Bar-Plot
ggplot(data=sub3, aes(x=reorder(StartStationName, value), y=value, fill=StartStationN
ame )) +
  geom_bar(stat="identity")+
  labs(title="Top 10 Most Popular Start Stations (2019)",
       x = "Start Station Name", y = "Count")+
  geom_text(aes(label=value), vjust=1.6, color="black", size=2)+
  theme(axis.text.x=element_text(angle=45, hjust=1),legend.position = "none")
```



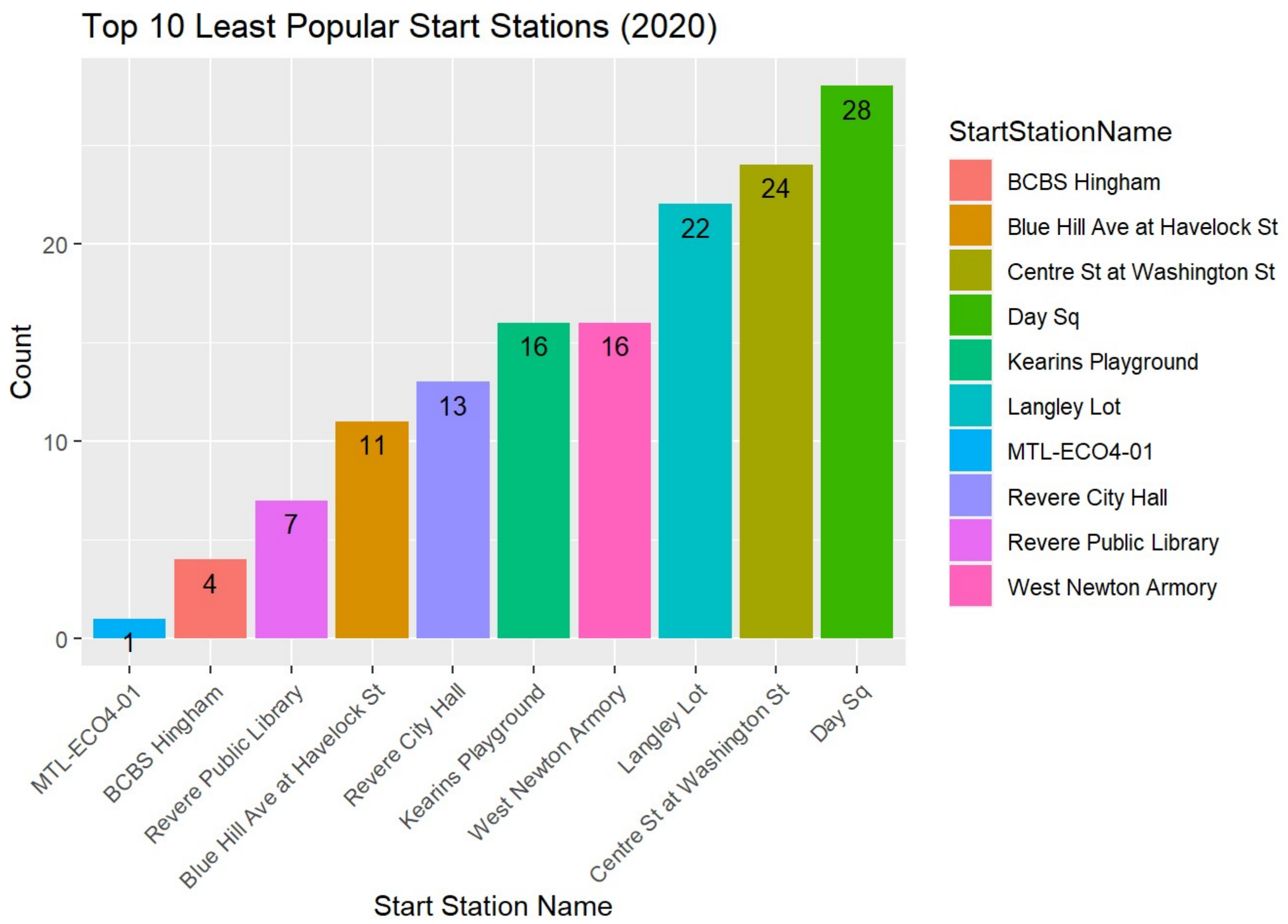
```
ggplot(data=sub3a, aes(x=reorder(StartStationName, value), y=value, fill=StartStation
Name )) +
  geom_bar(stat="identity")+
  labs(title="Top 10 Least Popular Start Stations (2019)",
       x = "Start Station Name", y = "Count")+
  geom_text(aes(label=value), vjust=1.6, color="black", size=3.5)+
  theme(axis.text.x=element_text(angle=45, hjust=1))
```



```
#arrange in order
sub3 = head(arrange(sub2n,desc(value)),10)
sub3a = head(arrange(sub2n,value),10)

#use Bar-Plot
ggplot(data=sub3, aes(x=reorder(StartStationName, value), y=value, fill=StartStationN
ame )) +
  geom_bar(stat="identity")+
  labs(title="Top 10 Most Popular Start Stations (2020)",
       x = "Start Station Name", y = "Count")+
  geom_text(aes(label=value), vjust=1.6, color="black", size=2)+
  theme(axis.text.x=element_text(angle=45, hjust=1),legend.position = "none")
```

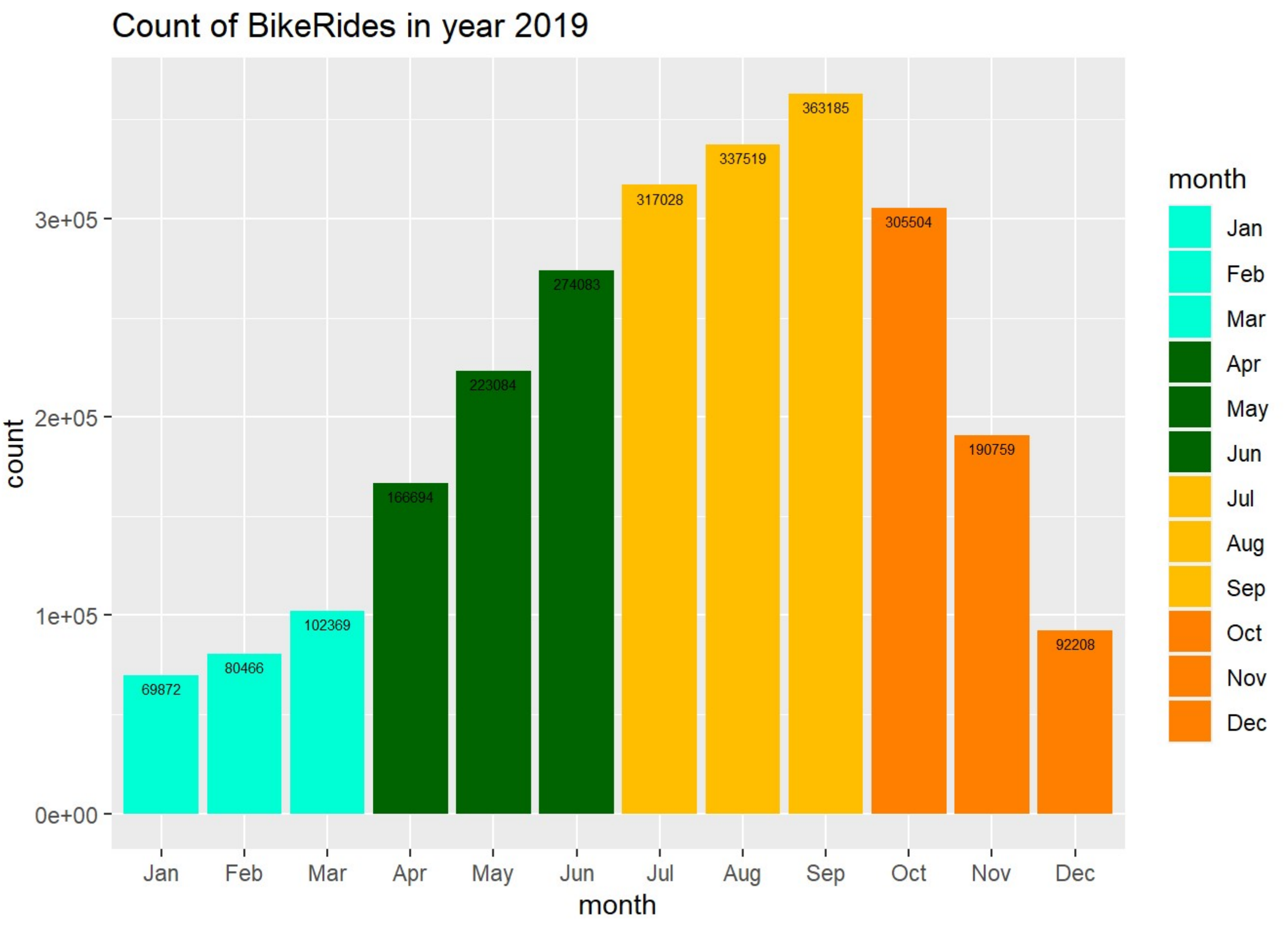


```
ggplot(data=sub3a, aes(x=reorder(StartStationName, value), y=value, fill=StartStation
Name )) +
  geom_bar(stat="identity")+
  labs(title="Top 10 Least Popular Start Stations (2020)",
       x = "Start Station Name", y = "Count")+
  geom_text(aes(label=value), vjust=1.6, color="black", size=3.5)+
  theme(axis.text.x=element_text(angle=45, hjust=1), )
```

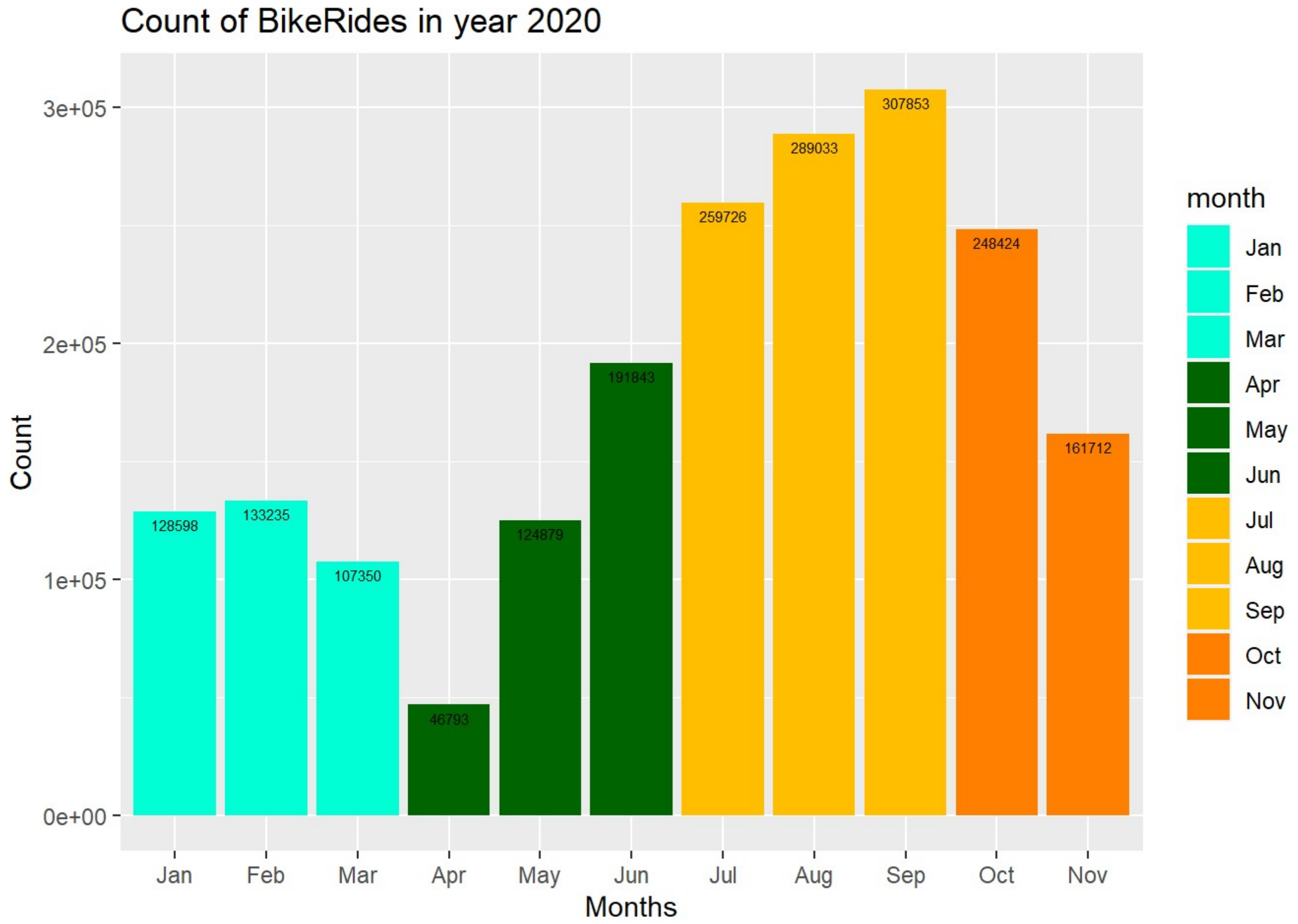


Observation: In 2019: The most popular start station was MIT at Mass Ave / Amherst St station while the least popular start station was 8D QC Station 01. Most Popular Stations in terms of count of trips are located near the universities. Least Popular Stations comprise some temporary stations. In 2020: Most Popular Station changed from MIT at Mass Ave / Amherst St station to Central Square at Mass Ave / Essex St as a result of many universities offering remote study option during the pandemic. Most Popular Stations have less count of those located near universities due to the the pandemic. Also the count of the Most Popular Station has reduced to half from 2019 to 2020.

```
d1 = data19

d1$month = as.factor(d1$month)
d1 = d1[order(d1$month),]
d1$month = month.abb[d1$month]

nameOfMonths = c("Jan","Feb","Mar",
              "Apr","May","Jun",
              "Jul","Aug","Sep",
              "Oct","Nov","Dec")

d1$month <- factor(d1$month, levels=nameOfMonths)

ggplot(data = d1,aes(x = month, fill = month)) +
  geom_bar() +
  geom_text(stat='count', aes(label=..count..), vjust=1.6, color = "black", size = 2)
+
  labs(title="Count of BikeRides in year 2019")+
  scale_fill_manual(values=c("#00ffd5","#00ffd5","#00ffd5",
                             "#006400", "#006400","#006400","#ffbf00",
                             "#ffbf00", "#ffbf00", "#ff8000", "#ff8000", "#ff8000"))
```

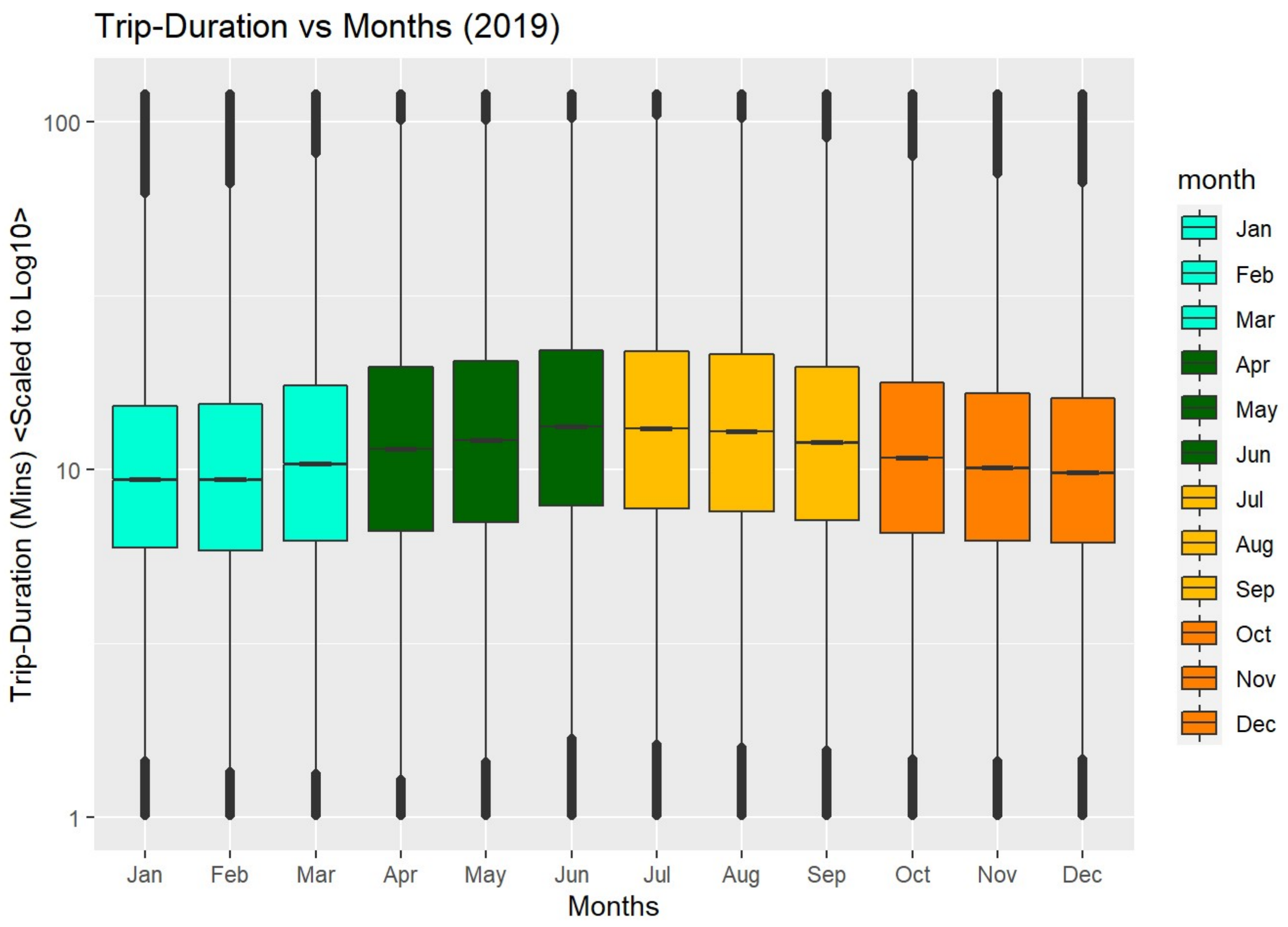


```
d2 = data20

d2$month = as.factor(d2$month)
d2 = d2[order(d2$month),]
d2$month = month.abb[d2$month]

d2$month <- factor(d2$month, levels=nameOfMonths)

ggplot(data = d2,aes(x = month, fill = month)) +
  geom_bar() +
  geom_text(stat='count', aes(label=..count..), vjust=1.6, color = "black", size = 2)
+
  labs(title="Count of BikeRides in year 2020", x = "Months", y ="Count")+
  scale_fill_manual(values=c("#00ffd5","#00ffd5","#00ffd5", "#006400",
                             "#006400","#006400","#ffbf00", "#ffbf00",
                             "#ffbf00", "#ff8000", "#ff8000", "#ff8000"))
```

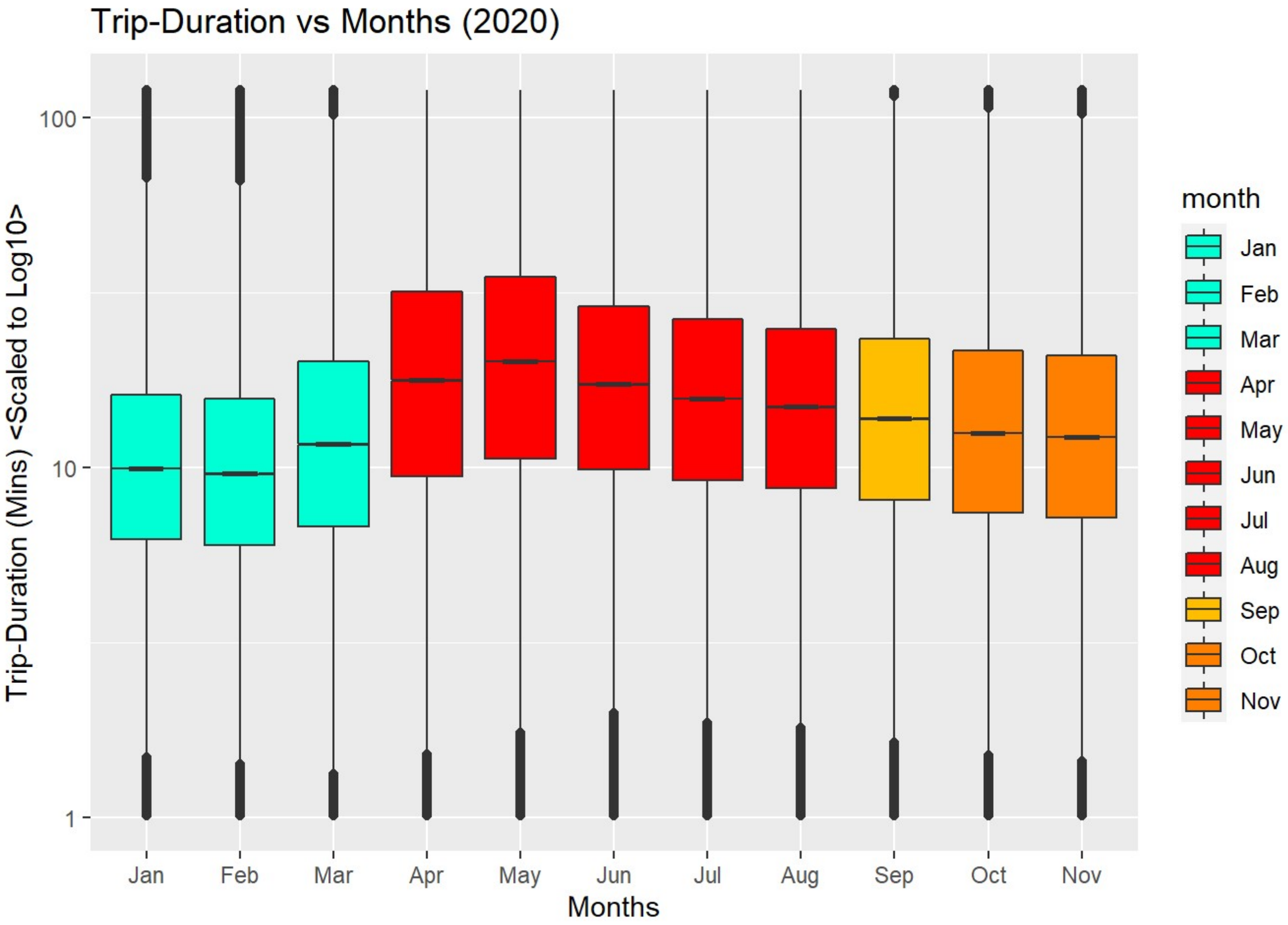


Observation: In 2019: Number of Bike-Rides increases in the Summer and initial Fall months (June, Aug, Sep, Oct) but then decreases by almost 1/4th in the Winter Months(Dec, Jan, Feb). In 2020: Count of Bike-Rides decreases significantly with most decrease in the month of April where the Pandemic was at its peak in the 1st wave. Then, it again increases from the month of May and June but the count is less as compared to last year rides in the same months.

```
d1 = data19[data19$tripduration<7200,c("month", "tripduration")]

d1$month = as.factor(d1$month)
d1 = d1[order(d1$month),]

d1$tripduration = (d1$tripduration)/60
d1$month = month.abb[d1$month]

nameOfMonths = c("Jan","Feb","Mar",
              "Apr","May","Jun",
              "Jul","Aug","Sep",
              "Oct","Nov","Dec")

d1$month <- factor(d1$month, levels=nameOfMonths)

library(ggplot2)


ggplot(d1, aes(x=month, y=tripduration, fill=month)) +
  geom_boxplot(notch = TRUE)+
  labs(title="Trip-Duration vs Months (2019)",x="Months",
       y = "Trip-Duration (Mins) <Scaled to Log10>")+
  scale_y_log10()+
  scale_fill_manual(values=c("#00ffd5","#00ffd5","#00ffd5",
                             "#006400", "#006400","#006400","#ffbf00",
                             "#ffbf00", "#ffbf00", "#ff8000", "#ff8000",
                             "#ff8000"))
```

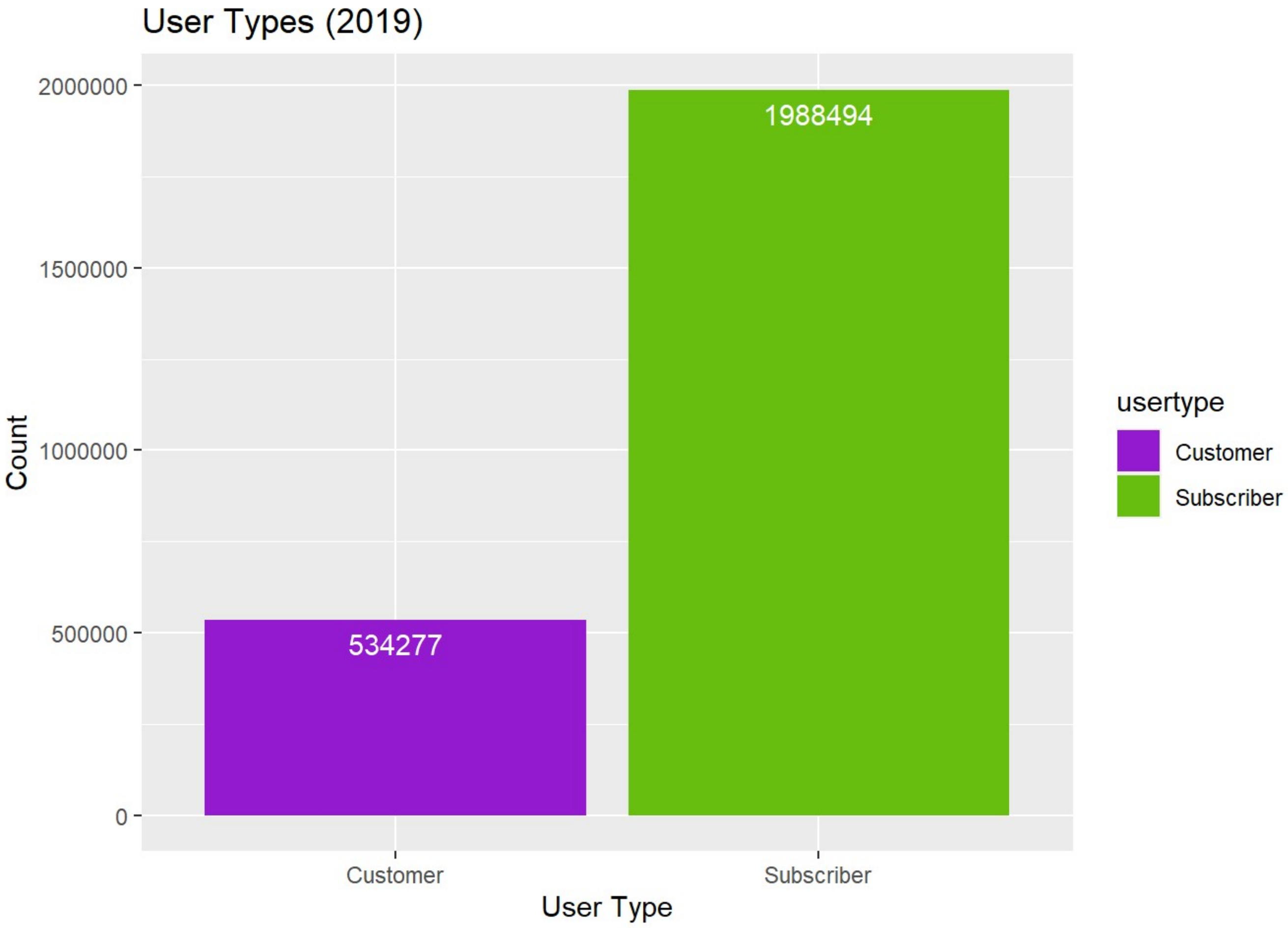


```
d2 = data20[data20$tripduration<7200,c("month", "tripduration")]

d2$month = as.factor(d2$month)
d2 = d2[order(d2$month),]

d2$tripduration = (d2$tripduration)/60
d2$month = month.abb[d2$month]

d2$month <- factor(d2$month, levels=nameOfMonths)

ggplot(d2, aes(x=month, y=tripduration, fill=month)) +
  geom_boxplot(notch = TRUE)+
  labs(title="Trip-Duration vs Months (2020)",x="Months",
       y = "Trip-Duration (Mins) <Scaled to Log10>")+
  scale_y_log10()+
  scale_fill_manual(values=c("#00ffd5","#00ffd5","#00ffd5",
                             "red", "red","red","red","red", "#ffbf00",
                             "#ff8000", "#ff8000", "#ff8000"))
```

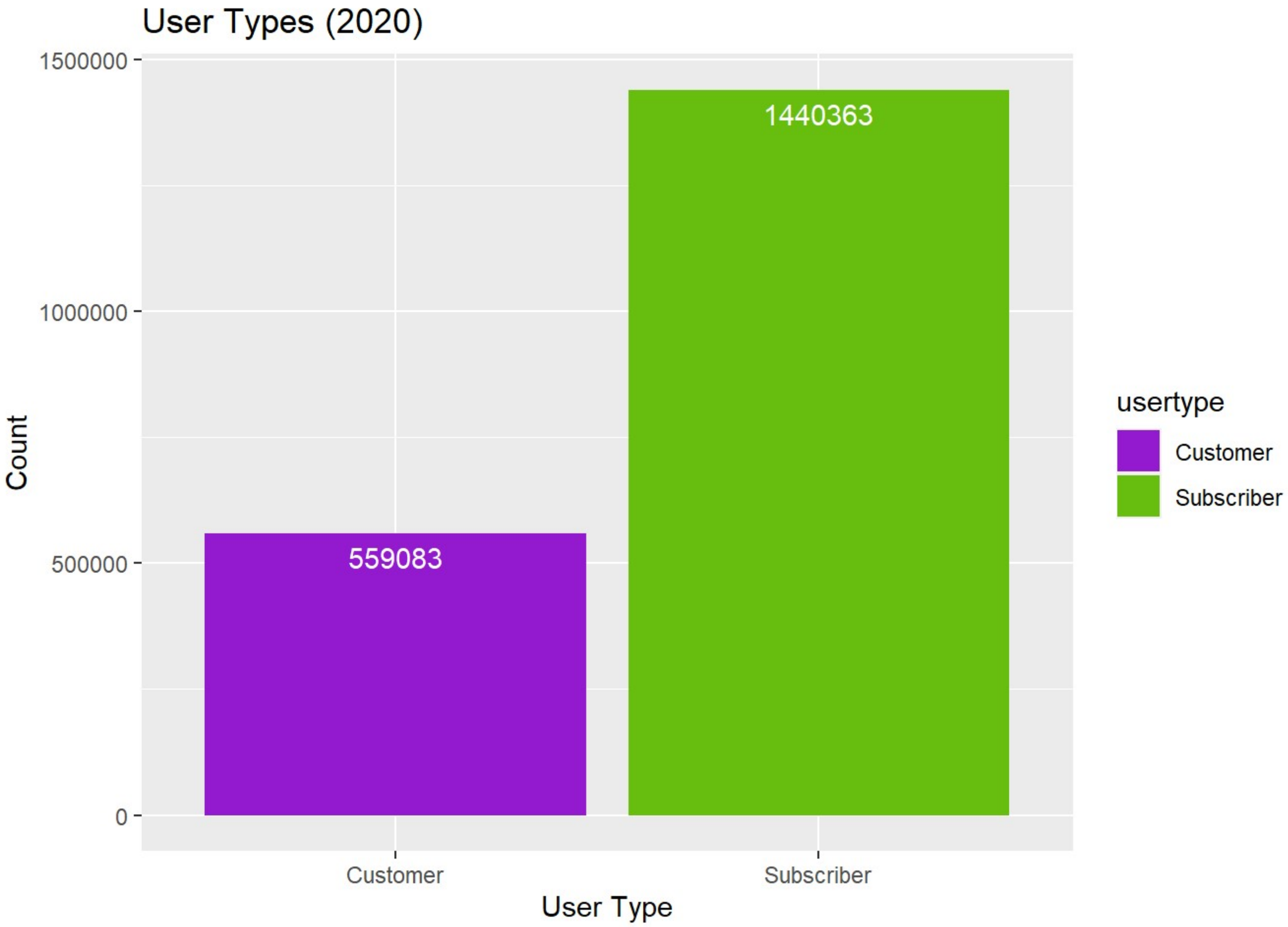


Observation: in 2019: Even after scaling using Log, all the medians lie very close to each other with respect to y axis. The Interqaurtile range is almost same for plots of every month. There is a slight increase in trip duration for the months of Apr to Sep which is in Summer and Fall seasons. in 2020: Interestingly, the median of trip duration increased for the months which experienced 1st wave of Covid (Apr-Aug) when compared to 2019.

```
ggplot(data = data19,aes(x = usertype, fill=usertype)) +
  geom_bar() +
  geom_text(stat='count', aes(label=..count..), vjust=1.6, color = "white")+
  labs(title="User Types (2019)",x="User Type", y = "Count")+
  scale_fill_manual(values=c("#931ACF","#67BE0E"))
```



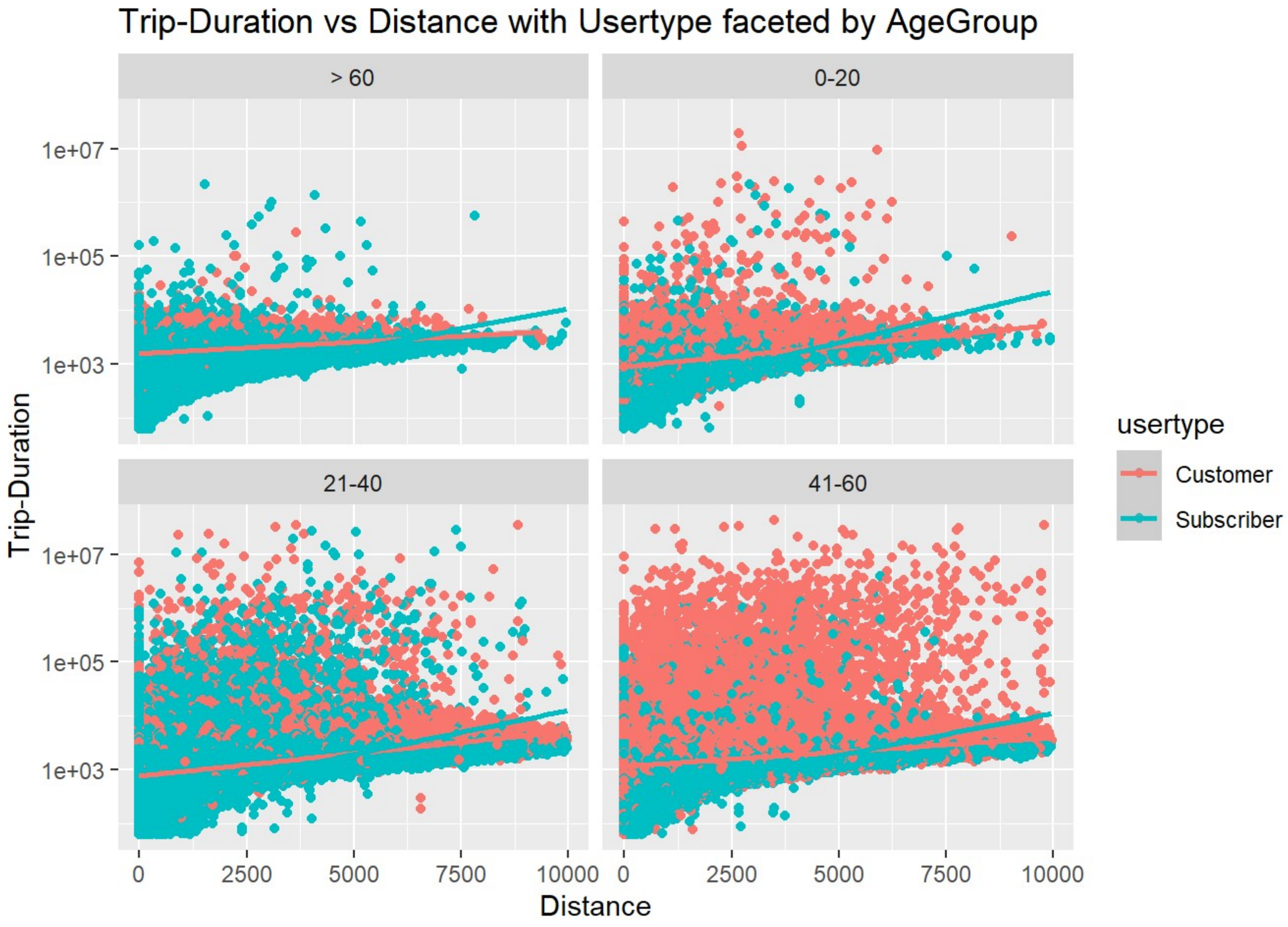
```
ggplot(data = data20,aes(x = usertype, fill=usertype)) +
  geom_bar() +
  geom_text(stat='count', aes(label=..count..), vjust=1.6, color = "white")+
  labs(title="User Types (2020)",x="User Type", y = "Count")+
  scale_fill_manual(values=c("#931ACF","#67BE0E"))
```



Observation: The no. of customers has increased in 2020 compared to previous year while that of Subscibers is decreased. Most of the users may have not issude or renewed their subsription due to the pandemic situation in 2020. Overall, the no. of customer are significantly less than no. of subscribers.

```
library(geosphere)

my_dist <- function(long1, lat1, long2, lat2) {
  rad <- pi/180
  a1 <- lat1*rad
  a2 <- long1*rad
  b1 <- lat2*rad
  b2 <- long2*rad
  dlon <- b2 - a2
  dlat <- b1 - a1
  a <- (sin(dlat/2))^2 + cos(a1)*cos(b1)*(sin(dlon/2))^2
  c <- 2*atan2(sqrt(a), sqrt(1 - a))
  R <- 6378137
  d <- R*c
  return(d)
}

data19$dist = my_dist(data19$"start station longitude",
                      data19$"start station latitude",
                      data19$"end station longitude",
                      data19$"end station latitude")
data20$dist = my_dist(data20$"start station longitude",
                      data20$"start station latitude",
                      data20$"end station longitude",
                      data20$"end station latitude")
```

```
d1 = data19[data19$dist<10000,c("gender", "dist","tripduration",
                                "usertype","birth year")]
d1$gender = as.factor(d1$gender)

library(ggplot2)

d1$age = 2019 - d1$"birth year"

d1[d1$age <= 20, "age_group"] <- "0-20"
d1[d1$age > 20 & d1$age <=40, "age_group"] <- "21-40"
d1[d1$age > 40 & d1$age <=60, "age_group"] <- "41-60"
d1[d1$age > 60, "age_group"] <- "> 60"

ggplot(d1, aes(x=dist, y=tripduration, color=usertype)) +
  geom_point()+
  facet_wrap(~age_group) +
  geom_smooth(method=lm, formula = 'y ~ x')+
  scale_y_log10()+
  labs(title="Trip-Duration vs Distance with Usertype faceted by AgeGroup",
       x="Distance", y = "Trip-Duration")
```



Observation: Age group 41-60 has the greater number of customers when compared to other age groups. Most of the points are concentrated near the smooth lines in age group >60. The trip duration for customers is more than subscribers overall. >60 have more number of subscribers than customers. The smooth line for all age groups grows in similar way with subscribers being more than customers at all times.