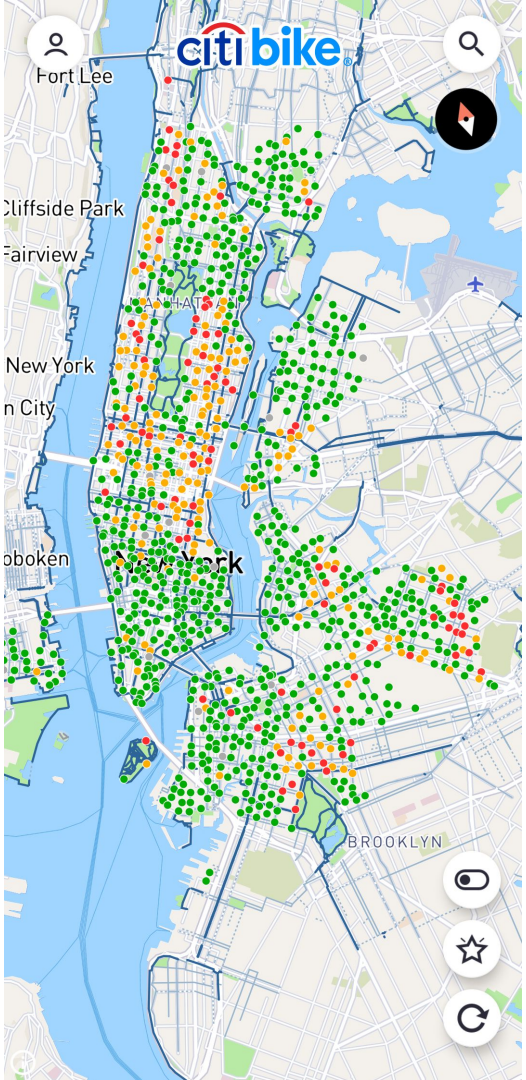
Analyzing CitiBike and Taxi Data in NYC

Jaival Desai, Julia Muallem, Chris Kevin Andrade, AnaPatricia Olvera

Background

- Citi Bike launched in May of 2013 with 330 stations and 5,000 bikes in only a small section of New York City.
    - Member: unlimited 45 min rides ($179 annually)
    - Subscriber: one 30 min ride ($3) or unlimited 30 min rides for 24hrs ($12)
- Growing in popularity, it has expanded to 14,500 bikes and 950 stations (1,000 electric bikes by the end of this year).
- Citi Bike has played a vital role in the in the mobility of the New York City.



- The New York City Subway system is limited.
- Most Citi Bike stations are within a 5 minute walk from subway entrances.
    - Citi Bike has become a solution for the last mile problem.

CitiBike Data Analysis

Cleaning the data

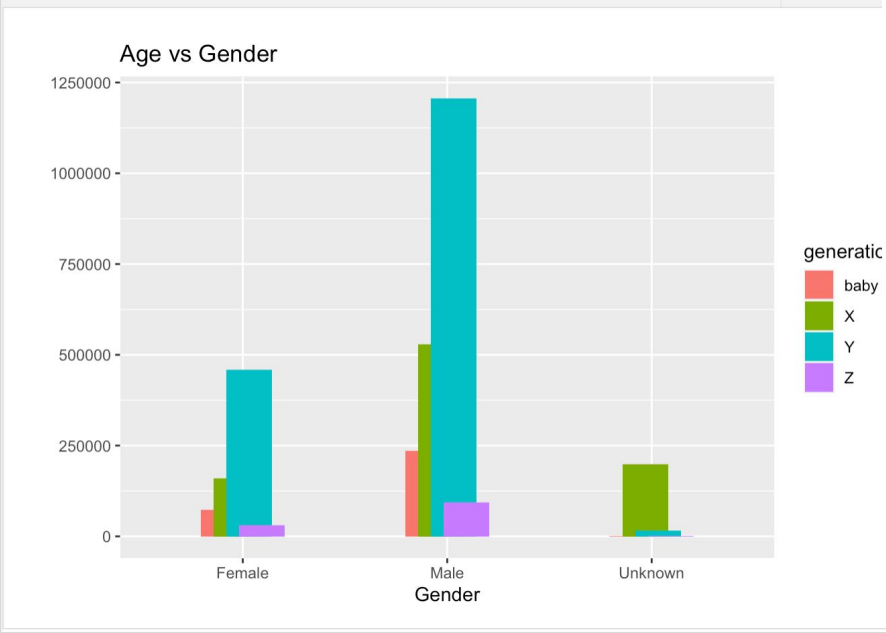

```r
augDecData <- augDecData%>% na.omit()
```

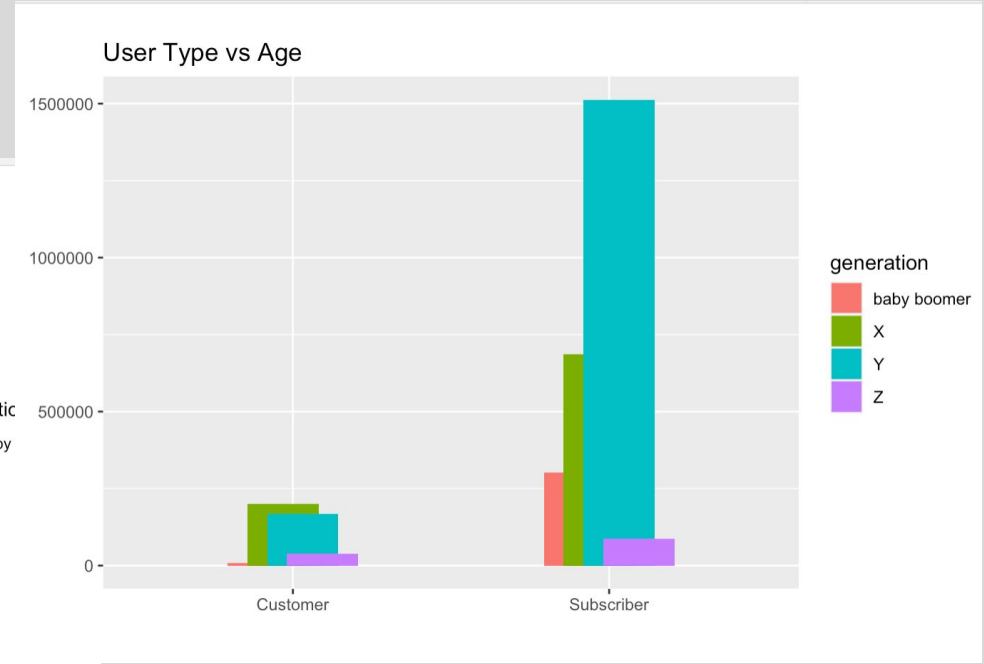
```r
augDecData <- augDecData %>% filter( tripduration >=0 & tripduration <= 1800)
```

```r
augDecData <- augDecData %>% filter( (2019 - birth.year ) <= 70)
```

Source: https://www.citibikenyc.com/system-data

Demographics





+Baby Boomer: 1946-1964
+Generation X: 1965-1979
+Generation Y (also known as Millennial): 1980-1996
+Generation Z: 1997-2010

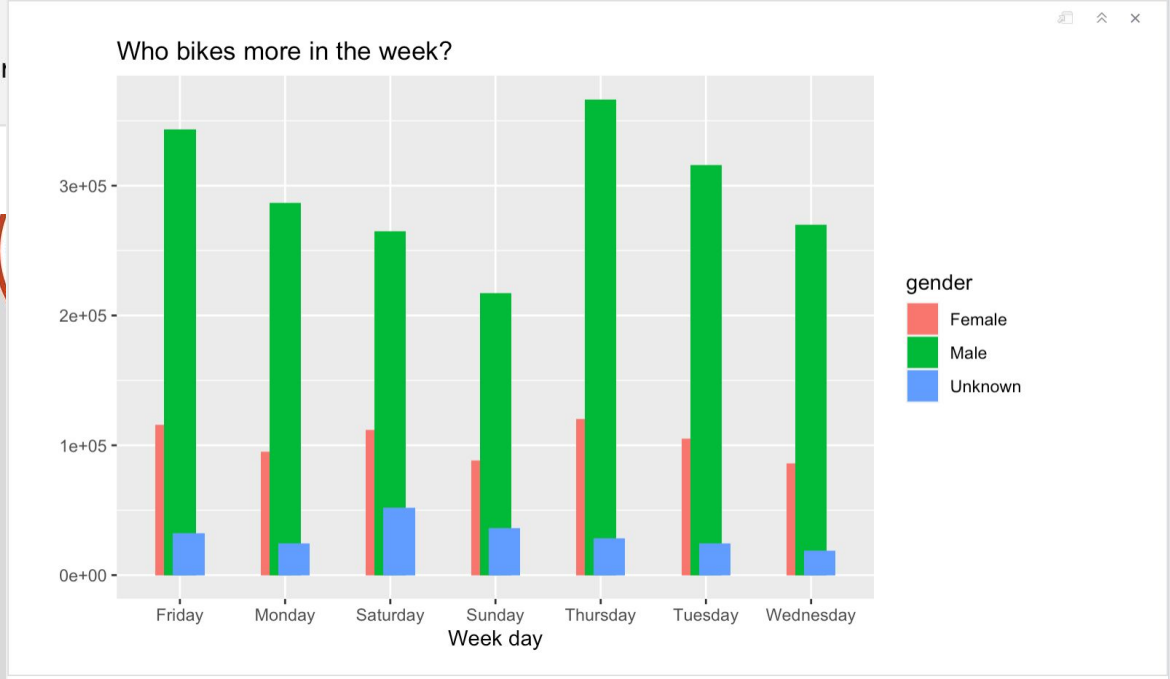
Average trip duration

```r
#average in minutes
mean( augDecData$tripduration, tr
```

```
[1] 11.33524
```

The busiest day of the week

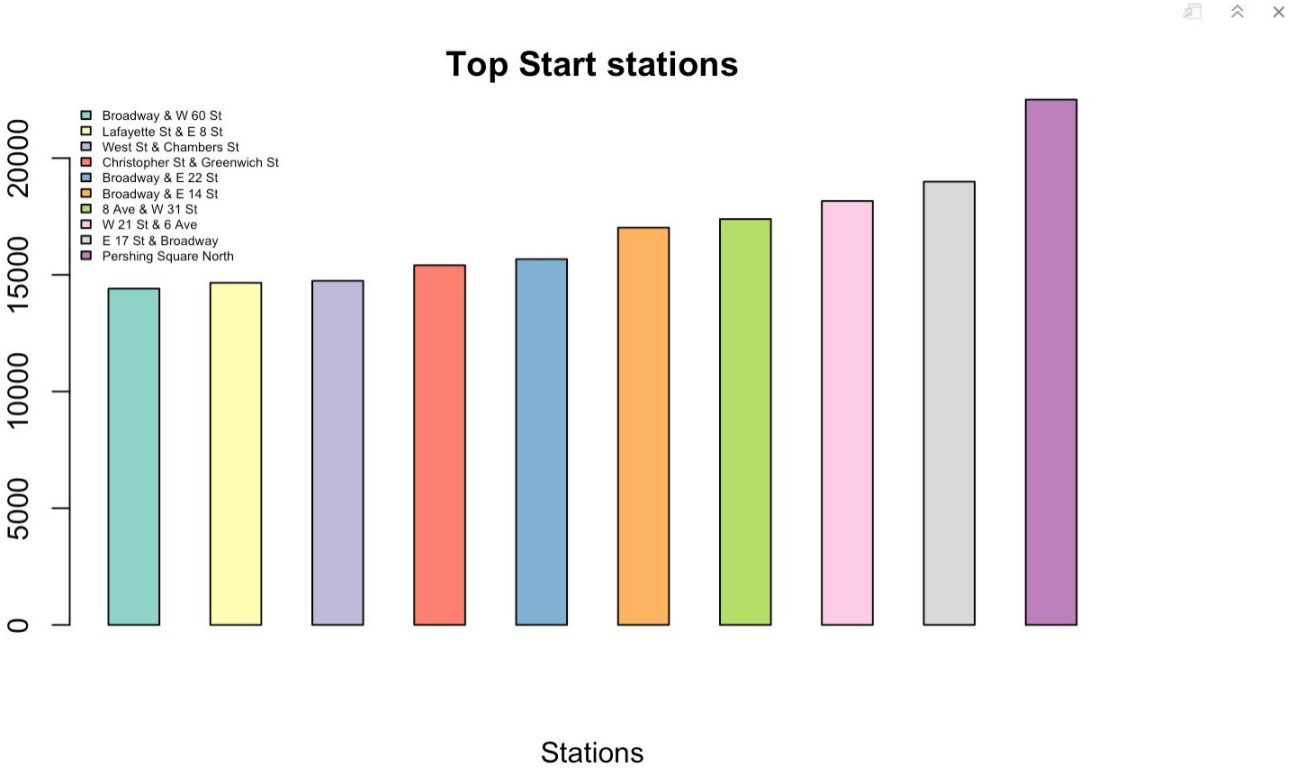
```r
ggplot(augDecData, aes(x= weekday, fill= gender)) + geom_bar( stat = "count" , na.rm= T,
                                                              position = position_dodge(width = 0.25)) +
labs(
  title= "Who bikes more in the week?",
  x= "Week day", y= ""
)
```



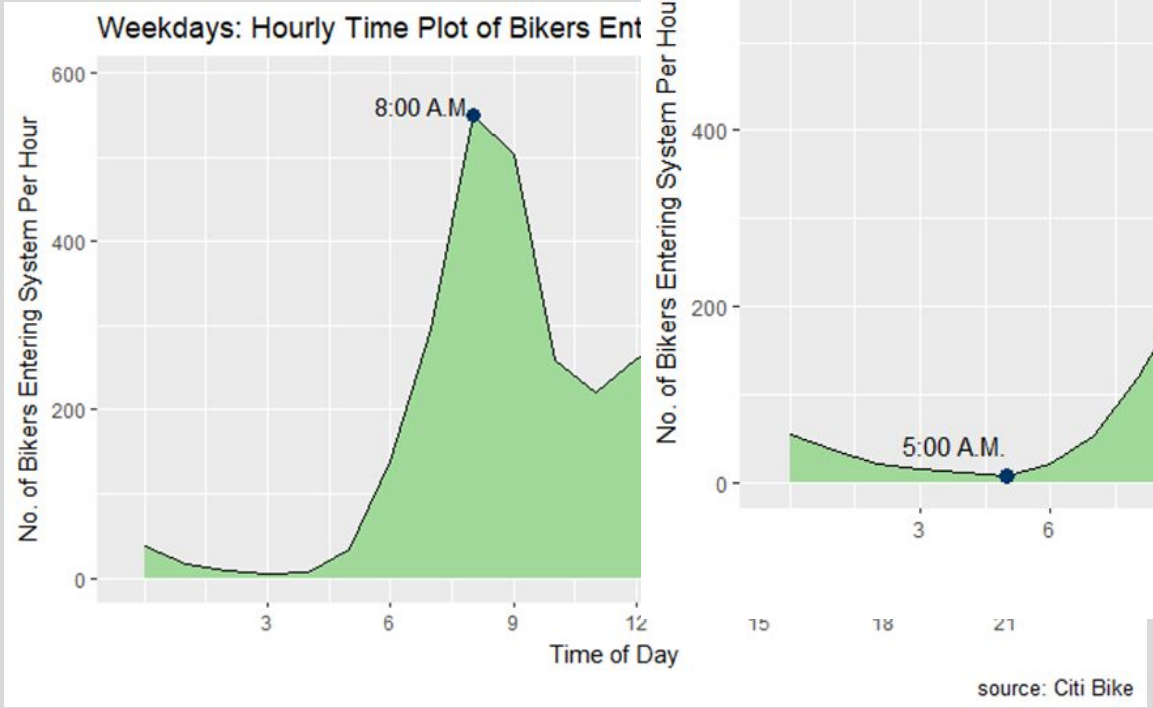
Popular stations

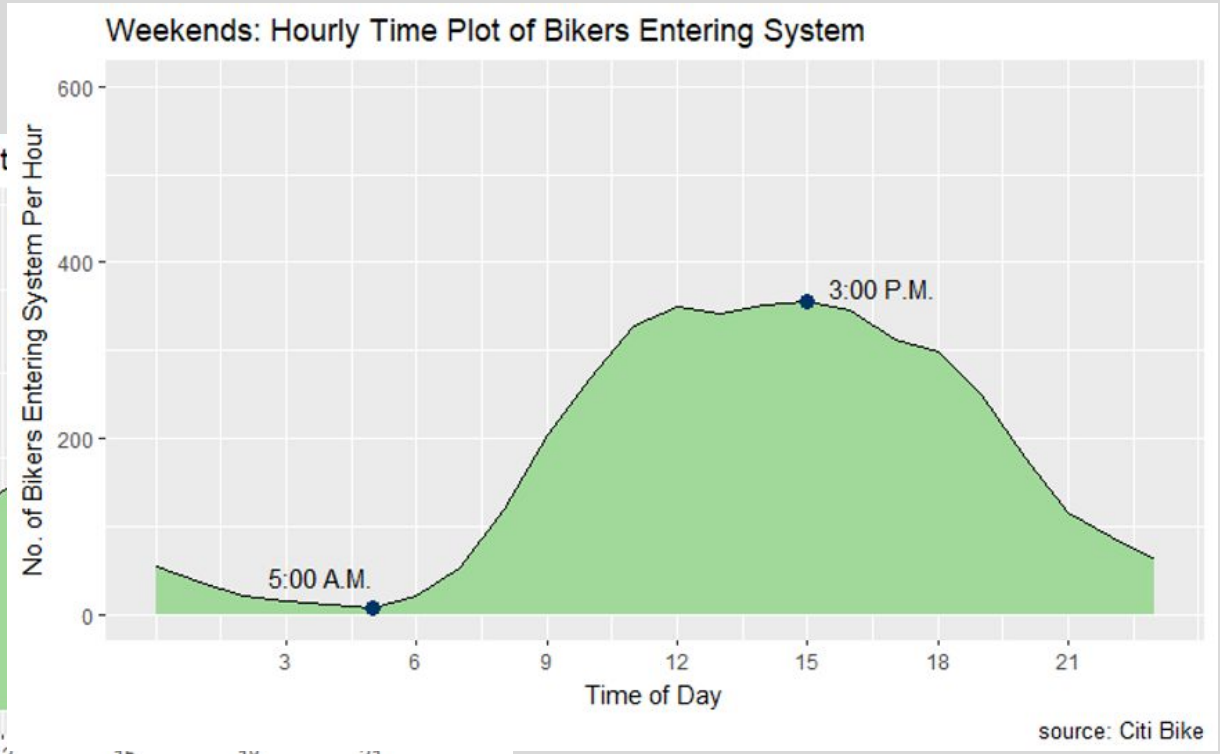
```
with(diamonds, barplot(rev(sort(table(augDecData$start.station.name), decreasing = T )[1:10]), width = 10,
main= "Top Start stations", space= 1, xlab = "Stations", legend.text = T, col= colBar, names.arg = F,
args.legend= list(x = "topleft", bty = "n", cex= 0.45)))
```

Pershing Square North



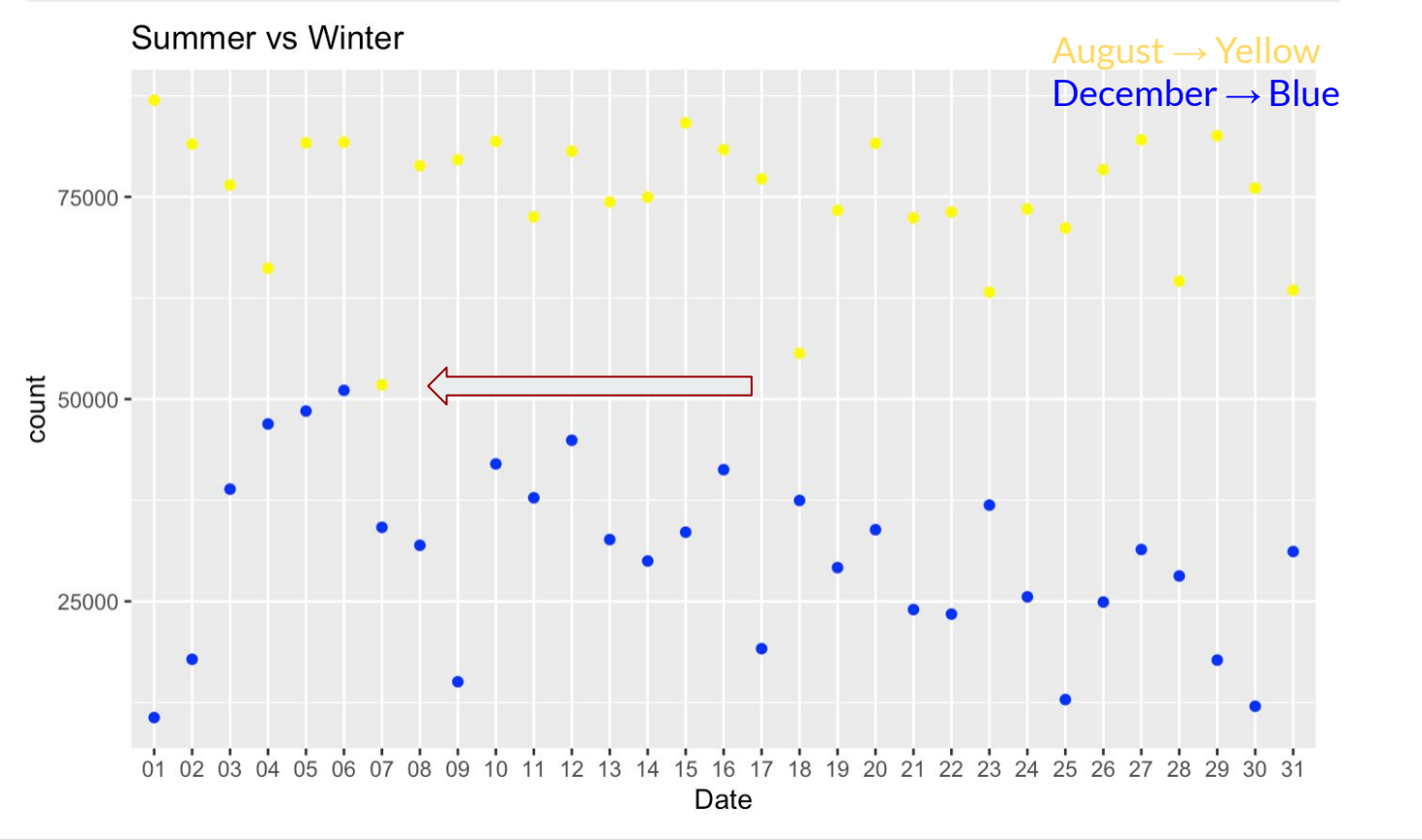
The busiest hour of the day





Comparing August 2019 vs December 2019

```
ggplot() +
geom_point(data=aug , aes(x=aug$day), color='yellow', stat = "count", na.rm= T) +
geom_point(data= dec, aes(x=dec$day), color='blue', stat = "count", na.rm= T) +
```



March-June 2019 vs March-June 2020

Average trip dur
2019:

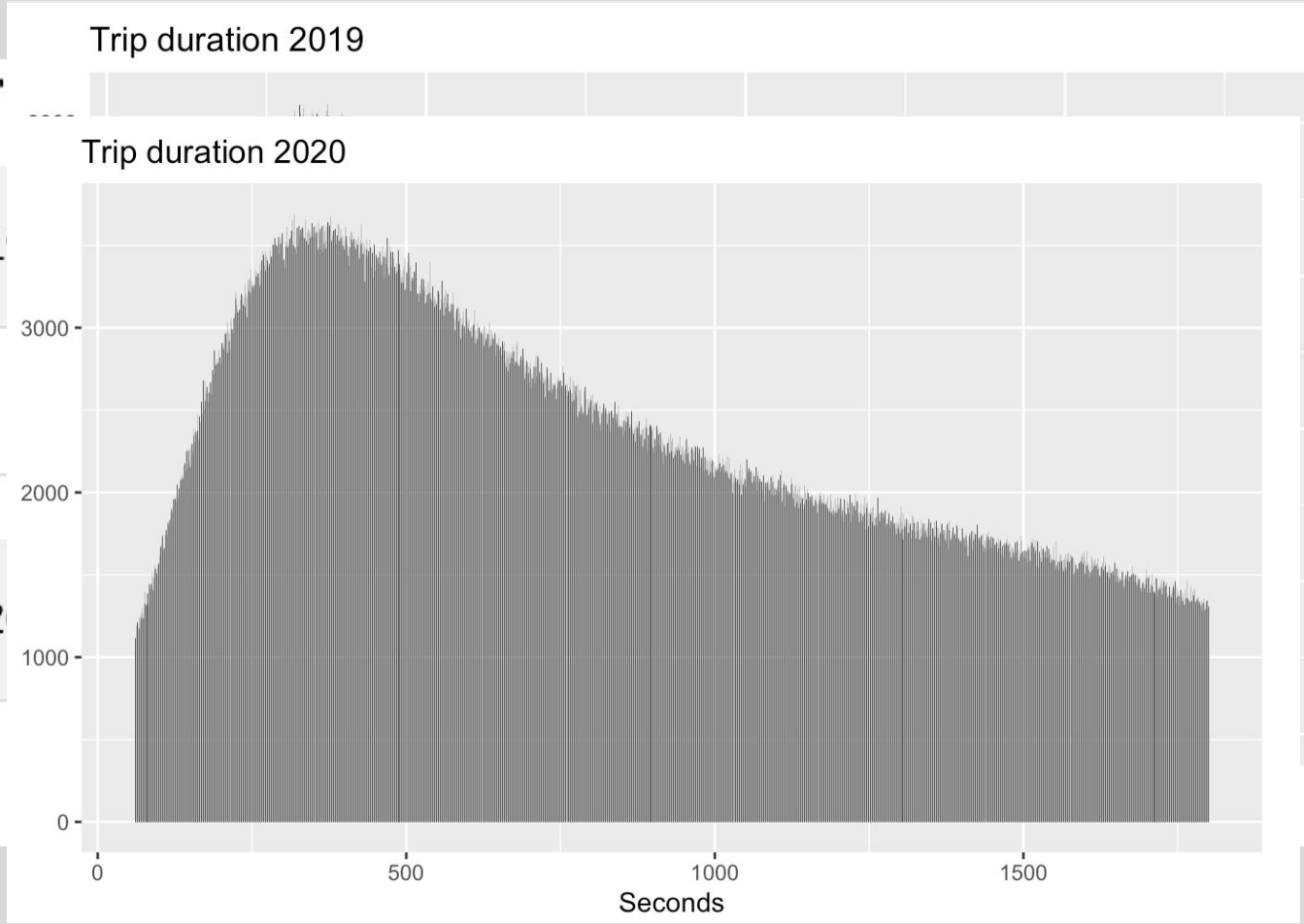
```{r}
mean( marchJune1
```

[1] 11.3472

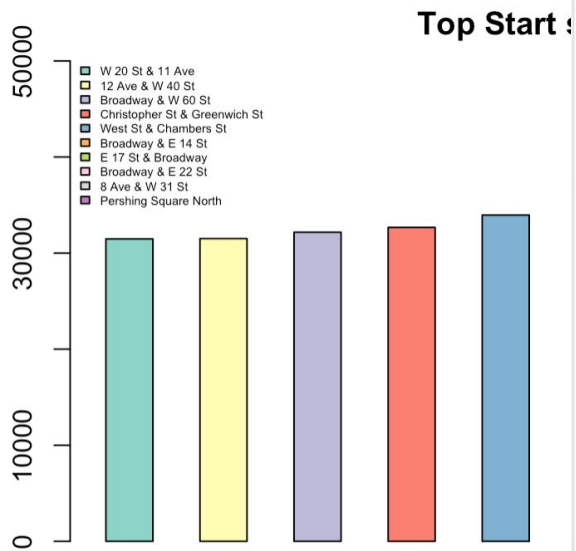
2020:

```{r}
mean( marchJune2
```

[1] 13.57526

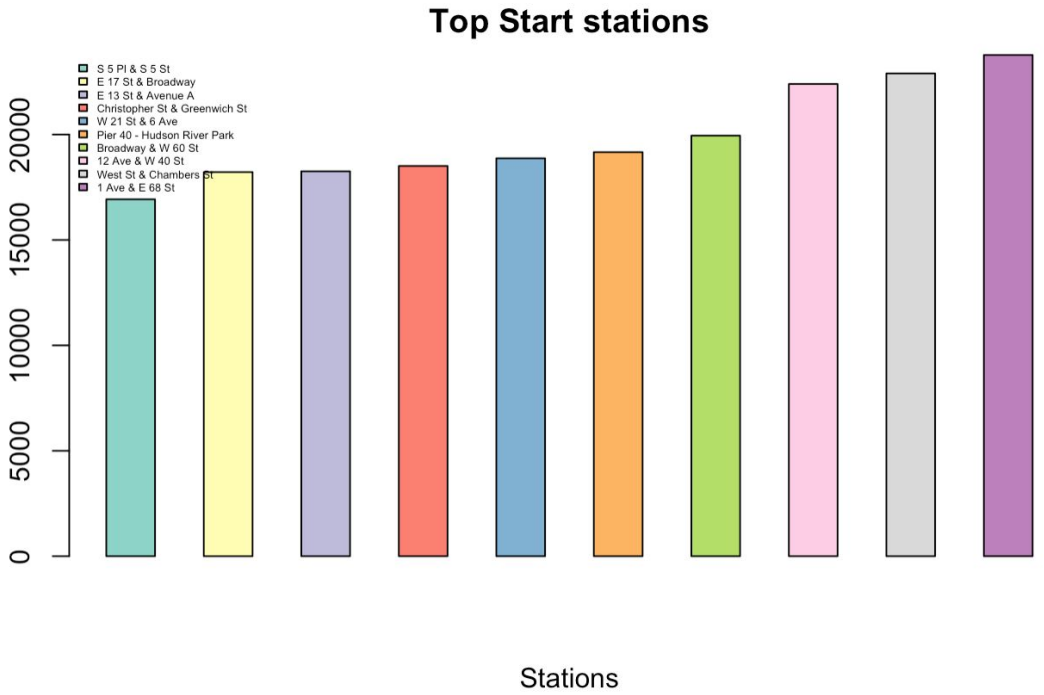


2019 Popular Stations

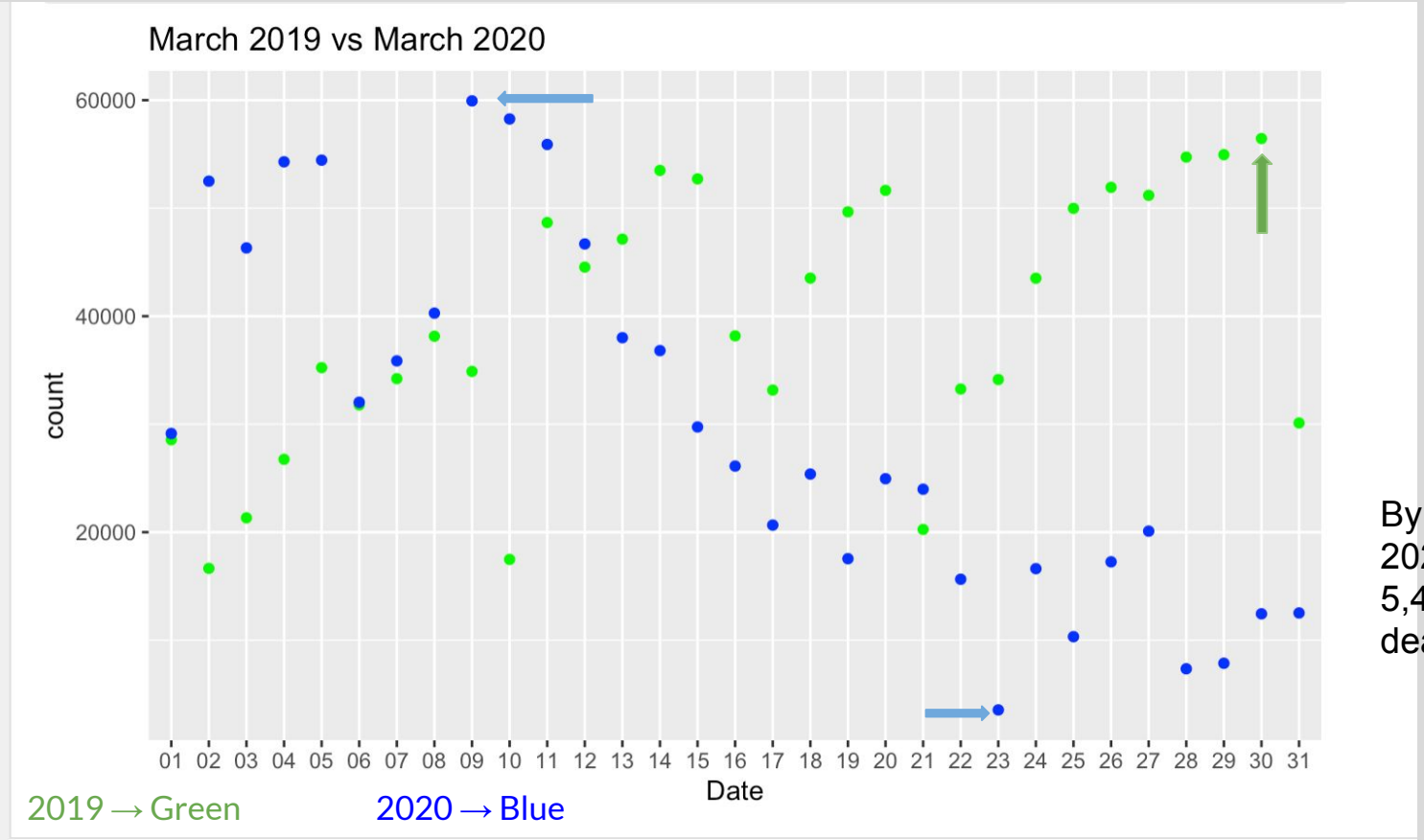


Pershing Square North

2020 Popular Stations

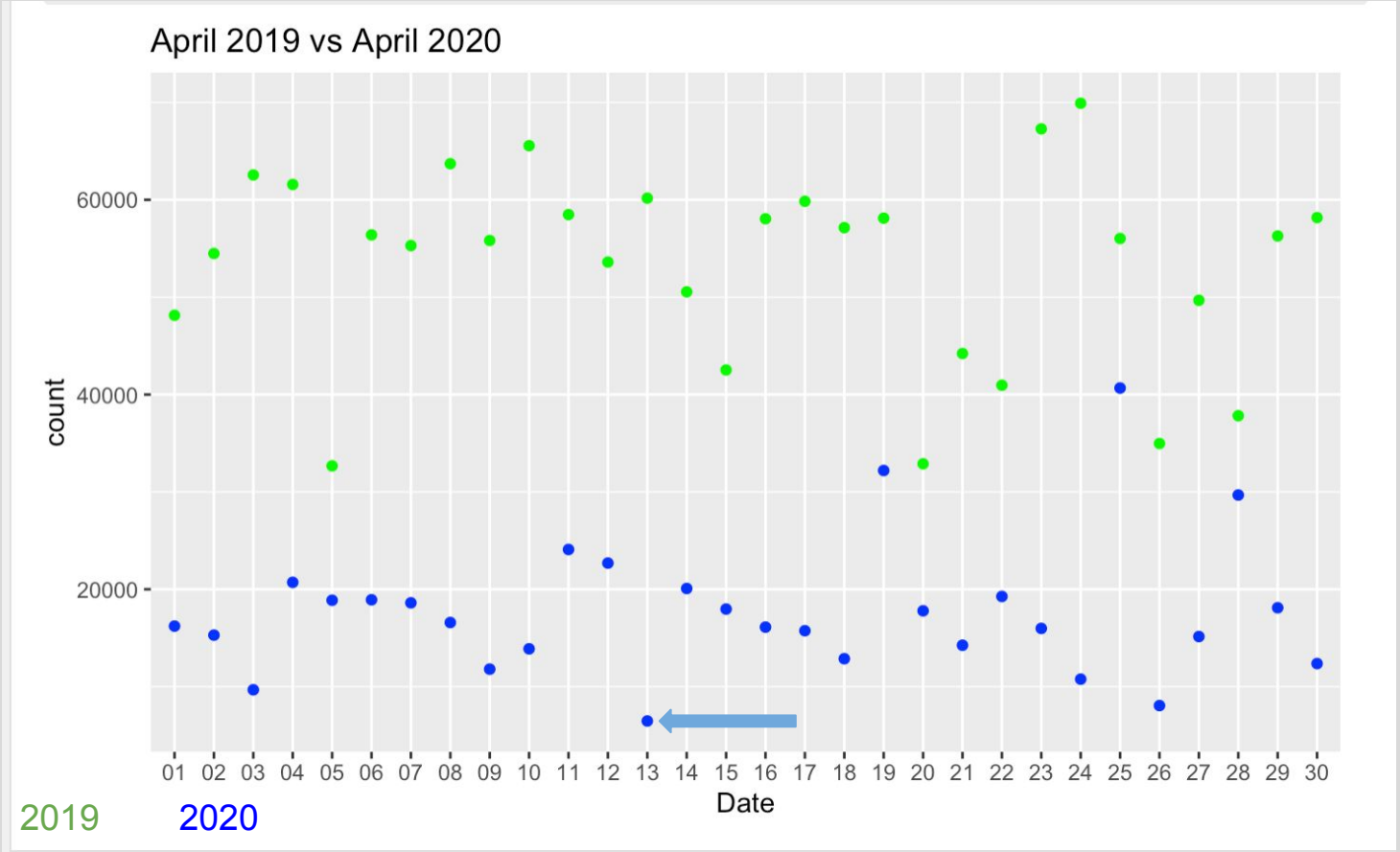


1 Ave & E 68St



By the end of March 2020, there were 5,473 cases and 384 deaths.

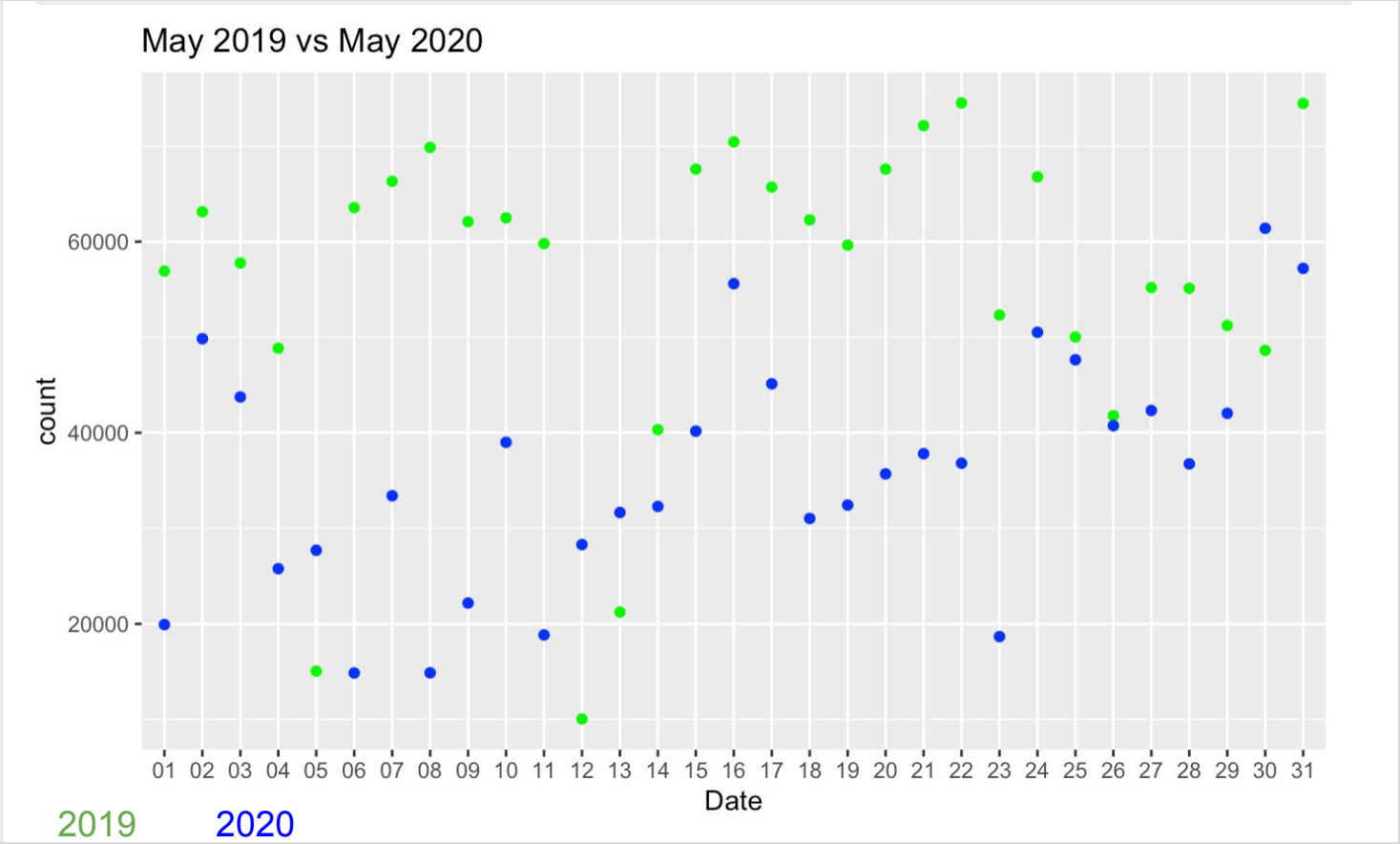
Source: https://www1.nyc.gov/site/doh/covid/covid-19-data.page



April 25th= 1,595 cases and 265 deaths

April 30th= 2,024 cases and 222 deaths

Source: https://www1.nyc.gov/site/doh/covid/covid-19-data.page

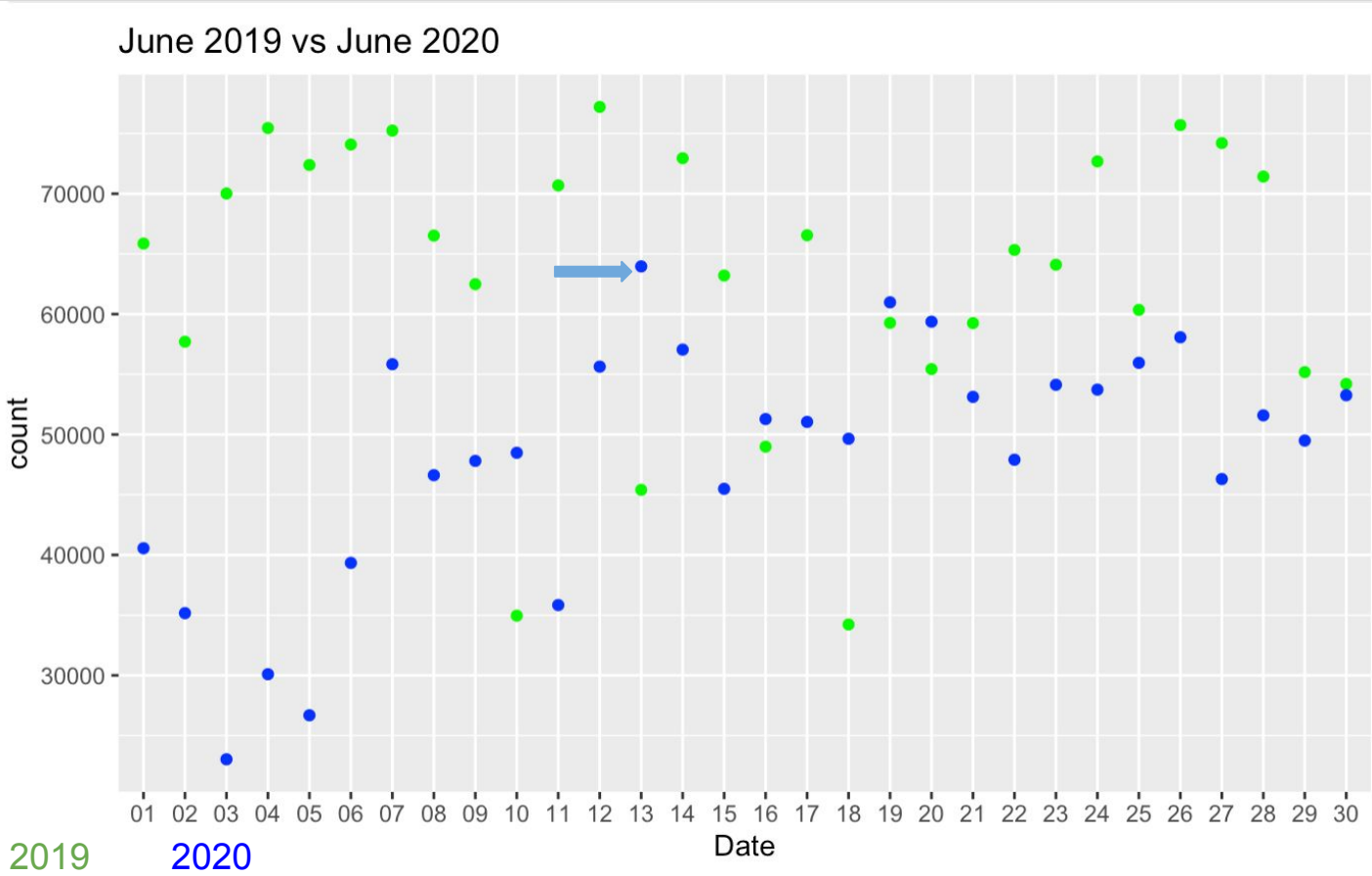


May 16th= 488 cases, 83 deaths

May 30th= 349 cases, 54 deaths

May 31st= 217 cases, 39 deaths

Source: https://www1.nyc.gov/site/doh/covid/covid-19-data.page



June 13th= 196 cases, 14 deaths

June 30th= 443 cases, 24 deaths.

Source: https://www1.nyc.gov/site/doh/covid/covid-19-data.page

GREEN AND YELLOW CAB ANALYSIS

Background

- Two varieties of taxicabs in NYC
  - Yellow “medallion taxis” and green “Boro taxis”
    - Green Taxis introduced in 2013 to serve boroughs other than Manhattan
- All NYC Taxis operated by private companies, but are licensed by the New York City Taxi and Limousine Commission (TLC).
- In 2019 there were 2,568,296 yellow and 185,400 green taxi trips

“Taxi and Ridehailing App Usage in New York City.” Toddwschneider.com, toddwschneider.com/dashboards/nyc-taxi-ridehailing-uber-lyft-data/.

Green and Yellow Cab Data

The data was obtained at TLC's website:

https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Cleaning and filtering of data

```
yellowData01 <- read.csv("yellow_tripdata_2019-01.csv")
yellowData01 <- yellowData01%>%na.omit()
yellowData01 <- filter(yellowData01, VendorID <= 2, passenger_count >= 1, trip_distance >= 0.50, trip_distance <= 300.0, Rat
ecodeID >= 1 & RatecodeID <= 6, fare_amount >= 0, extra >= 0, mta_tax >= 0, tip_amount >= 0, tolls_amount >= 0, improvement_
surcharge >= 0, total_amount >= 0, congestion_surcharge >= 0)
yellowData01 <- yellowData01[sample(nrow(yellowData01), 10000),]
```

"Na.omit" - omits all observations with an NA value. I.e. observations with incomplete data.

Data is then filtered to a certain degree of reasonability.

```
greenData01 <- read.csv("green_tripdata_2019-01.csv")
greenData01 <- greenData01[-15]
greenData01 <- greenData01%>%na.omit()
greenData01 <- filter(greenData01, VendorID <= 2, passenger_count >= 1, trip_distance >= 0.50, trip_distance <= 300.0, Ratec
odeID >= 1 & RatecodeID <= 6, fare_amount >= 0, extra >= 0, mta_tax >= 0, tip_amount >= 0, tolls_amount >= 0, improvement_su
rcharge >= 0, total_amount >= 0, congestion_surcharge >= 0)
greenData01 <- greenData01[sample(nrow(greenData01), 10000),]
```

Repeat code for the remaining 11 months of Data.

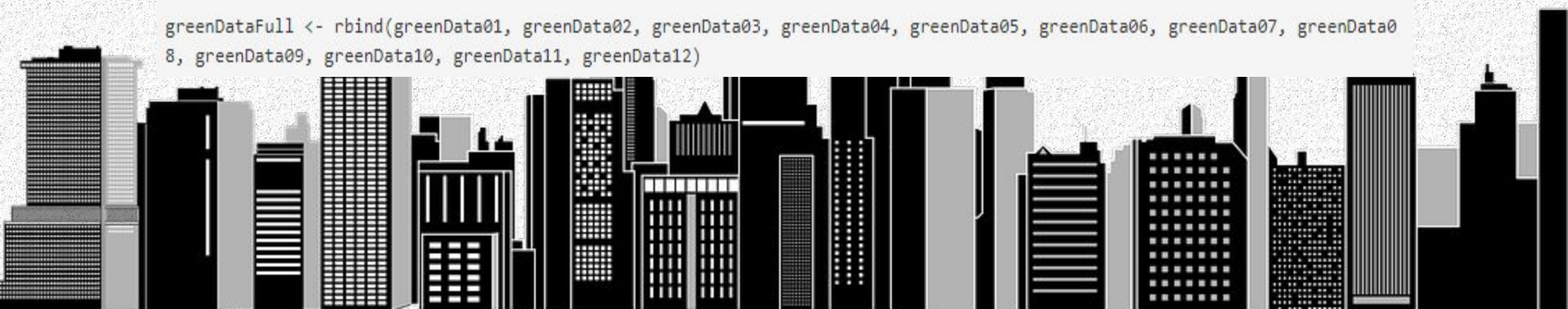
Taking a sample size and Combining the data

This line of code, from the previous chunk extracts 10,000 samples at random from one month's worth of data to be our sample population.

```
yellowData01 <- yellowData01[sample(nrow(yellowData01), 10000),]
```

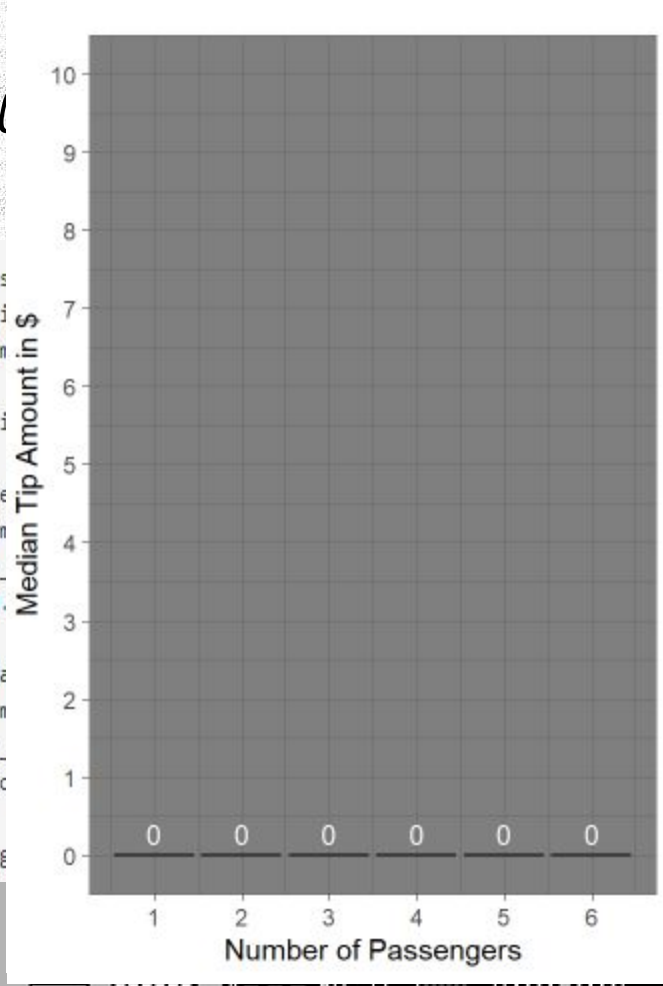
The R-bind function is then used to combine all the sample sizes into one data frame for us to work with and analyze.

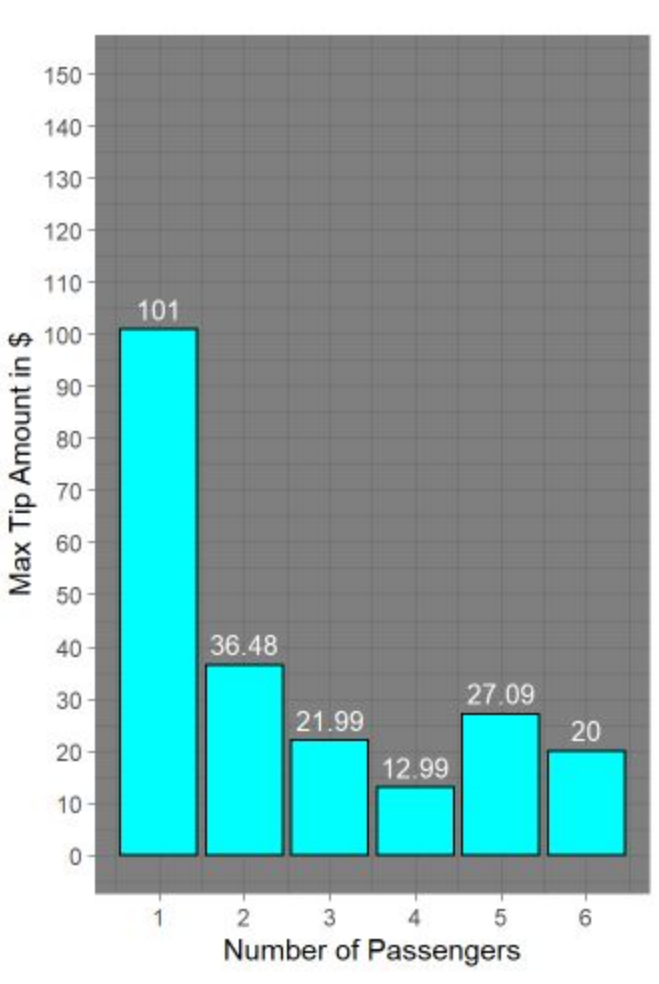
```
yellowDataFull <- rbind(yellowData01, yellowData02, yellowData03, yellowData04, yellowData05, yellowData06, yellowData07, ye
llowData08, yellowData09, yellowData10, yellowData11, yellowData12)

greenDataFull <- rbind(greenData01, greenData02, greenData03, greenData04, greenData05, greenData06, greenData07, greenData0
8, greenData09, greenData10, greenData11, greenData12)
```



Fi… …of

```
numOfPas
medianTi
maxTipAm

yellowti

yellowMe
    geom
+ scale_
ust= -0.

yellowMa
    geom
scale_y_
-0.5, co

ggarrang
```

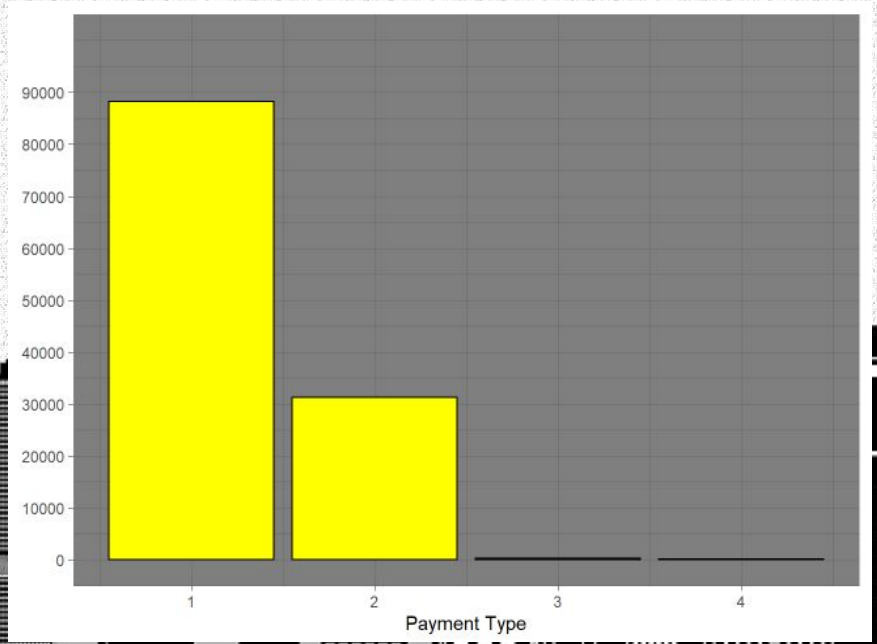


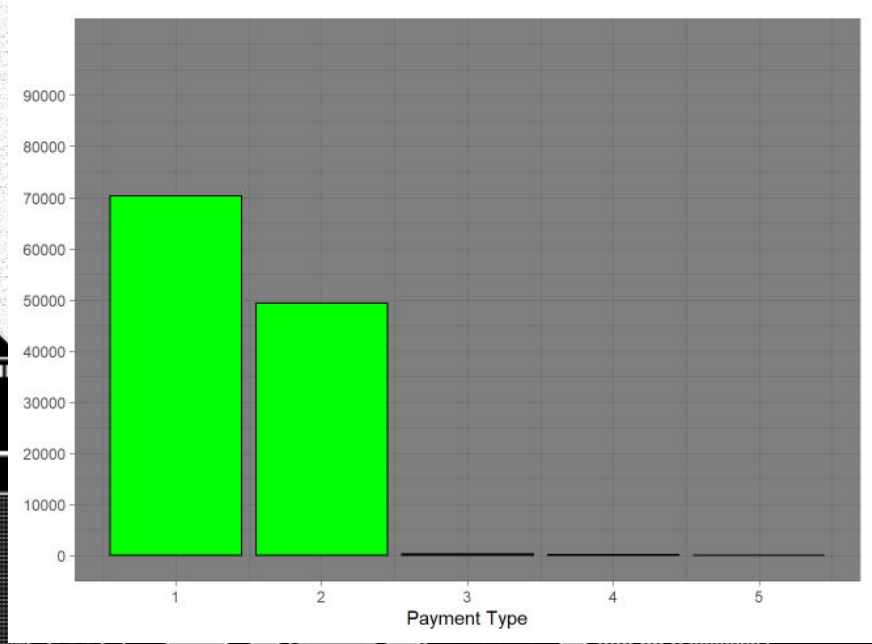



```
= 1:6)
nt), vj

1:6) +
vjust=
```

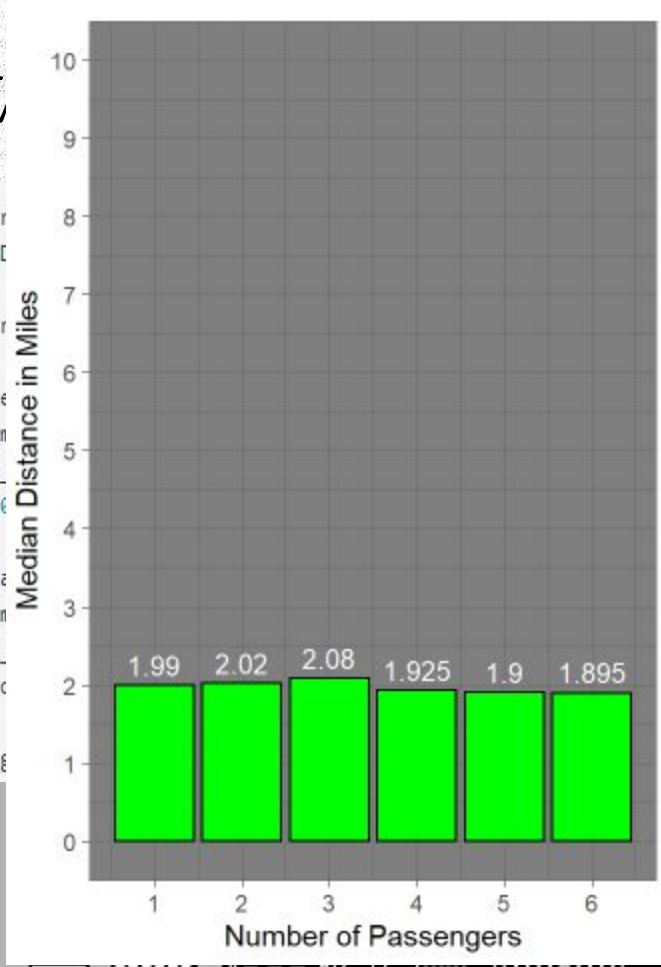
Payment_type	A numeric code signifying how the passenger paid for the trip. **1= Credit card** **2= Cash** **3= No charge** **4= Dispute** **5= Unknown** **6= Voided trip**

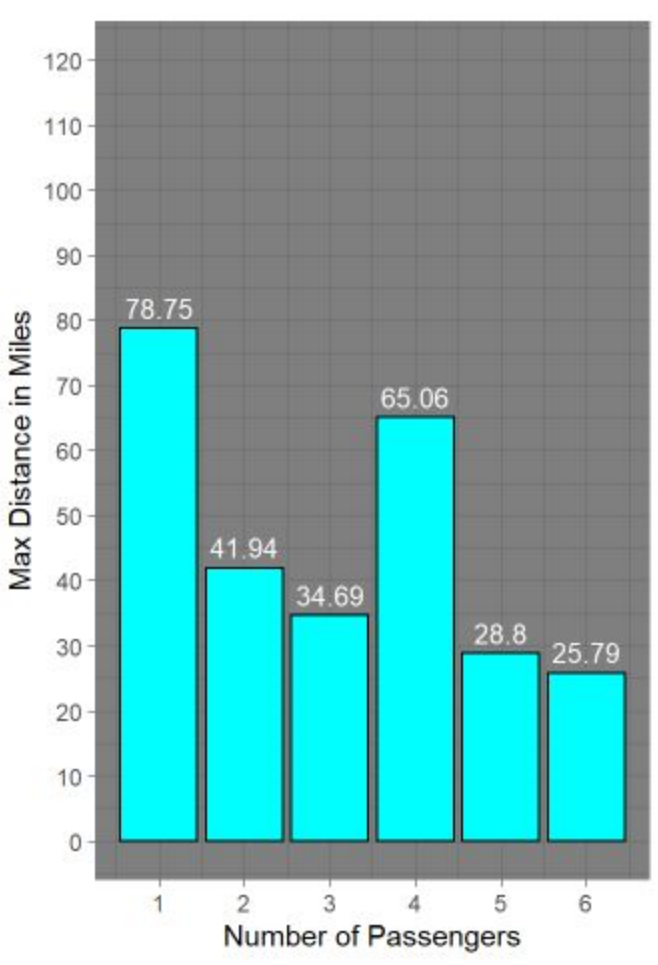
```
ggplot(yel
        ge                                                    ), limits
= c(0,100000)) + theme_dark() + xlab("Payment Type")
```

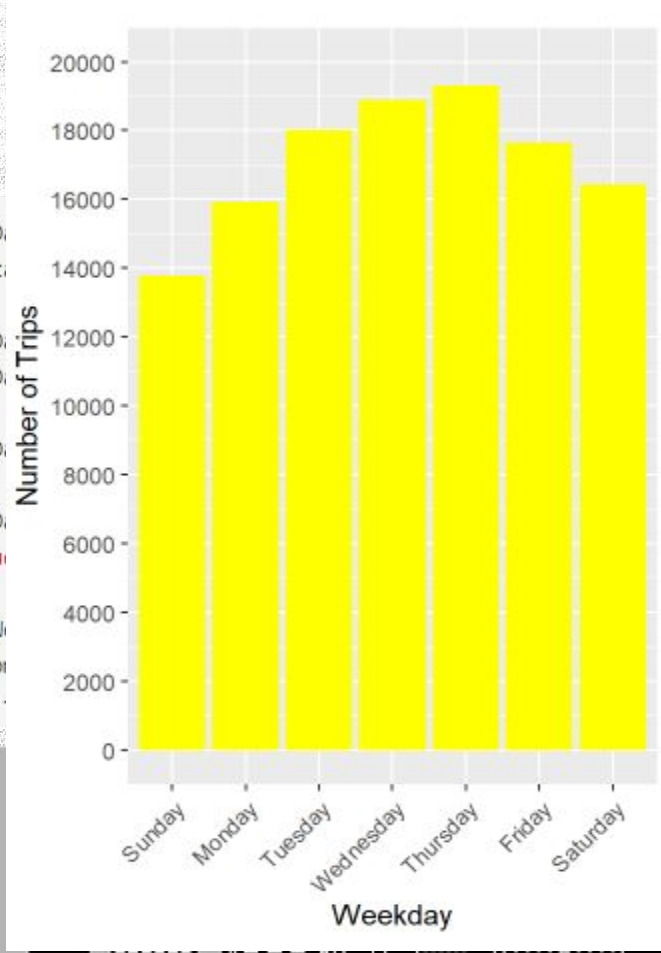


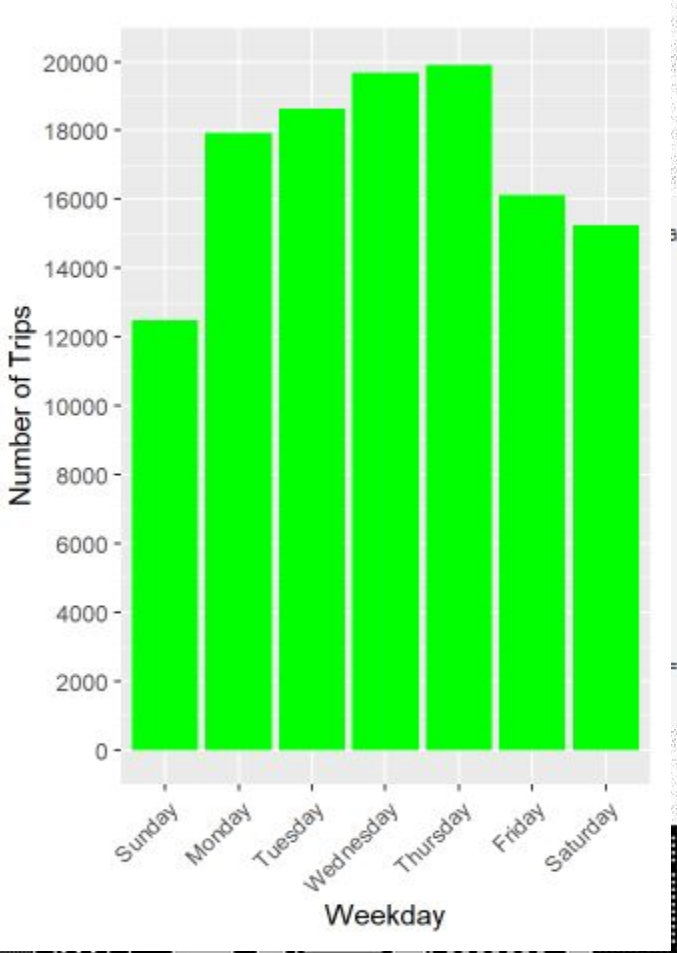


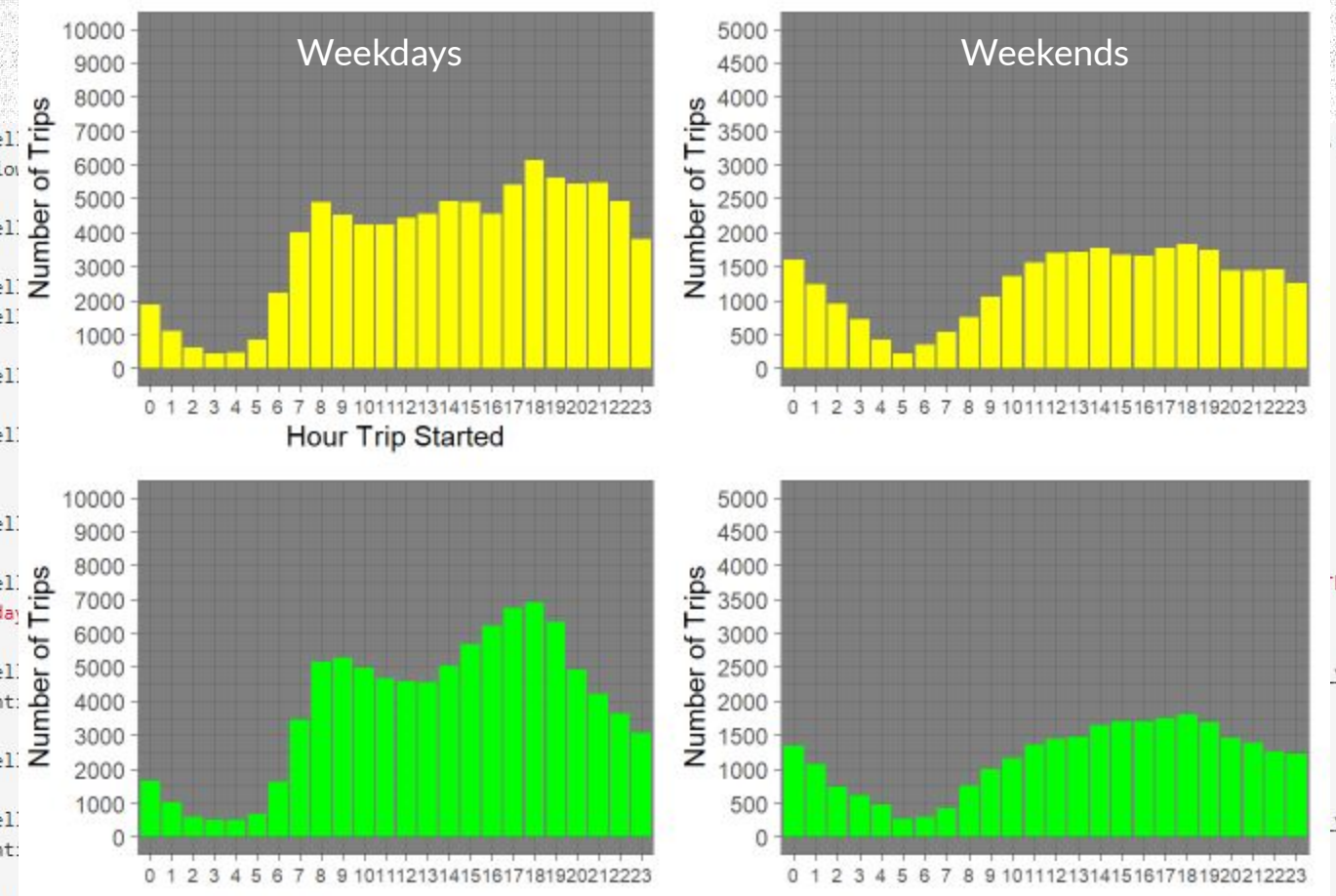
MEDI NGERS


Median Distance in Miles
0
1
2
3
4
5
6
7
8
9
10
1.99
2.02
2.08
1.925
1.9
1.895
1
2
3
4
5
6
Number of Passengers

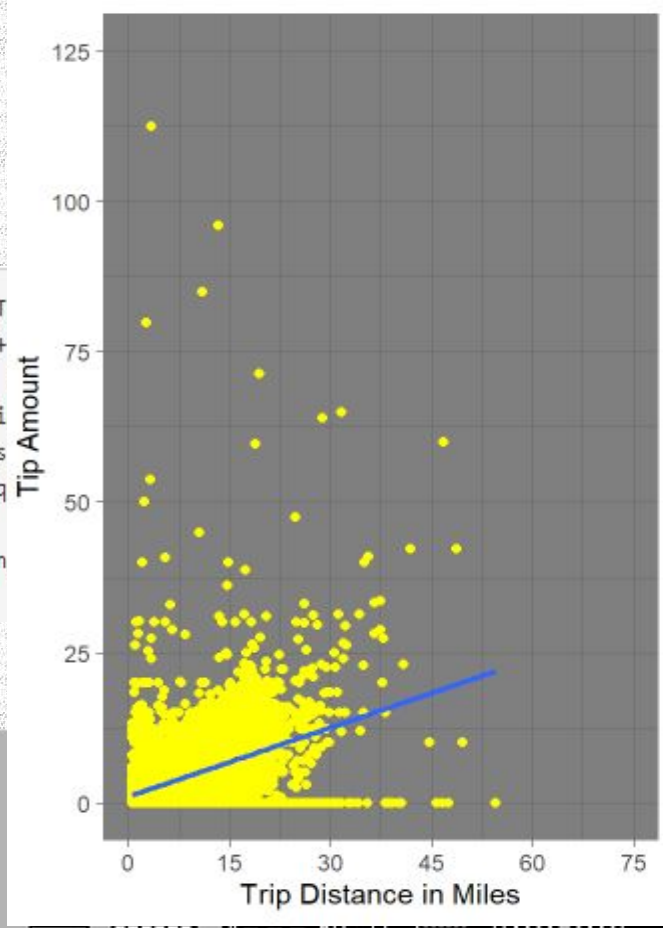

Max Distance in Miles
0
10
20
30
40
50
60
70
80
90
100
110
120
78.75
41.94
34.69
65.06
28.8
25.79
1
2
3
4
5
6
Number of Passengers

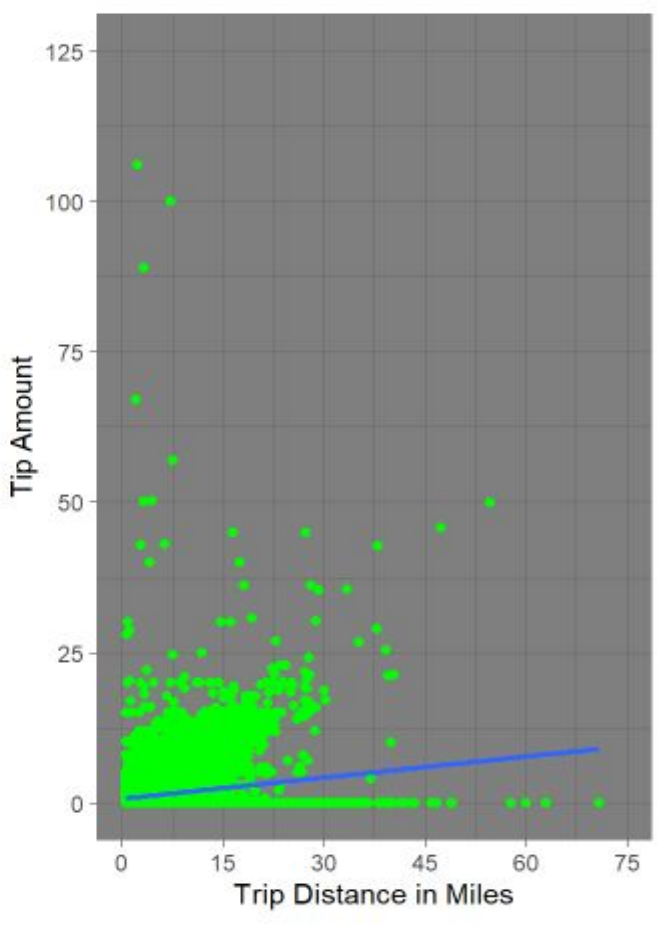

20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Number of Trips
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Weekday


20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Number of Trips
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Weekday




```
yellowHourEnd <- yellowHourEnd + theme(axis.text.x = element_text(size = 7))
```


125
100
75
50
25
0
Tip Amount
0
15
30
45
60
75
Trip Distance in Miles


125
100
75
50
25
0
Tip Amount
0
15
30
45
60
75
Trip Distance in Miles