New York City Parking Violation Analysis

4 Boroughs- Leela Sai Mounika Bandaru, Shruti Chandra, Nishith Varma Penumatsa, Sai Jai Krishna Ankith Srungarapu

25 April 2021

## 1.Introduction

Parking violations are a continual presence and source of annoyance in the lives of many residents of large modern cities, especially New York city. They can also be an important source of revenue for cities in the 2017 fiscal year. Parking disorderly not only brings inconvenience to traffic flow but also has a negative impact on the image of the city.

Especially with on-street parking relatively difficult to find in certain districts in the City, tickets can make commuting or performing errands by car particularly burdensome, restricting the hours and locations that parking is available; New York Cites weekly alternate side parking policies for street cleaning sometimes lead apartment building doormen to temporarily double-park groups of residents’ cars on the opposite side of the street to avoid tickets. They can also be an important source of revenue for cities.

## Research Questions:

Our research is to evaluate several factors that are associated with the violations that happened in New York City parking ticket data and to determine if parking violations are more likely to occur on specific times, to identify which precincts has a greater number of violations.

## Why this dataset?

From our analysis on this data, we would produce improvements for pedestrians, vehicular safety through education and a timely decision of parking, and traffic violations. Further,it would help the government to resolve violation issues in a systematic manner. We have used RStudio, spark to analyze the statistical data of the violation rate and visualize on various aspects to give the insight of parking violations.

## Dataset Information

The first data set used in the analysis consists all set of 10M parking ticket violations issued in New York City in 2017; each observation comprises one ticket issued. The other data set consists of 63M records and 19 Variables, this data file contains open parking and camera violations issued by the city of New York for different states. The NYC department of finance collects data on every parking ticket issued in NYC (~10M per year!). The Department of Finance is responsible for collecting and processing payments for all parking tickets and camera violations. This data set gives an insight on ticket resolution and to guide policymakers. ([NYC parking tickets](https://www.kaggle.com/new-york-city/nyc-parking-tickets)) ([Open parking and camera violations](https://data.cityofnewyork.us/City-Government/Open-Parking-and-Camera-Violations/nc67))

### Libraries

Below are all the libraries and data frames used for the project. Code for lower casing the column names has also been shown. We used inner join to join a new table to the original data set containing violation charges and description. We have also converted the df\_join dataframe into a tibble as it was required for converting column to row in the following code for clustering.

library(tidyverse)  
library(RColorBrewer)  
library(scales)

### Importing datasets

pvi <- read\_csv("Parking\_Violations\_Issued\_2017.csv")

## 2.Body

Working with the large data set regardless of the software can cause problems during the execution. So we are considering the sample of data set.we have applied random sampling to our datasets.

pvi\_sd <- pvi %>% sample\_n(500001, replace = FALSE) #Sampling the data

## Data Tidying

We had the last column in our open parking and camera violation table that contained the image of the summon issued to every person. Also, there are other columns like meter number,Issuer squad, sub division etc and the images of the summon are not going to be of any use in the analysis we decided to get rid of it and that was the only consolidation we did.

## Removing columns that are not required  
pvi\_sd <- select(pvi\_sd,1:9,15,17,22,25,34,36,40,16,20)

we are changing the column names in the original dataset to lowercase as the usage of column names will be easy during analysis.

pvi\_sd <- pvi\_sd %>% select\_all(~gsub("\\s+|\\.", "\_", .)) %>% select\_all(tolower)

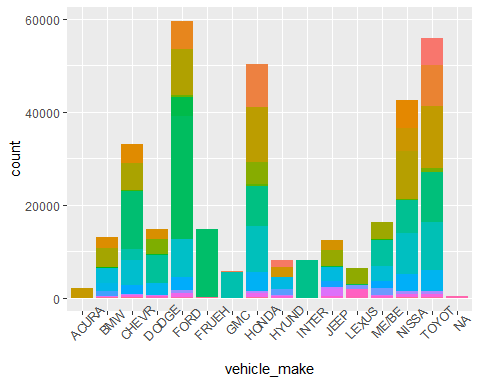
We want to separate issue date column into individual as the analysis we want to perform based on the day of violation issued.

pvi\_sd<- separate(pvi\_sd, issue\_date, c("month", "day","year"))# split issue\_date into MDY format  
pvi\_sd$month <- as.integer(pvi\_sd$month)  
pvi\_sd$day <- as.integer(pvi\_sd$day)  
pvi\_sd$year <- as.integer(pvi\_sd$year)

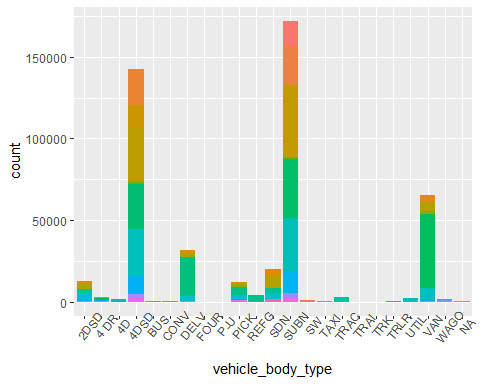
## Visualizations

We have created visualizations for the variables in the dataset that we are considering that will impact parking violations in NYC.Now, we want to know how the following variables ticket issued day,year,vehicle body type and vehicle make variables impact parking violation and then vioaltions by violation counties and their impact.

## Number of parking Violations Issued by Vehicle Make  
by\_vehiclemake <- pvi\_sd %>% group\_by(violation\_county,vehicle\_make) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 slice(1:10)   
ggplot(data = by\_vehiclemake, aes(x = vehicle\_make, y = count, fill = vehicle\_make)) +  
 geom\_histogram(stat = "identity") +  
 ggtitle("Parking Violations Issued by Vehicle Make") +  
 theme(axis.text.x = element\_text(angle = 45))



## Observation From the chart, we can observe that the Vehicle make “Ford” has the most number of tickets issued with almost 50k tickets while Acura has the least number of tickets issued within the range of hundred tickets.

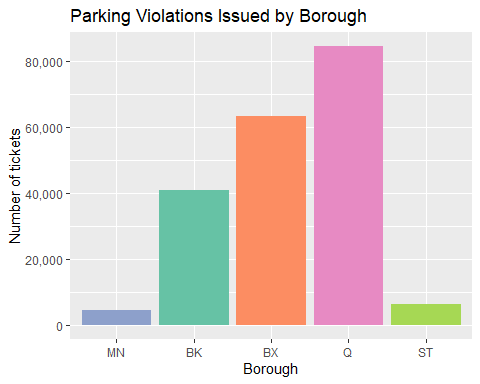
## Number of parking Violations Issued by Vehicle Body type  
by\_vehiclebodytype <- pvi\_sd %>% group\_by(violation\_county,vehicle\_body\_type) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 slice(1:10)   
ggplot(data = by\_vehiclebodytype, aes(x = vehicle\_body\_type, y = count, fill = vehicle\_body\_type)) +  
geom\_histogram(stat = "identity") +  
ggtitle("violations by vehicle body") +  
 theme(axis.text.x = element\_text(angle = 50))

## Observation

From the chart, we can observe that the Vehicle body type “SUBN” has the most number of tickets issued with almost 50k tickets while Acura has the least number of tickets issued within the range of hundred tickets.

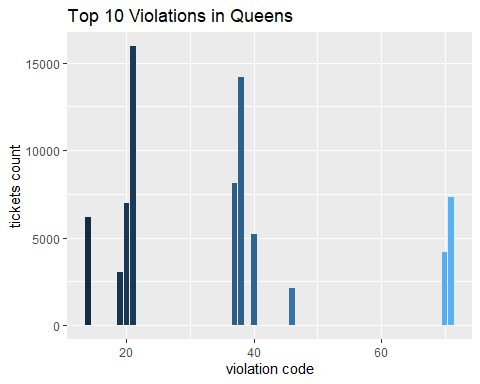
In the following analysis, we are considering only Boroughs of NYC which means the main 5 cities in NYC such as Bronx (BX), Manhattan (MN),Queens(Q),Staten Island(SI),Brooklyn(BK) as they are the busiest cities. Now lets find out which city has highest number of violations.

## parking Violations by Borough  
ggplot(pvi\_sd,aes(x=violation\_county,fill=violation\_county)) +   
geom\_bar() +   
ggtitle('Parking Violations Issued by Borough') +  
xlab('Borough') + ylab('Number of tickets') +  
scale\_x\_discrete(limits=c("MN","BK","BX","Q","ST"))+  
scale\_y\_continuous(labels=comma)+  
scale\_fill\_brewer(palette = "Set2")+  
theme(legend.position = 'none')

 ## Observation From the output, we can see that “Queens” city has the most number of violations compared to other cities followed by Bronx.Also the least number of violations in Manhattan.The output might be surprising because Manhattan got the least as we can say people there use public transport more than own cars.

As we observed from the above output, “Queens” has more number of violations, lets get into deep observation and find out what are the most violations(there are various violation codes upto 99 in the data set , so considering top 10 would be more visually appealing)issued in Queens.

## Top 10 violation codes issued in Queens  
Queens <- pvi\_sd %>% filter(violation\_county=="Q")  
Top10Violations <- group\_by(Queens,violation\_code) %>%  
 summarise(count = n()) %>%  
 arrange(desc(count)) %>%  
 slice(1:10)   
ggplot(Top10Violations,aes(x=violation\_code,y=count,fill=violation\_code)) +   
 geom\_bar(stat='sum')+  
 ggtitle(paste0('Top 10 Violations in Queens'))+  
 xlab('violation code') +   
 ylab('tickets count') +  
 theme(legend.position = 'none')

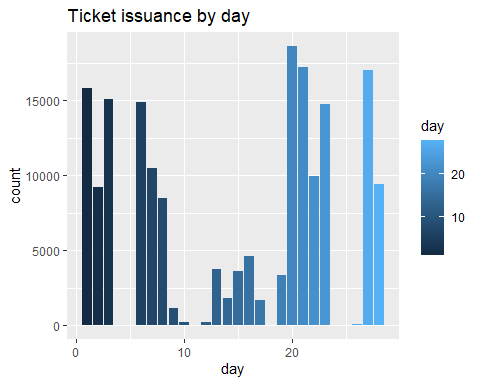


## Observation

From the output we can say that the most violation code issued in Queens was code 20 (NO PARKING-DAY/TIME LIMITS) and code 40 (FIRE HYDRANT).

we found out what violation codes most occurred and now lets figure out which months the tickets are issued most in Queens.

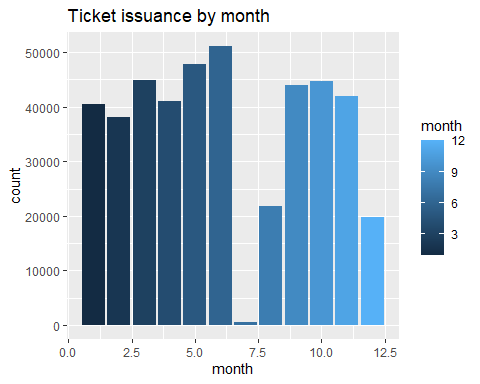
## Number of parking Violations based on day in Queens  
by\_dates <- pvi\_sd %>%   
 group\_by(violation\_county,day) %>%  
 summarise(count= n()) %>%  
 arrange(desc(count)) %>%   
 slice(1:10)  
ggplot(data = by\_dates, aes(x = day, y = count,   
 fill = day)) +  
 geom\_bar(stat = "identity") +  
 ggtitle("Ticket issuance by day") +  
 theme\_grey()



## Observation

From the output we can say that at the violations are more initially at first weekend and then reduced and again increased in the middle. This can explain us there is lot of fluctuation in the tickets issued on each and every day.

## Number of parking Violations based on month in Queens  
by\_dates <- pvi\_sd %>%   
 group\_by(violation\_county,month) %>%  
 summarise(count= n()) %>%  
 arrange(desc(count)) %>%   
 slice(1:10)  
ggplot(data = by\_dates, aes(x = month, y = count,   
 fill = month)) +  
 geom\_bar(stat = "identity") +  
 ggtitle("Ticket issuance by month") +  
 theme\_grey()



## Observation

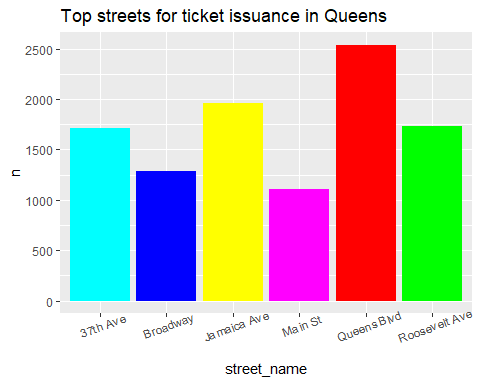
From the output we can say that for the months May and June tickets issued are around 50k and the least tickets issued is in July which is around 200.This make sense because in summer people try to be outside rather than in February.

## Most number of parking violations grouped by streets in Queens county. we are just focusing on Queens county because initially that city got the most parking violations and we are trying

## Observation

From the Bar Chart, we can observe that Queens Blvd had highest violations recorded with 2500 tickets within Queens and Steinway street is the least violations with around 900 tickets.

by\_street <- count(Queens,street\_name) %>%   
 arrange(desc(n))  
## Top streets in Queens county with most number of parking violations  
by\_street\_head <- head(by\_street,6)  
## Visualizing streets in Queens county  
ggplot(by\_street\_head,aes(street\_name,n))+  
ggtitle('Top streets for ticket issuance in Queens')+  
geom\_bar(stat = "identity", fill = rainbow(n=length(by\_street\_head$n)))+  
theme(axis.text.x = element\_text(angle = 20))



**Observation**

From the Bar Chart, we can observe that Queens Blvd had highest violations recorded with 2500 tickets within Queens and Steinway street is the least violations with around 900 tickets.

Determining most frequent Parking Violation at specific times (AM/PM)

Created a variable, to find out the top 3 violation codes that have occurred most frequently. Used group by for grouping violation codes, and then summarize for counting the occurrences, arrange for descending order of display.

The Top 3 most frequently occurring Violation Codes were 21, 36 and 38. VC 21 occurred 70,656 times, VC 36 occurred 64944 times, VC 38 occurred 49102 times.

Below, 3 columns were created using Mutate function in a dataframe df\_VC. Str\_sub was used to return strings from 1 to 4 and the last string from Violation Time column. They were stored in Two separate columns for the Hours and AM/PM i.e. ViolationTime and TimeRange. And, a new column for assigning a TimeRange to the specific violation times as they were given in HHMM format, so that they can be shown clearly on the plot.Then, they were grouped by Violation Code, ViolationTime and TimeRange, counted by Violation Code using Summarise, arranged in descending order, and filtered by the top 3 Violation Codes for making the Bar Plot.

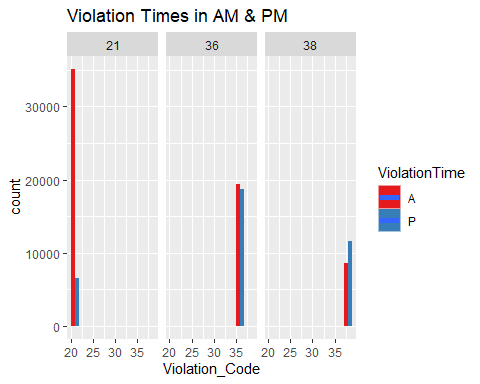
df\_VC<-PVI\_Final %>%   
 mutate(PVI\_Final, ViolationTime=str\_sub(`Violation\_Time`, -1), ViolationHours=str\_sub(`Violation\_Time`, 1, 4),   
 TimeRange=case\_when(  
 between(ViolationHours, 0000, 0200) ~ "12-2",  
 between(ViolationHours, 0200, 0400) ~ "2-4",  
 between(ViolationHours, 0400, 0600) ~ "4-6",  
 between(ViolationHours, 0600, 0800) ~ "6-8",  
 between(ViolationHours, 0800, 1000) ~ "8-10",  
 between(ViolationHours, 1000, 1200) ~ "10-12",  
 between(ViolationHours, 1200, 1259) ~ "12-2" )) %>%   
 group\_by(`Violation\_Code`, ViolationTime, TimeRange) %>%   
 summarise(count=n()) %>%   
 arrange(desc(count)) %>%   
 filter(`Violation\_Code`== 36 |`Violation\_Code`== 21 |`Violation\_Code`== 38)

df\_VC ## Viewing the table

## # A tibble: 33 x 4  
## # Groups: Violation\_Code, ViolationTime [6]  
## Violation\_Code ViolationTime TimeRange count  
## <dbl> <chr> <chr> <int>  
## 1 21 A 8-10 35122  
## 2 21 A 10-12 20123  
## 3 36 A 10-12 19342  
## 4 36 P 12-2 18753  
## 5 36 A 8-10 15552  
## 6 38 P 12-2 11612  
## 7 38 P 2-4 9866  
## 8 38 A 10-12 8673  
## 9 36 P 2-4 8159  
## 10 38 P 4-6 7820  
## # ... with 23 more rows

Below, a Grouped Bar Chart was created using ggplot geom\_bar to show the difference between the count of the Top 3 Violation Codes by AM/PM.Aesthetics used were Violation Code and the grouped bars are filled by Violation Time as AM or PM.It was colored red and blue using scale\_fill. Gg\_title was used to give title to the graph.Facet\_wrap was used to show top 3 Violation Codes in separate grids alongside.

ggplot (df\_VC, aes(x=`Violation\_Code`, y=count, fill=ViolationTime))+  
 geom\_bar(stat="identity", position = "dodge") +   
 scale\_fill\_brewer(palette = "Set1")+  
 ggtitle("Violation Times in AM & PM")+  
 facet\_wrap(~`Violation\_Code`)+  
 geom\_smooth(size=2)



**Observation**

VC 21 has higher occurrences in the AM hours as compared to its PM occurrences. For VC 36, the occurrences in both AM and PM are about the same. As for VC 38, AM occurrences are slightly higher than PM.

Below, a Grouped bar chart was created using ggplot geom\_bar to show the VC codes 21, 36, 38, by Time Range and Count and by AM and PM. Aesthetics used were TimeRange and it was filled by AM or PM. Geom\_Smooth was used to draw a line through the data, however, being in different grids that could not be shown. It was colored using scale\_fill. gg\_title was used to give title to the graph. Facet\_wrap was used to show top 3 Violation Codes in separate grids alongside.

## Conclusions:

1. From the analysis, we can conclude that all the variables taken such as Violation code in particular Standing Zone, New York county from Violation County,5th May 2017 from Issue Date, and Vehicle make Ford had major impact on the parking violation issued in New York city when compared to other variables. Hence, we can say that in every way possible violations are more and thus increasing the revenue for the cities by paying through violation payment.
2. We also observed that certain precincts/streets/locations had a greater number of violations. By knowing that, authorities can identify the reasons for the high number of violations there and curb them depending upon the reasons.
3. From the output observed, Violation Code 21, 36 and 38 are the most frequently occurring violations. VC 21 means Street Cleaning: No parking where parking is not allowed by sign, street marking or traffic control device. Whereas VC 36 is: exceeding the posted speed limit in or near a designated school zone. VC 38 is when one fails to show a receipt or tag in the windshield given by the Parking Meter. Drivers get a 5-minute grace period past the expired time on parking meter receipts. Given the time(hours) of occurrences, the authorities can identify the time of the day when these violations are higher. For e.g. it was observed that VC 21 occurred mainly during office hours 8AM to 10AM, when people are in a rush to find a parking spot and park where it is not allowed. They can come up with discounts if one parks before 8AM. For VC 36, most speeding occurred during the school hours, here, they can then come up with measures like having speed breakers to lower the vehicle speeds or deploying traffic policemen near areas like school zones where exceeding speed limits can by highly dangerous. As for the VC 38, which occurs throughout the day and frequently, higher charges can be fined to the defaulters to prevent people from not showing parking tickets. ([linked phrase](https://www1.nyc.gov/site/finance/vehicles/services-violation-codes.page))
4. Our team have come to following conclusions based on our initial analysis of our New York parking violations data. 1.Initially, we focused on the variables like vehicle make, vehicle body type and cities that have impact on parking tickets. From our analysis, we can conclude that there is huge violation issued based on vehicle make compared to other variables taken. At the same time, we can conclude that Queens is the most violations issued city among Borough. 2.After finding the city with most violations, we focused to do our analysis on variables which can lead us in detail about Queens county. From the analysis, we can conclude that two violations are more often such as NO PARKING-DAY/TIME LIMITS and fire hydrant in May and June especially in the street of Queen Blvd.

## Summary of the changes in part1

We have done changes to our project part1 from professor’s feedback. The following are the changes:

1. We have added two more new dataset for performing analysis using modeling techniques.

2. We made sure that we remove warnings while we knit the file.

3. Explaining each and every line of code as much as possible

4. Legend of the charts and color was changed

5. We changed few research questions and now they are related to each other, so we tried to perform better in story telling this time.

6. We tried to apply this now to remove grammatical and case sensitive errors.

7. Overall, we tried to apply each, and every suggestion professor gave us on part1 and part2.

## Project part 2

## Libraries

Below are the libraries and data frames used for the project. Code for lower casing the column names has also been shown. We used inner join to join a new table to the original data set containing violation charges and description. We have also converted the df\_join dataframe into a tibble as it was required for converting column to row in the following code for clustering.

library(sparklyr)  
library(corrr)  
library(dbplot)  
library(cluster)  
library(factoextra)  
library(modelr)# provides easy pipeline modeling functions  
library(broom) # helps to tidy up model outputs  
library(gridExtra)  
library(kernlab) # SVM methodology  
library(e1071) # SVM methodology  
library(ISLR) # contains example data set "Khan"

sc <- spark\_connect(master = "local", version = "2.3")  
  
pv\_csv\_data<-spark\_read\_csv(sc, "parking\_violations1.csv")  
vc\_csv\_data<-spark\_read\_csv(sc, "violation\_charges1.csv")  
pv\_lower<-rename\_with(pv\_csv\_data, tolower) ##lower-casing the column names  
df\_join<-sparklyr::inner\_join(pv\_lower, vc\_csv\_data, by="violation\_code")  
df\_tibble<-as\_tibble(df\_join) ## converted to tibble as it was not visible

**K-Means Clustering**

Goal:

We use the K-Means Clustering method to classify and identify the counties which generate higher or lower average and total charges. This will also help understand the concentration of parking violations county-wise. Below, a new column is created to replace the existing N/A values with the more machine compatible NA in the violation charge column and then it has been converted to integer data type.

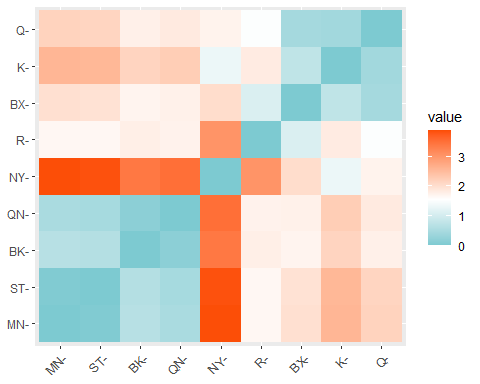
df\_tidy<-df\_tibble %>%   
 mutate(violation\_charge=ifelse(violation\_charge=="N/A", NA, violation\_charge)) %>%   
 mutate(violation\_charge=as.integer(violation\_charge))

After that, in df\_cluster dataframe, we grouped by violation county and two columns for average violation charge and total charges for each county has been created and then the violation county has been converted from column to row names.

df\_cluster<-df\_tidy %>%   
 group\_by(violation\_county) %>%   
 summarize(avg\_charge=mean(violation\_charge, na.rm=T),total\_charges=n()) %>%   
 column\_to\_rownames(var="violation\_county")

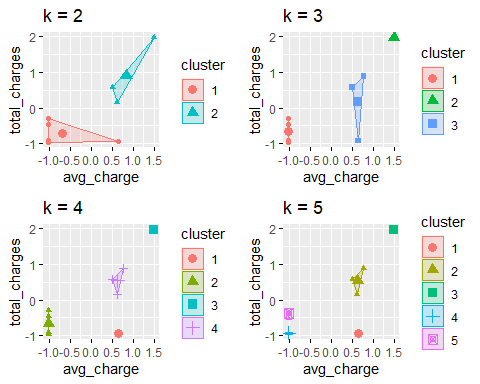
Below, we use the R scale function to standardize the variables, and we also create a distance matrix for visualizing the counties with similar average and total charges using the functions *get\_dist* and *fviz\_dist*. In the distance matrix, we can see that the counties are differentiated on the basis of their characteristics of high and low violation charges and have been grouped with the same color. As for the gradient, it shows the density or concentration depending on the value of charges.Counties with similar characteristics follow the same color code.

df\_scale<-df\_cluster %>% scale()   
distance<-get\_dist(df\_scale)  
fviz\_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))



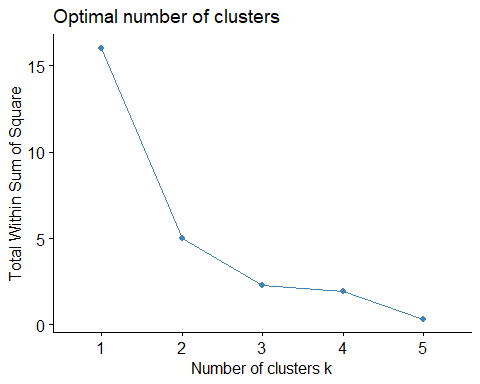
Then, we create clusters with different k values and visualize them in a grid to see which clusters look relevant and to get an idea as to how many clusters would be appropriate. By looking at these clusters we see that the plot with two clusters seems the most appropriate one as it is clustering all the counties into two separate distinguished clusters as compared to the others which are showing individual counties in silos.

k2 <- kmeans(df\_scale, centers = 2, nstart = 25)   
k3 <- kmeans(df\_scale, centers = 3, nstart = 25)  
k4 <- kmeans(df\_scale, centers = 4, nstart = 25)  
k5 <- kmeans(df\_scale, centers = 5, nstart = 25)  
  
p1 <- fviz\_cluster(k2, geom = "point", data = df\_scale) + ggtitle("k = 2")  
p2 <- fviz\_cluster(k3, geom = "point", data = df\_scale) + ggtitle("k = 3")  
p3 <- fviz\_cluster(k4, geom = "point", data = df\_scale) + ggtitle("k = 4")  
p4 <- fviz\_cluster(k5, geom = "point", data = df\_scale) + ggtitle("k = 5")  
  
grid.arrange(p1, p2, p3, p4, nrow = 2)

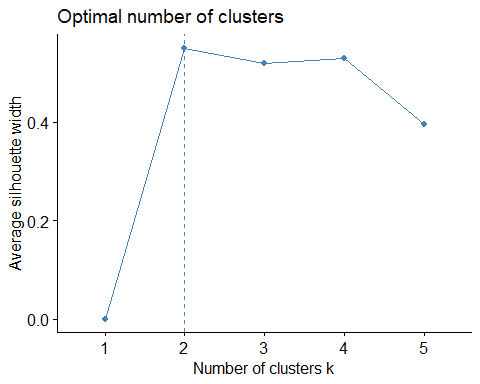


In the below code, we use the elbow, silhouette and the gap statistic methods to determine the optimal number of clusters that would be suitable for our data set. Looking at the plots, we can safely say that there are two possible clusters in which we can group our variables as we could see in the distance matrix. So, we finalize taking k = 2 clusters for our data.

set.seed(1234)  
  
# function to compute total within-cluster sum of square   
wss <- function(k) {  
 kmeans(df\_scale, k, nstart = 2 )$tot.withinss  
}  
  
##Elbow method  
  
set.seed(1234)  
fviz\_nbclust(df\_scale, FUNcluster= kmeans, method = "wss", k.max = 5)



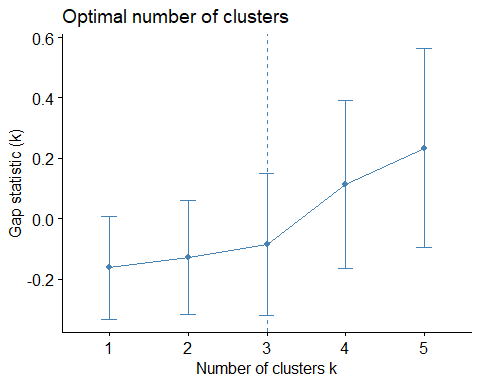
##Average Silhouette   
  
fviz\_nbclust(df\_scale, kmeans, method = "silhouette", k.max=5)



##Gap statistic  
set.seed(1234)  
gap\_stat <- clusGap(df\_scale, FUN = kmeans, nstart = 25,  
 K.max = 5, B = 50)  
  
print(gap\_stat, method = "firstmax")

## Clustering Gap statistic ["clusGap"] from call:  
## clusGap(x = df\_scale, FUNcluster = kmeans, K.max = 5, B = 50, nstart = 25)  
## B=50 simulated reference sets, k = 1..5; spaceH0="scaledPCA"  
## --> Number of clusters (method 'firstmax'): 5  
## logW E.logW gap SE.sim  
## [1,] 1.24807098 1.08524406 -0.16282692 0.1704086  
## [2,] 0.56415778 0.43656891 -0.12758888 0.1882793  
## [3,] 0.05503645 -0.03019312 -0.08522957 0.2339058  
## [4,] -0.55967145 -0.44660774 0.11306371 0.2774916  
## [5,] -1.14936747 -0.91541359 0.23395388 0.3298925

fviz\_gap\_stat(gap\_stat)



After finalizing the number of cluster,we run the final cluster with k = 2.

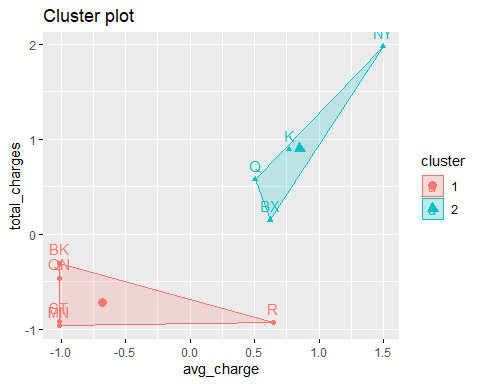
##K-means clustering with k = 2  
  
str(k2)

## List of 9  
## $ cluster : Named int [1:9] 1 2 2 1 2 2 1 1 1  
## ..- attr(\*, "names")= chr [1:9] "BK" "BX" "K" "MN" ...  
## $ centers : num [1:2, 1:2] -0.679 0.849 -0.717 0.897  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "1" "2"  
## .. ..$ : chr [1:2] "avg\_charge" "total\_charges"  
## $ totss : num 16  
## $ withinss : num [1:2] 2.59 2.42  
## $ tot.withinss: num 5.02  
## $ betweenss : num 11  
## $ size : int [1:2] 5 4  
## $ iter : int 1  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

tidy(k2)

## # A tibble: 2 x 5  
## avg\_charge total\_charges size withinss cluster  
## <dbl> <dbl> <int> <dbl> <fct>   
## 1 -0.679 -0.717 5 2.59 1   
## 2 0.849 0.897 4 2.42 2

fviz\_cluster(k2, data = df\_scale)



**Observation:**

The counties in the orange cluster: New York, Kings, Queens and Bronx generate the highest charges/revenue. Whereas, the counties in the Blue cluster: Brooklyn, Staten Island, Richmond, Manhattan generate the lowest charges/revenue.

In the above clusters, we observe that the Manhattan county, is shown in the cluster low number of violation and lower average charges, however, this county is the most densely populated of the five boroughs of New York. So to further understand it, we conducted a Support Vector Machine classification of the counties into precincts to figure if any precincts affected the clustering.

**Support Vector Machine**

Goal:

Classifying Violation Precincts based on the average charges and total number of charges/violations using Support Vector Machine method. Classes have been divided as high violation precincts and low violation precincts. Classifying and identifying precincts can help the authorities pinpoint the causes for high violation and implement measures to control them.

In the below code, we have aggregated violation precincts on the basis of average violation charges and total number of charges.

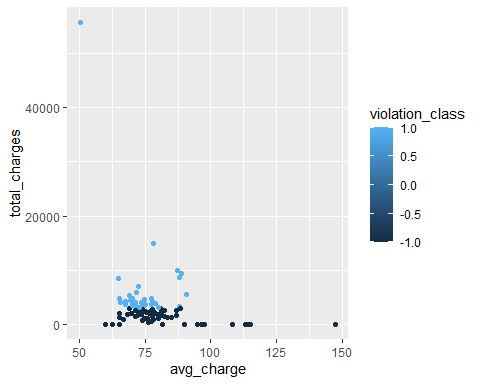
df\_svm<-df\_tidy %>%   
group\_by(violation\_precinct) %>%   
summarize(avg\_charge=mean(violation\_charge, na.rm=TRUE), total\_charges=n())

Then we classified the precincts data based on the high and low average charges based on observations in cluster analysis.

df\_svm\_class<- df\_svm %>%  
mutate(df\_svm,violation\_class=ifelse(total\_charges > 3000, 1 , -1)) %>%  
column\_to\_rownames(var="violation\_precinct")

The following ggplot was used for visualizing the data before doing an SVM fit plot.

df\_svm\_class %>%   
ggplot(mapping=aes(x=avg\_charge, y=total\_charges, color = violation\_class))+  
geom\_point()



Now, we use the SVM model to classify the violation classes based on the classes identified above. Thereafter, we tried to plot the SVM model.

# Fit Support Vector Machine model to data set  
svmfit <- svm(violation\_class~., data = df\_svm\_class, kernel = "linear", scale = FALSE)  
  
# Plot Results  
plot(svmfit, data = df\_svm\_1, violation\_class~.)

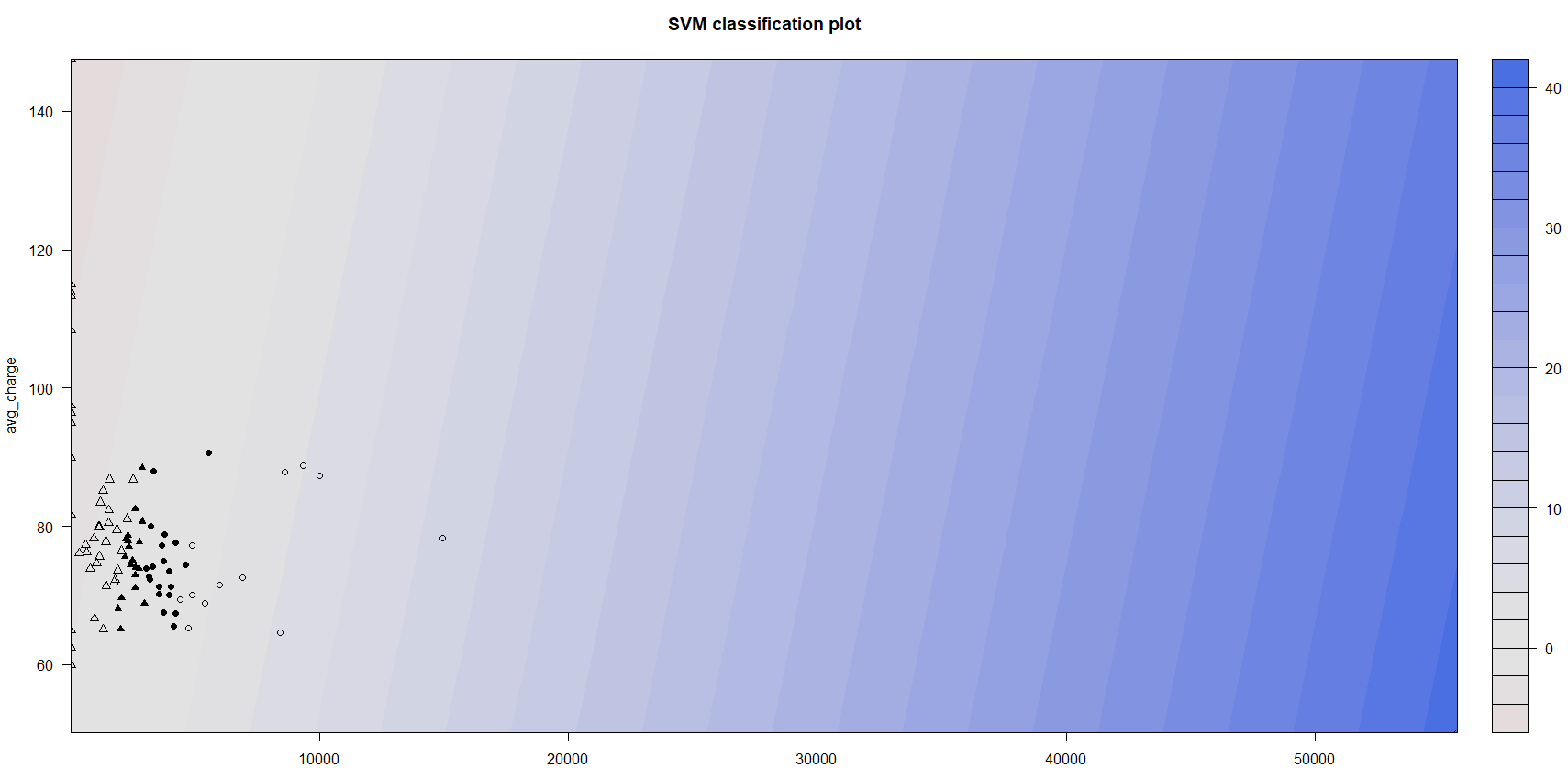
***Note:***

*The svmfit model plot was executing without error, however it was not showing a plot due to possible issues with the package as noted in the doubt session. Therefore, the below kernfit plot was used.*

# fit model and produce plot  
kernfit <- ksvm(x = violation\_class~., data = df\_svm\_class, type = "C-svc", kernel = 'vanilladot')

## Setting default kernel parameters

# Plot Results  
plot(kernfit, data = df\_svm\_class)



**Observation:**

We observe that the average charges and total number of violations may appear higher due to the higher charges and violations in some of the precincts. The plot exhibits that with the dots towards the right with darker gradient. Classifying and identifying precincts can help the authorities pinpoint the causes for high violation and implement measures to control them in the respective highly impacted precincts.

Limitation:

Lack of numeric variables; difficult to identify classes based on these variables.

**Linear Regression**

Goal:

To analyze the trend of the number of violations on different times of the year to identify the peaks and minimizing the number of violations.

Below, we aggregate the data by issue date to identify number of violation and total charges applied on a particular day.

set.seed(123)  
  
df\_regression<-df\_tidy %>%   
 group\_by(issue\_date) %>%   
 summarize(sum\_charges=sum(violation\_charge, na.rm=T), no\_of\_violations=n()) %>%   
 column\_to\_rownames(var="issue\_date")

Below, we divided the dataset into training and test data by 60% and 40% respectively.

sample <- sample(c(TRUE, FALSE), nrow(df\_regression), replace = T, prob = c(0.6,0.4))  
train <- df\_regression[sample, ]  
test <- df\_regression[!sample, ]

A linear regression model was created using training data with number of violations as the independent variable and sum of charges as the dependent variable.

model1 <- lm(sum\_charges ~ no\_of\_violations , data = train) #model building

Then, we run the summary on the model as below.

summary(model1) #model output and accuracy

##   
## Call:  
## lm(formula = sum\_charges ~ no\_of\_violations, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5378.1 -1596.1 -196.9 1467.1 5962.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3815.8330 356.3306 10.71 <2e-16 \*\*\*  
## no\_of\_violations 65.2348 0.4081 159.87 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2136 on 217 degrees of freedom  
## Multiple R-squared: 0.9916, Adjusted R-squared: 0.9915   
## F-statistic: 2.556e+04 on 1 and 217 DF, p-value: < 2.2e-16

confint(model1) #assessing coefficients

## 2.5 % 97.5 %  
## (Intercept) 3113.52093 4518.14508  
## no\_of\_violations 64.43058 66.03909

Observation:

Since P value < α (0.05), total charges in a county are dependent on the number of violations. In counties where total charges in a day are higher than that in the other counties, if the charges are increased by a minimal percentage, say 10%, the total revenue from parking ticket violations will increase substantially and in a shorter time frame than compared with the counties having lesser number of violations. Increasing violation charge rate may also be used to reduce the number of violations in those counties.

The coefficients indicates that with 95% level of significance the value of the sum of charges may increase between $64+$6.4 = $70.4(10% increase) to $66+$6.6=$72.6(10% increase) if the violation charges are increased by 10%. So, we can say that there is definitely a positive relationship between the number of violations and the total charges.

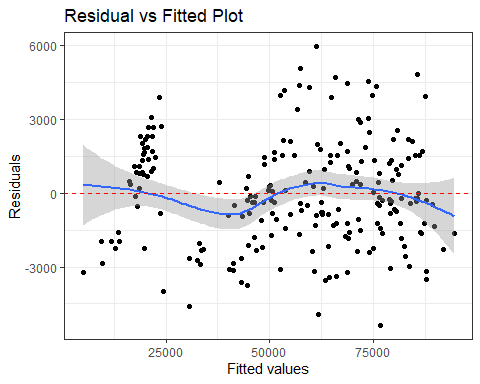
The Residual Standard Error indicates that the total charges will deviate from the true regression line by approximately 2136 units.

The R Squared result suggests that our model with 1 predictor can explain 99% of the variability in our parking violation charges data. R Square value is 0.9916 and the adjusted R2 value is 0.9915. Since we do not have much difference between the two values and also since we have only one predictor variable, we say that the predictor variable is relevant and the complexity is justified.

Since our F statistic is greater than 1 and the p-value P < α (0.05) is significant, we can say that our model is a good fit and number of violations is a significant predictor of violation charges.

Based on the above, the residuals vs fitted plot was drawn on model 1 data which shows number of charges and their sum distributed across the days in the year.

model1\_results <- augment(model1, train) %>%  
 mutate(Model = "Model 1")# add model diagnostics to our training data  
  
ggplot(model1\_results, aes(.fitted, .resid))+   
 geom\_point()+  
 stat\_smooth(method="loess")+  
 geom\_hline(yintercept=0, col="red", linetype="dashed")+  
 xlab("Fitted values")+  
 ylab("Residuals")+  
 ggtitle("Residual vs Fitted Plot")+  
 theme\_bw()



Final observation:

Analyzing the results from the clustering, svm and regression, we observed how counties fall in high violation and low violation classes; how precincts impact the classification of the counties in the categories, and how total number of violations affect charges based on certain times in the year. Further deep diving into the analysis can help the authorities target those precincts given the times of the year when the violations were high implement measures like newspaper articles, TV campaign, town halls, increasing the charges etc. to minimize the number of violations in those precincts.

## Data importing and Manipulations

we are importing two other data sets that will be useful for performing modeling techniques because there are not enough numerical variables to perform clustering and regression analysis in the existing dataset.

vc <- read.csv("violation\_charges2.csv",stringsAsFactors = FALSE)  
df2 <- read.csv("Open\_Parking\_and\_Camera\_Violations.csv")  
df3 <- df2 %>% sample\_n(500001, replace = FALSE)  
df3<- df3[ -c(1,2,3,8,17,18,19)]  
df3 <- df3 %>%   
 select\_all(~gsub("\\s+|\\.", "\_", .)) %>%   
 select\_all(tolower)  
colnames(df3)[4] <-"violation\_description"  
colnames(df3)[11] <-"violation\_county"

We are merging two datasets by violation county so that we can use the numerical variables such as payment amount, fine amount, interest amount etc in our joining. we are using merge function here.

by\_amount <- df3 %>% select(c(fine\_amount,penalty\_amount,reduction\_amount,payment\_amount,  
 violation\_county)) %>%   
 group\_by(violation\_county) %>%   
 summarise\_all(mean)  
by\_count <- df3 %>% group\_by(violation\_county) %>%   
 summarise(total\_tickets = n())  
  
vc<- vc[ -c(3)]  
#merging by\_amount and by\_count dataframe  
final\_df <- merge(by\_amount,by\_count)

### Requirements

We will use New york Violations data, Violation charges and open parking and camera violations provided by the NYC open data website. We will also use a few packages that provide data manipulation, visualization, pipeline modeling functions, and model output tidying functions.

library(modelr) # provides easy pipeline modeling functions  
library(broom)# helps to tidy up model outputs

### Data Preparation

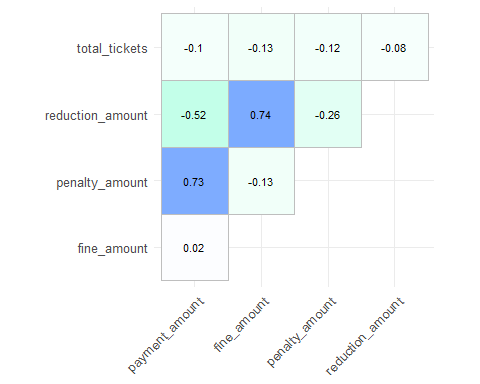
We will use a conventional 60% / 40% split where we training our model on 60% of the data and then test the model performance on 40% of the data that is withheld.

set.seed(123)  
sample <- sample(c(TRUE, FALSE), nrow(final\_df), replace = T, prob = c(0.6,0.4))  
train <- final\_df[sample, ]  
test <- final\_df[!sample, ]

## Correlation

install.packages("ggcorrplot")

library(ggcorrplot)  
  
c <- final\_df %>% select(payment\_amount,fine\_amount,penalty\_amount,  
 reduction\_amount,total\_tickets)  
options(repr.plot.width=12, repr.plot.height=12)  
corr <- round(cor(c, use="complete.obs"), 2)  
ggcorrplot(corr, lab = TRUE,colors = c("aquamarine", "white", "dodgerblue"),  
 show.legend = F, outline.color = "gray", type = "upper",  
 tl.cex = 10, lab\_size = 3, sig.level = .1) +  
 labs(fill = "Correlation")



### 

### Multiple Regression Model

model2 <- lm(payment\_amount ~ fine\_amount + reduction\_amount + penalty\_amount, data = train)

### Model2 output:

summary(model2)

##   
## Call:  
## lm(formula = payment\_amount ~ fine\_amount + reduction\_amount +   
## penalty\_amount, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -20.710 -2.986 -0.561 1.180 39.579   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.97166 3.76046 -0.258 0.797   
## fine\_amount 0.93397 0.05363 17.415 <2e-16 \*\*\*  
## reduction\_amount -0.93225 0.04540 -20.533 <2e-16 \*\*\*  
## penalty\_amount 1.48791 0.06982 21.311 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.697 on 72 degrees of freedom  
## Multiple R-squared: 0.9419, Adjusted R-squared: 0.9395   
## F-statistic: 389.4 on 3 and 72 DF, p-value: < 2.2e-16

tidy(model2)

## # A tibble: 4 x 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -0.972 3.76 -0.258 7.97e- 1  
## 2 fine\_amount 0.934 0.0536 17.4 1.60e-27  
## 3 reduction\_amount -0.932 0.0454 -20.5 8.08e-32  
## 4 penalty\_amount 1.49 0.0698 21.3 8.06e-33

We can see from the summary of our models that our coefficients for fine amounts , penalty amount and reduction amounts with payment amount are statistically significant (p-value < 0.05).

In this case, all the three independent variables are significant and so we proceed to interpret its coefficient estimate (impact on the dependent variable). The fine amount coefficient suggests that for every dollar increase in payment amount, holding all other predictors constant, we can expect an increase of 0.93 units in fine amount, on average.

The reduction amount coefficient suggests that for every dollar increase in payment amount, holding all other predictors constant, we can expect an decrease of 0.9 units in reduction amount on average.

The penalty amount coefficient suggests that for every dollar increase in payment amount, holding all other predictors constant, we can expect an increase of 1.4 units in penalty amount, on average.

### Assessing Coefficients for model2

confint(model2)

## 2.5 % 97.5 %  
## (Intercept) -8.467992 6.5246799  
## fine\_amount 0.827059 1.0408791  
## reduction\_amount -1.022761 -0.8417444  
## penalty\_amount 1.348733 1.6270920

Our results show us that our 95% confidence interval for β1 (fine amount) is [8.4 .6.52] Thus, since zero is not in this interval we can conclude that as the fine amount increases by dollar we can expect the payment amount to increase by 0.8-1.0 units. Our results show us that our 95% confidence interval for β2 (reduction amount) is [.217, .748]. Thus, since zero is not in this interval we can conclude that as the reduction amount increases by dollar we can expect the payment to increase by 2.1-7.5 units.

Thus, we can conclude that there is a positive relationship between fine amount and reduction amount with payment amount.

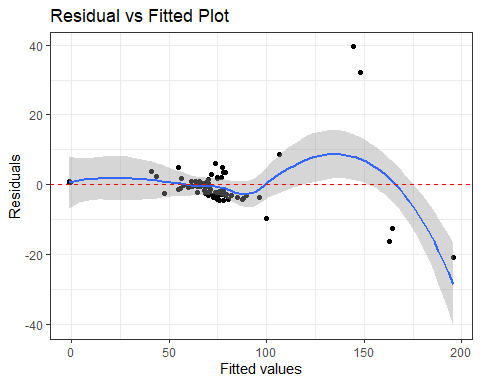
### Assesing model2 accuracy

In our summary print out above for model 2 we saw that F = 7.507 with p<0.05 suggesting that our fine amount, reduction amount and penalty amount must be related to payment amount.Combined, our RSE, R2, and F-statistic results suggest that our model has a good fit.

### Assessing Our Model2 Visually

The first plot you want to visualize is the residuals versus fitted values plot.

ggplot(model2, aes(.fitted, .resid))+   
 geom\_point()+  
 stat\_smooth(method="loess")+  
 geom\_hline(yintercept=0, col="red", linetype="dashed")+  
 xlab("Fitted values")+  
 ylab("Residuals")+  
 ggtitle("Residual vs Fitted Plot")+  
 theme\_bw()



The above plots enable us to analyze two important concerns (assumptions):

Non-linearity: Our plot indicates that the assumption of linearity is not fair for model 2.

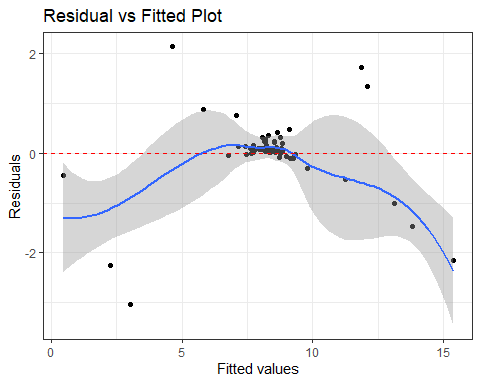
Heteroskedasticity: Our plot indicates that there is a funnel shape with our residuals, as slightly in our plot, then we have violated the homoskedasticity assumption. Sometimes this can be resolved with a log or square root transformation of Y in our model.

### Applying square root transformation for model2

model2a <- lm(sqrt(payment\_amount) ~ fine\_amount + reduction\_amount + penalty\_amount, data = train)  
  
model1\_results <- augment(model1, train)  
model2a\_results <- augment(model2a, train)# add model diagnostics to our training data

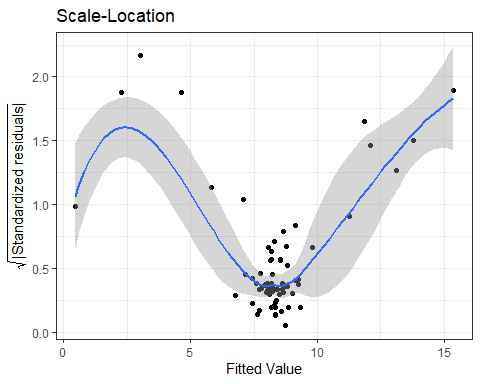
#Assessing Our Model Visually

ggplot(model2a\_results, aes(.fitted, .resid)) +   
 geom\_point()+  
 stat\_smooth(method="loess")+  
 geom\_hline(yintercept=0, col="red", linetype="dashed")+  
 xlab("Fitted values")+  
 ylab("Residuals")+  
 ggtitle("Residual vs Fitted Plot")+  
 theme\_bw()

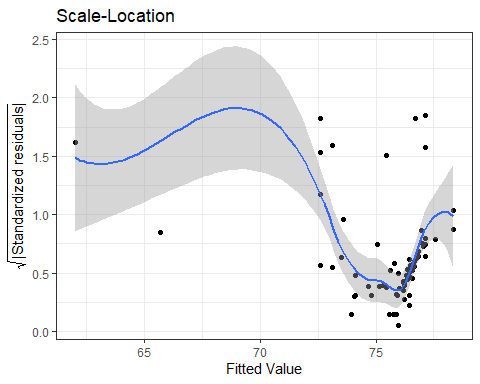


The second is the scale-location plot. This plot shows if residuals are spread equally along the ranges of predictors. This provide you an alternative way to check the assumption of equal variance (homoscedasticity). It’s good if you see a horizontal line with equally (randomly) spread points. ### Normality of the residuals for model1 and model2

ggplot(model2a\_results, aes(.fitted, sqrt(abs(.std.resid))))+  
 geom\_point(na.rm=TRUE) +  
 stat\_smooth(method="loess", na.rm = TRUE)+  
 xlab("Fitted Value")+  
 ylab(expression(sqrt("|Standardized residuals|")))+  
 ggtitle("Scale-Location")+  
 theme\_bw()



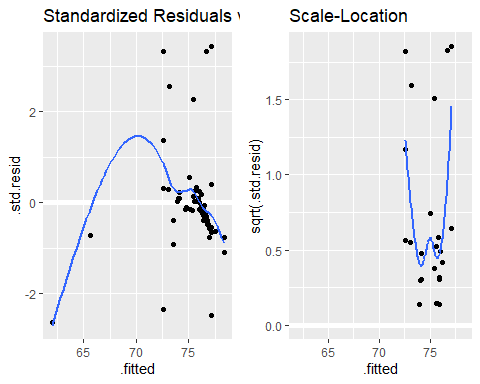
ggplot(model1\_results, aes(.fitted, sqrt(abs(.std.resid))))+  
 geom\_point(na.rm=TRUE) +  
 stat\_smooth(method="loess", na.rm = TRUE)+  
 xlab("Fitted Value")+  
 ylab(expression(sqrt("|Standardized residuals|")))+  
 ggtitle("Scale-Location")+  
 theme\_bw()



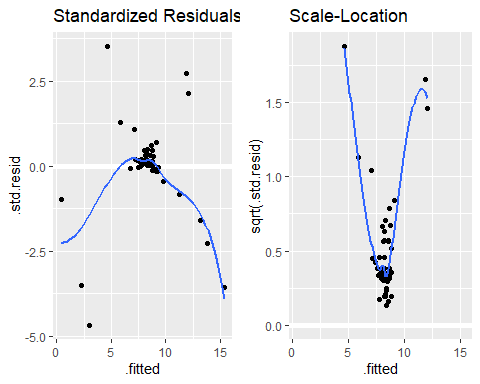
OR

## Error in eval(expr, envir, enclos): object 'OR' not found

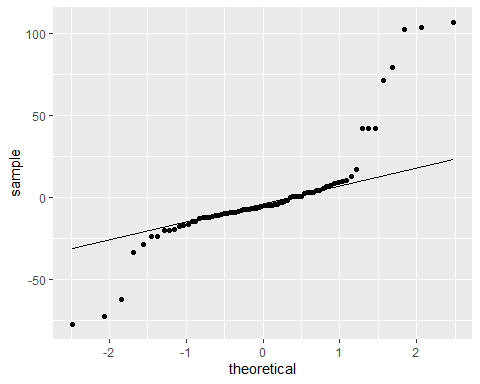
p1 <- ggplot(model1\_results, aes(.fitted, .std.resid)) +  
 geom\_ref\_line(h = 0) +  
 geom\_point() +  
 geom\_smooth(se = FALSE) +  
 ggtitle("Standardized Residuals vs Fitted")  
  
p2 <- ggplot(model1\_results, aes(.fitted, sqrt(.std.resid))) +  
 geom\_ref\_line(h = 0) +  
 geom\_point() +  
 geom\_smooth(se = FALSE) +  
 ggtitle("Scale-Location")  
  
gridExtra::grid.arrange(p1, p2, nrow = 1)



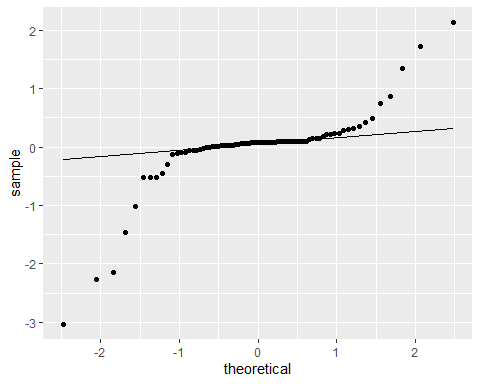
p1\_2 <- ggplot(model2a\_results, aes(.fitted, .std.resid)) +  
 geom\_ref\_line(h = 0) +  
 geom\_point() +  
 geom\_smooth(se = FALSE) +  
 ggtitle("Standardized Residuals vs Fitted")  
  
p2\_2 <- ggplot(model2a\_results, aes(.fitted, sqrt(.std.resid))) +  
 geom\_ref\_line(h = 0) +  
 geom\_point() +  
 geom\_smooth(se = FALSE) +  
 ggtitle("Scale-Location")  
  
gridExtra::grid.arrange(p1\_2, p2\_2, nrow = 1)



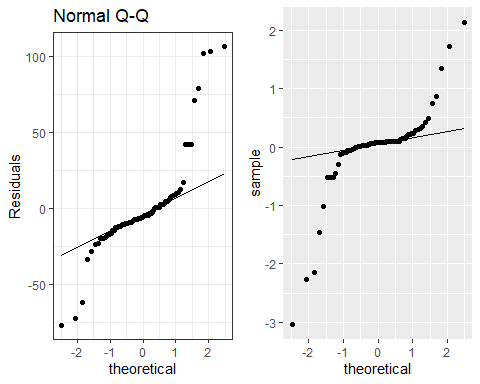
ggplot(model1\_results, aes(sample = .resid)) +  
 stat\_qq() +  
 stat\_qq\_line()



ggplot(model2a\_results, aes(sample = .resid)) +  
 stat\_qq() +  
 stat\_qq\_line()



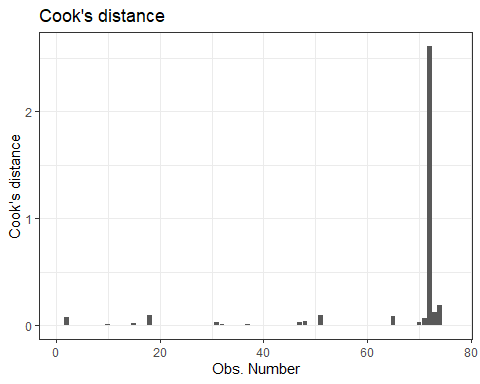
p3 <- ggplot(model1\_results, aes(sample = .resid)) +  
 stat\_qq() +  
 stat\_qq\_line()+  
 ylab("Residuals")+  
 ggtitle("Normal Q-Q")+  
 theme\_bw()# try also look at the std residuals (.std.resid). Left: model 1  
p4 <- ggplot(model2a\_results, aes(sample = .resid)) +  
 stat\_qq() +  
 stat\_qq\_line()# Right: model 2  
gridExtra::grid.arrange(p3, p4, nrow = 1)



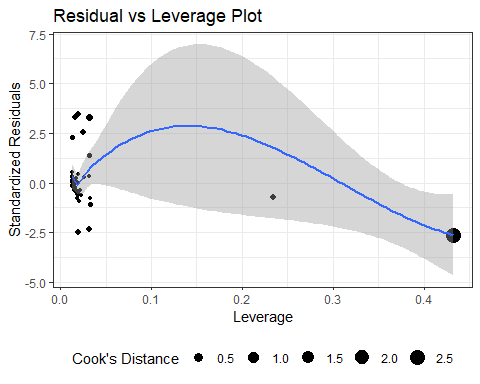
In our case we have a little deviation in the bottom left-hand side which likely is the concern we mentioned earlier that as the reduction amount approaches 0 the relationship with payment amount appears to start veering away from a linear relationship.

### Cook’s Distance and residuals versus leverage plot for model1

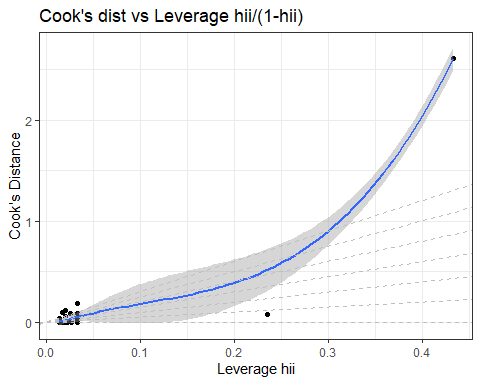
ggplot(model1\_results, aes(seq\_along(.cooksd), .cooksd))+  
 geom\_bar(stat="identity", position="identity")+  
 xlab("Obs. Number")+  
 ylab("Cook's distance")+  
 ggtitle("Cook's distance")+  
 theme\_bw()



ggplot(model1\_results, aes(.hat, .std.resid))+  
 geom\_point(aes(size=.cooksd), na.rm=TRUE)+  
 stat\_smooth(method="loess", na.rm=TRUE)+  
 xlab("Leverage")+  
 ylab("Standardized Residuals")+  
 ggtitle("Residual vs Leverage Plot")+  
 scale\_size\_continuous("Cook's Distance", range=c(1,5))+  
 theme\_bw()+  
 theme(legend.position="bottom")

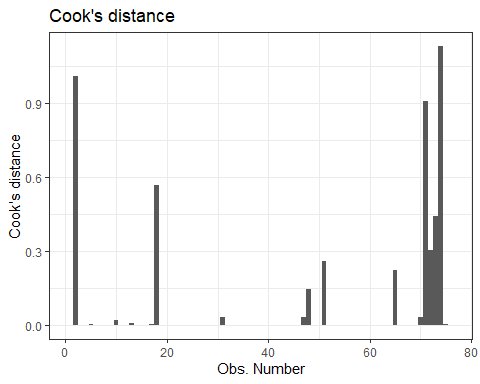


ggplot(model1\_results, aes(.hat, .cooksd))+  
 geom\_point(na.rm=TRUE)+  
 stat\_smooth(method="loess", na.rm=TRUE)+  
 xlab("Leverage hii")+  
 ylab("Cook's Distance")+  
 ggtitle("Cook's dist vs Leverage hii/(1-hii)")+  
 geom\_abline(slope=seq(0,3,0.5), color="gray", linetype="dashed")+  
 theme\_bw()

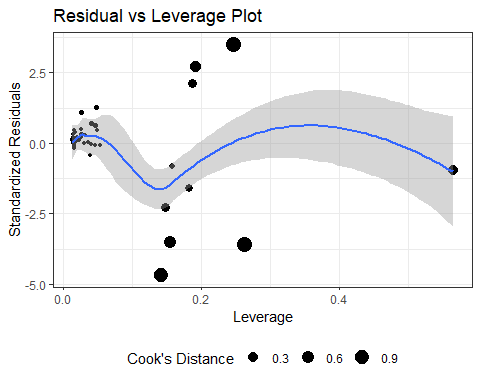


### Cook’s Distance and residuals versus leverage plot for model2

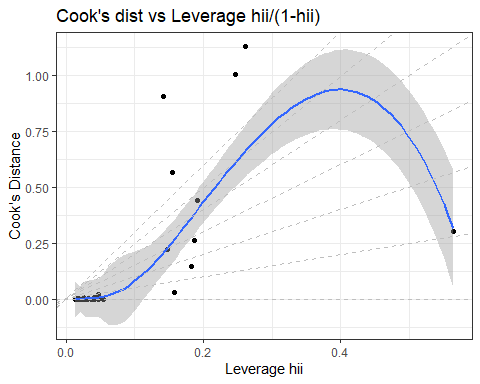
ggplot(model2a\_results, aes(seq\_along(.cooksd), .cooksd))+  
 geom\_bar(stat="identity", position="identity")+  
 xlab("Obs. Number")+  
 ylab("Cook's distance")+  
 ggtitle("Cook's distance")+  
 theme\_bw()



ggplot(model2a\_results, aes(.hat, .std.resid))+  
 geom\_point(aes(size=.cooksd), na.rm=TRUE)+  
 stat\_smooth(method="loess", na.rm=TRUE)+  
 xlab("Leverage")+  
 ylab("Standardized Residuals")+  
 ggtitle("Residual vs Leverage Plot")+  
 scale\_size\_continuous("Cook's Distance", range=c(1,5))+  
 theme\_bw()+  
 theme(legend.position="bottom")



ggplot(model2a\_results, aes(.hat, .cooksd))+  
 geom\_point(na.rm=TRUE)+  
 stat\_smooth(method="loess", na.rm=TRUE)+  
 xlab("Leverage hii")+  
 ylab("Cook's Distance")+  
 ggtitle("Cook's dist vs Leverage hii/(1-hii)")+  
 geom\_abline(slope=seq(0,3,0.5), color="gray", linetype="dashed")+  
 theme\_bw()



### top 5 observations with the highest Cook’s distance

model1\_results %>%  
 top\_n(5, wt = .cooksd)

## # A tibble: 5 x 13  
## .rownames violation\_county fine\_amount penalty\_amount reduction\_amount  
## <chr> <int> <dbl> <dbl> <dbl>  
## 1 36 37 115 0 115   
## 2 85 93 65 60 1.07  
## 3 120 401 243. 0 243.   
## 4 122 425 60 60 0   
## 5 123 803 115 60 0   
## # ... with 8 more variables: payment\_amount <dbl>, total\_tickets <int>,  
## # .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,  
## # .std.resid <dbl>

model2a\_results %>%  
 top\_n(5, wt = .cooksd)

## # A tibble: 5 x 13  
## .rownames violation\_county fine\_amount penalty\_amount reduction\_amount  
## <chr> <int> <dbl> <dbl> <dbl>  
## 1 3 2 199. 2.5 155.  
## 2 36 37 115 0 115   
## 3 119 400 60 0 60   
## 4 122 425 60 60 0   
## 5 123 803 115 60 0   
## # ... with 8 more variables: payment\_amount <dbl>, total\_tickets <int>,  
## # .fitted <dbl>, .resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>,  
## # .std.resid <dbl>

### Making Predictions

We can compare the test sample MSE to the training sample MSE with the following.

test %>%   
 add\_predictions(model1) %>%  
 summarise(MSE = mean((payment\_amount - pred)^2))# test MSE

## MSE  
## 1 NA

train %>%   
 add\_predictions(model1) %>%  
 summarise(MSE = mean((payment\_amount - pred)^2))# training MSE

## MSE  
## 1 961.0277

we can see that MSE value for train model is 779.51 which is lower compared to test model with 1083.3. Hence, train model is more preferred for model1.

test %>%   
 add\_predictions(model2a) %>%  
 summarise(MSE = mean((payment\_amount - pred)^2))# test MSE

## MSE  
## 1 NA

train %>%   
 add\_predictions(model2a) %>%  
 summarise(MSE = mean((payment\_amount - pred)^2))# training MSE

## MSE  
## 1 5346.216

we can see that MSE value for train model is 860.8532 which is lower compared to test model with 1232.3 Hence, train model is also preferred for model2a. ### compare MSE on test and train data

test %>%  
 gather\_predictions(model1, model2a) %>%  
 group\_by(model) %>%  
 summarise(MSE = mean((payment\_amount-pred)^2))

## # A tibble: 2 x 2  
## model MSE  
## \* <chr> <dbl>  
## 1 model1 NA  
## 2 model2a NA

train %>%  
 gather\_predictions(model1, model2a) %>%  
 group\_by(model) %>%  
 summarise(MSE = mean((payment\_amount-pred)^2))

## # A tibble: 2 x 2  
## model MSE  
## \* <chr> <dbl>  
## 1 model1 961.  
## 2 model2a 5346.

Here we see that model 1 drastically reduces MSE on the out-of-sample. So although we still have lingering concerns over residual normality model 1 is still the preferred model so far.

### Project Summary:

Using the K means clustering method and the SVM method we identified the counties and precincts with higher number of violations and average charges, and the counties having lower number of violations and average charges. So, the authorities can take measures accordingly to find the causes of violations in those counties and precincts to reduce the number of violations by putting in appropriate measures.

Using linear regression and multiple regression we identified that variables fine, reduction and penalty amount has significant effect on payment and found that our model is a good fit. From the above analysis, we can produce improvements for pedestrians, vehicular safety through education and a timely decision of Parking, and traffic violations. Also, we can make suggestions for reduction of the violations.