**NYC Parking Tickets: An Exploratory Analysis**

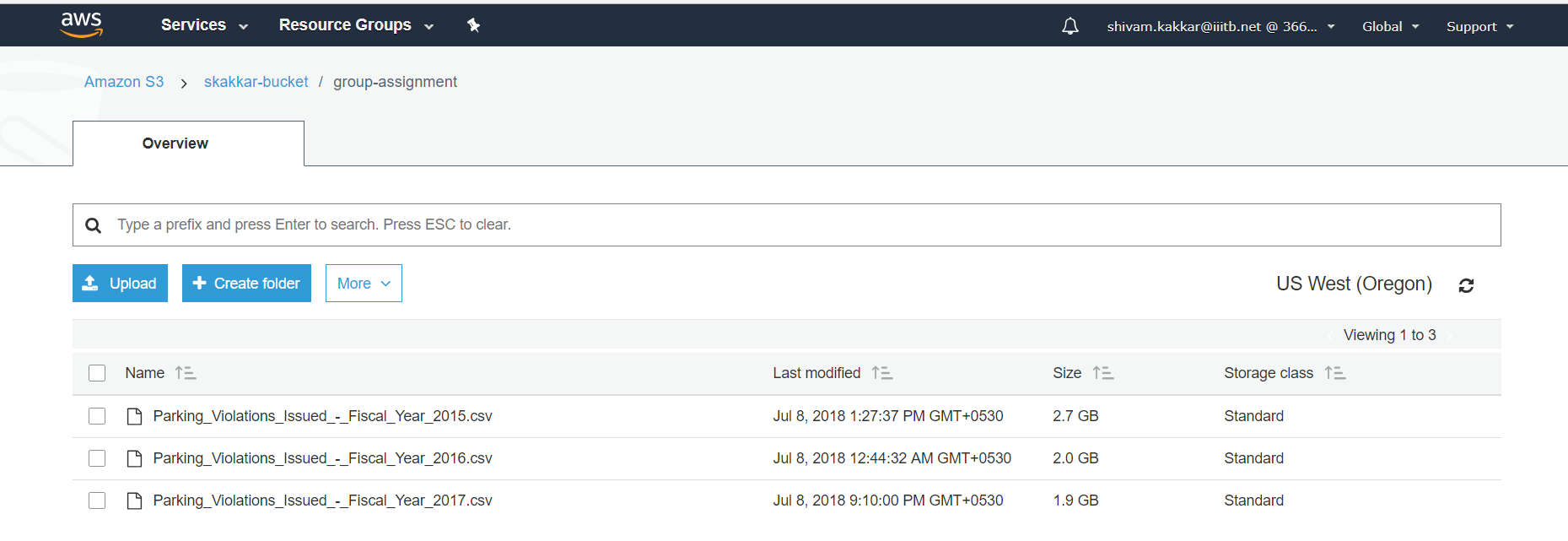
- Group Assignment

# Group Members:

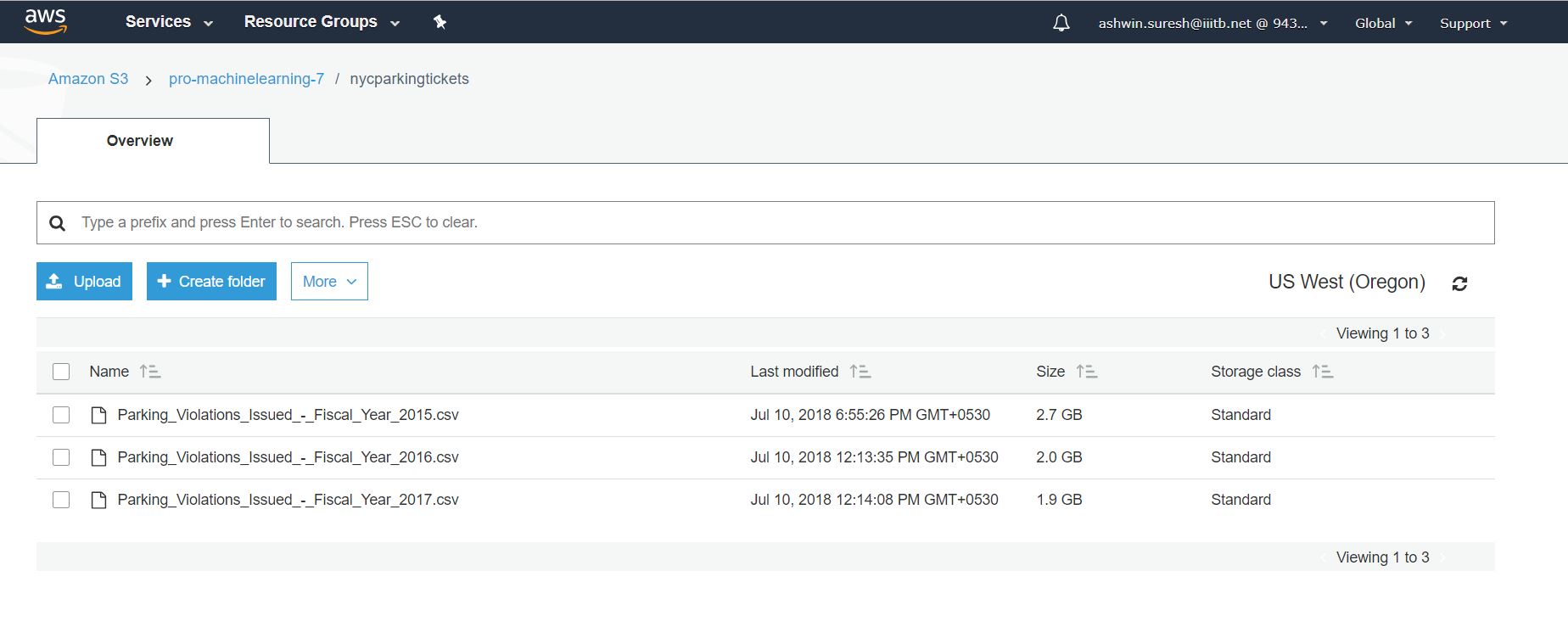
|  |  |
| --- | --- |
| **Name** | **AWS User name** |
| Shivam Kakkar | shivam.kakkar@iiitb.net |
| Ashwin Suresh | ashwin.suresh@iiitb.net |
| Manohar Shanmugasundaram | manohar.shanmugam@iiitb.net |
| P Sai Prathyusha | pabba.prathyusha@iiitb.net |

**Task 1: AWS S3 bucket screen shot**

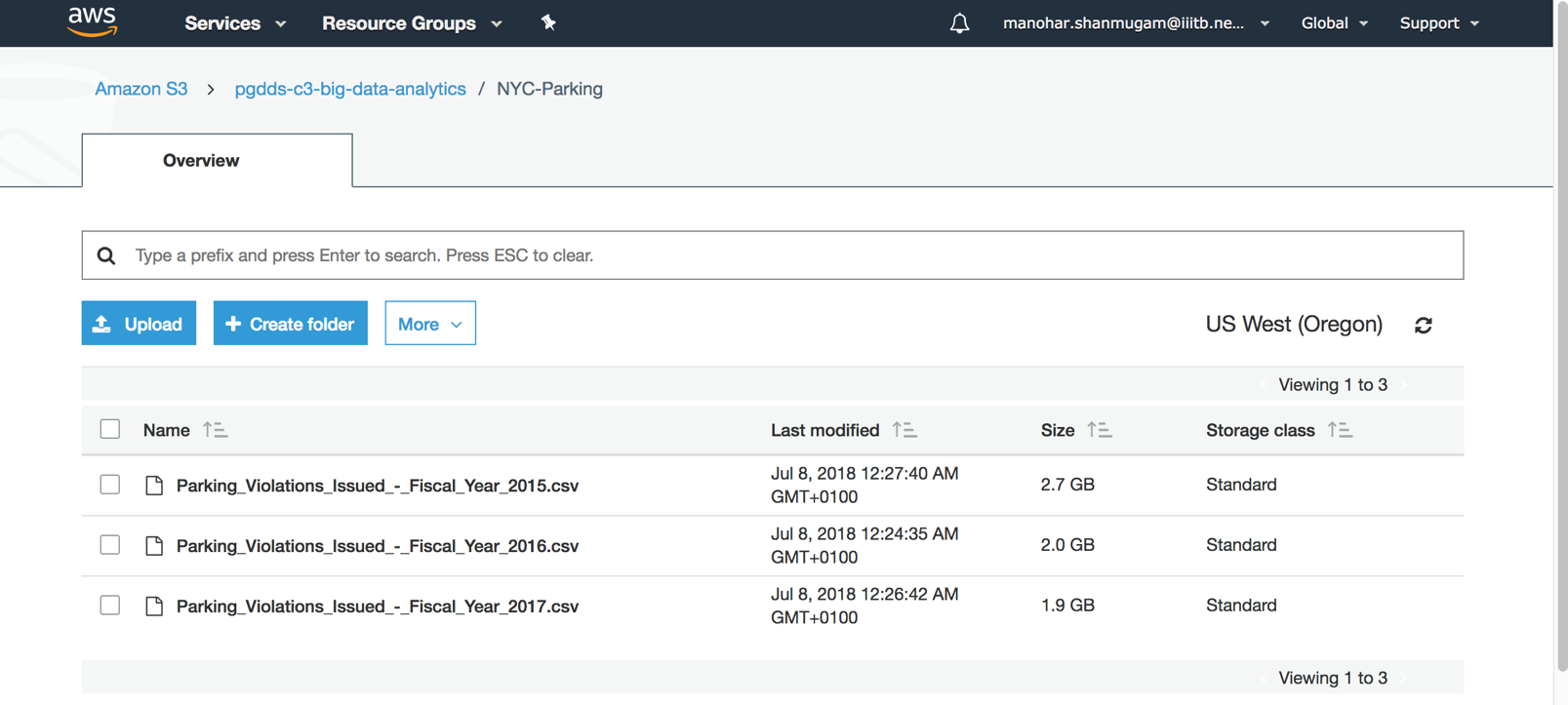
**Shivam Kakkar:**

****

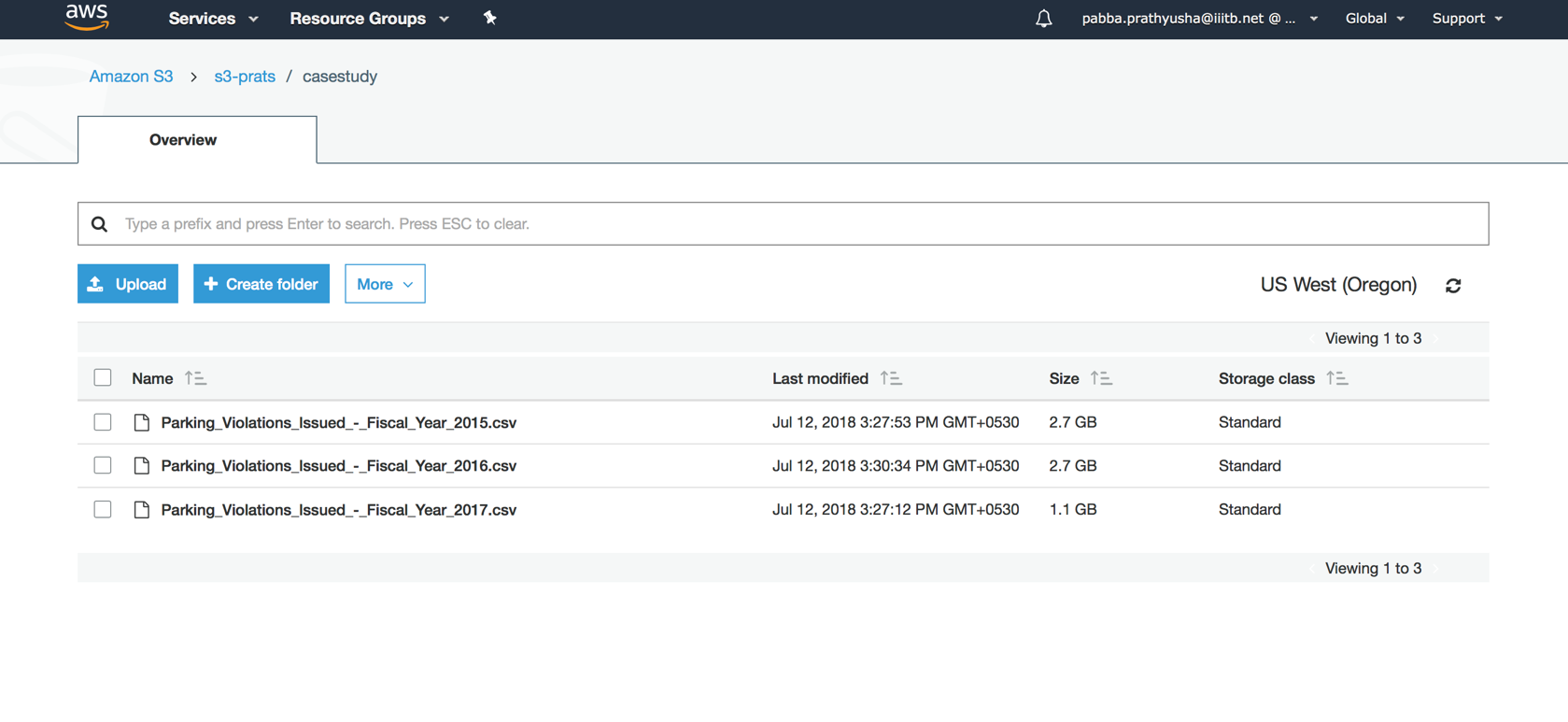
**Ashwin Suresh:**



**Manohar Shanmugasundaram:**



**P Sai Prathyusha:**



HIGHLIGHTS

* **Columns that are required for computation and analysis are selected. By this, the queries have run faster.**
* **The function “clean\_prepare\_data” is created, so that it can be applied to each of the fiscal year. This function is also used to remove duplicate rows**
* **The function count\_df is created to find the number of distinct records, so that it can be applied to each of the 3 fiscal years.**
* **The number of Nulls/NA’s vary from fiscal-year to fiscal-year and also in various columns.**
* **Hence separate filtering/cleaning conditions are applied for different queries. This way there is no loss of data.**
* **ggplots are used for visualization and analysis**

**Data Preparation and Cleaning:**

################################

# Loading the required libraries

################################

library(SparkR)

library(ggplot2)

library(gridExtra)

**# initiating the spark session**

sparkR.session(master = "local",appName="SparkR")

nyc\_parking\_2015 <- read.df("s3://pro-machinelearning-7/nycparkingtickets/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2015.csv",source = "csv", inferSchema = "true", header = "true",na.strings="")

nyc\_parking\_2016 <- read.df("s3://pro-machinelearning-7/nycparkingtickets/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2016.csv",source = "csv", inferSchema = "true", header = "true",na.strings="")

nyc\_parking\_2017 <- read.df("s3://pro-machinelearning-7/nycparkingtickets/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_2017.csv",source = "csv", inferSchema = "true", header = "true",na.strings="")

dim(nyc\_parking\_2015) # 11809233 , 51

dim(nyc\_parking\_2016) # 10626899 , 43

dim(nyc\_parking\_2017) # 10803028 , 43

printSchema(nyc\_parking\_2015)

printScehma(nyc\_parking\_2016)

printSchema(nyc\_parking\_2017)

clean\_prepare\_data <- function(df , year) {

**# subsetting only the following columns as they are only required for analysis:**

# "Summons Number",

# "Registration State"

# "Issue Date","Violation Code",

# "Vehicle Body Type",

# "Vehicle Make",

# "Violation Location",

# "Violation Precinct",

# "Issuer Precinct",

# "Violation Time",

# "House Number",

# "Street Name"

df <- select(df,"Summons Number","Registration State","Issue Date","Violation Code",

"Vehicle Body Type","Vehicle Make","Violation Location","Violation Precinct",

"Issuer Precinct","Violation Time","House Number","Street Name")

# Drop duplicate rows in the data-frame

df<-dropDuplicates(df)

# Converting the issue date to date type

df$`Issue Date` = to\_date(df$`Issue Date` , 'MM/dd/yyyy')

# creating an year column from the issue date

df <- mutate(df , Year = year(df$`Issue Date`))

# creating a month column from the issue date

df <- mutate(df , Month = month(df$`Issue Date`))

# We are considering the fiscal year: on the basis of it we are filterng the records

# For 2015 - Issue date should be in between 1 July, 2014 - 30th June 2015

# For 2016 - Issue date should be in between 1 July, 2015 - 30th June 2016

# For 2017 - Issue date should be in between 1 July, 2016 - 30th June 2017

df <- filter(df , (df$Month %in% c(7,8,9,10,11,12) & df$Year == year-1) |

(df$Month %in% c(1,2,3,4,5,6) & df$Year == year))

return(df)

}

nyc\_parking\_2015 <- clean\_prepare\_data(nyc\_parking\_2015,2015)

cache(nyc\_parking\_2015)

nyc\_parking\_2016 <- clean\_prepare\_data(nyc\_parking\_2016,2016)

cache(nyc\_parking\_2016)

nyc\_parking\_2017 <- clean\_prepare\_data(nyc\_parking\_2017,2017)

cache(nyc\_parking\_2017)

**#Function to check distinct rows and nulls in different columns**

count\_df <- function(df) {

# count of (distinct rows,Nulls in Violation Time,Nulls in House Number,Nulls in Street Name

# Nulls in Violation Code,Nulls in Violation Precinct,Nulls in Year, Nulls in Month,

# Nulls in Summons Number)

out <- numeric(9)

out[1] <- count(distinct(df))

out[2] <- count(where(df,isNull(df$`Violation Time`)))

out[3] <- count(where(df,isNull(df$`House Number`)))

out[4] <- count(where(df,isNull(df$`Street Name`)))

out[5] <- count(where(df,isNull(df$`Violation Code`)))

out[6] <- count(where(df,isNull(df$`Violation Precinct`)))

out[7] <- count(where(df,isNull(df$`Year`)))

out[8] <- count(where(df,isNull(df$`Month`)))

out[9] <- count(where(df,isNull(df$`Summons Number`)))

out

}

**#Output counts.Majority of Nulls are found in House Number, Street Name, Violation Code**

count\_df(nyc\_parking\_2015)

# 10598035 1438 1620679 5156 0 0 0 0 0

count\_df(nyc\_parking\_2016)

# 10396894 716 1962227 4482 0 1 0 0 0

count\_df(nyc\_parking\_2017)

# 10539563 53 2159447 3745 0 0 0 0 0

#############

**# Check for wrong entries in Year-column.Found no such records**

Issue\_date\_year<- function(df) {

agg(groupBy(df, df$Year), different\_years\_count = n(df$Year))

}

head(Issue\_date\_year(nyc\_parking\_2015))

# Year different\_years\_count

# 2015 5373971

# 2014 5224064

head(Issue\_date\_year(nyc\_parking\_2016))

# Year different\_years\_count

# 2015 5526176

# 2016 4870718

head(Issue\_date\_year(nyc\_parking\_2017))

# Year different\_years\_count

# 2016 5109661

# 2017 5429902

##############

**# Check for wrong entries in Month-column. Found no such records**

Issue\_date\_month<- function(df) {

agg(groupBy(df, df$Month), different\_months\_count = n(df$Month))

}

diff\_month\_2015<-Issue\_date\_month(nyc\_parking\_2015)

showDF(diff\_month\_2015)

# -----------------+

# |Month|different\_months\_count|

# +-----+----------------------+

# | 12| 671343|

# | 1| 777887|

# | 6| 1004087|

# | 3| 957743|

# | 5| 984991|

# | 9| 954744|

# | 4| 918253|

# | 8| 884733|

# | 7| 949486|

# | 10| 966232|

# | 11| 797526|

# | 2| 731010|

# +-----+----------------------+

diff\_month\_2016<-Issue\_date\_month(nyc\_parking\_2016)

showDF(diff\_month\_2016)

# |Month|different\_months\_count|

# +-----+----------------------+

# | 12| 767085|

# | 1| 814181|

# | 6| 427117|

# | 3| 1013888|

# | 5| 874469|

# | 9| 939355|

# | 4| 900709|

# | 8| 902634|

# | 7| 884785|

# | 10| 1096952|

# | 11| 935365|

# | 2| 840354|

# +-----+----------------------+

diff\_month\_2017<-Issue\_date\_month(nyc\_parking\_2017)

showDF(diff\_month\_2017)

# |Month|different\_months\_count|

# +-----+----------------------+

# | 12| 778704|

# | 1| 877365|

# | 6| 852187|

# | 3| 964737|

# | 5| 1020244|

# | 9| 960537|

# | 4| 888402|

# | 8| 801258|

# | 7| 700475|

# | 10| 969330|

# | 11| 899357|

# | 2| 826967|

# +-----+----------------------+

#######################################################################################

**Questions:**

**Examine the data**

1. Find total number of tickets for each year.

|  |  |
| --- | --- |
| Year | Total number of tickets |
| 2015 | 10598035 |
| 2016 | 10396894 |
| 2017 | 10539563 |

1. Find out how many unique states the cars which got parking tickets came from.

|  |  |
| --- | --- |
| Year | Number of states |
| 2015 | 69 |
| 2016 | 68 |
| 2017 | 67 |

1. Some parking tickets don’t have addresses on them, which is cause for concern. Find out how many such tickets there are.

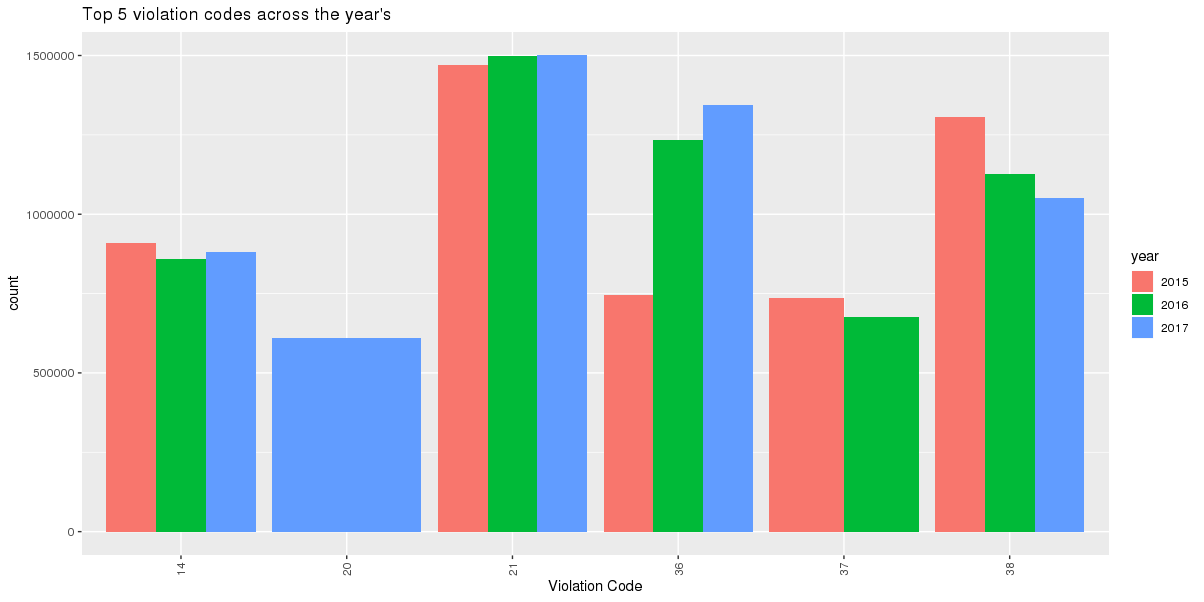
|  |  |
| --- | --- |
| Year | Number of tickets with no addresses |
| 2015 | 10598035 |
| 2016 | 10396894 |
| 2017 | 2160639 |

**Aggregate tasks:**

1. How often does each violation code occur? (frequency of violation codes - find the top 5)

**Answer - Plot:** Following are the top 5 violation codes across the 3 years. We could see violation

* (14, 21,36,38) – violation code has been in top5 across all the 3 years
* code ‘20’ is in top 5 only on 2017 and
* ‘37’ is only on top 5 on year’s 2015 & 2016.
* ‘21’ violation code has been the maximum & nearly constant across all 3 years.
* ‘36’ violation code has been continuously increasing from the year 2015 to 2017
* ‘38’ violation code has been continuously decreasing from the year 2015 to 2017

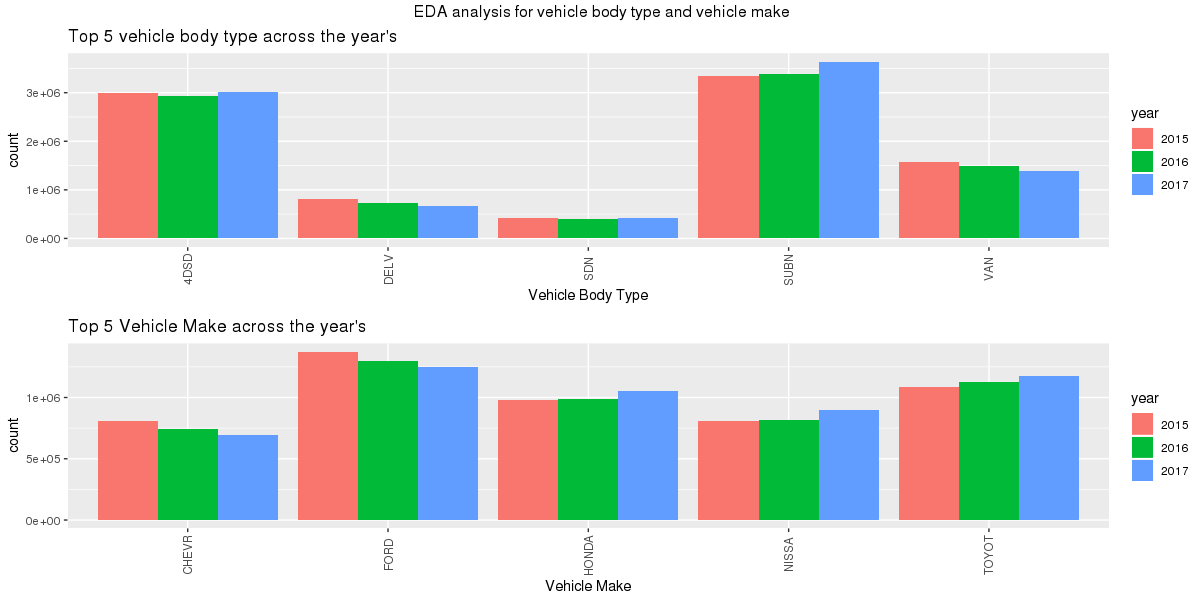


1. How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)

**Answer -** **Plot:**

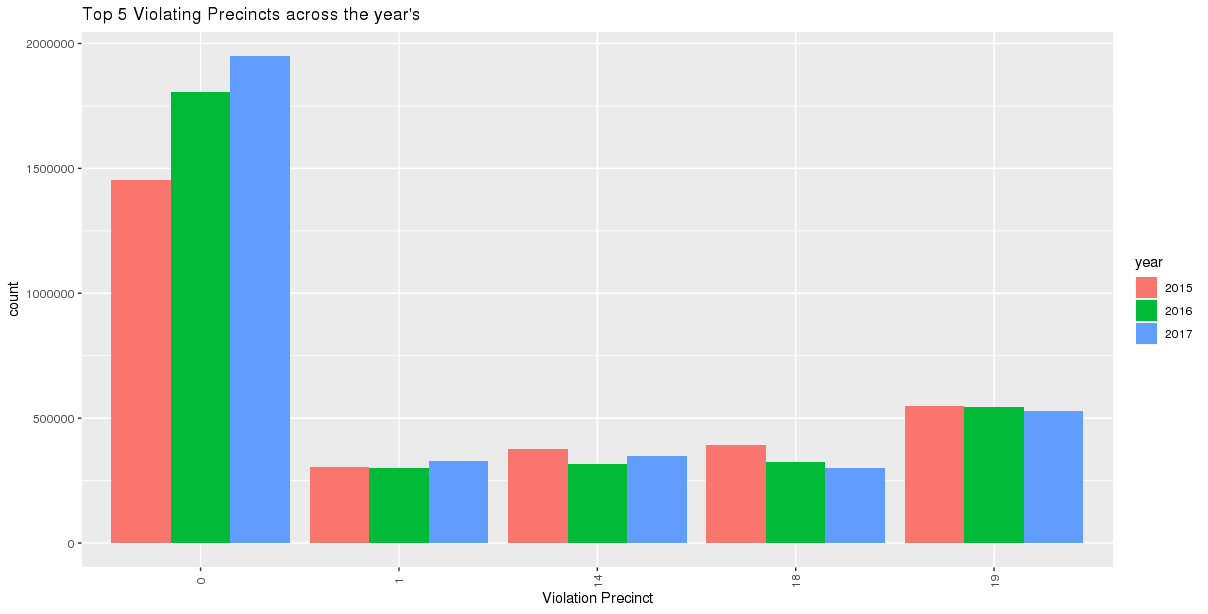
**Vehicle Body Type and Vehicle Make –**

* Vehicle Body Type – We could see the ‘SUBN’, ‘4DSD’, ‘VAN’, ‘DELV’ & ‘SDN’ are the top 5 vehicle body types got more parking tickets across the 3 years.
* Vehicle Make – We could see the ‘FORD’,‘TOYOT’, ‘HONDA’, ‘NISSA’ & ‘CHEVR’ are the top 5 vehicle makes got more parking tickets across the 3 years.



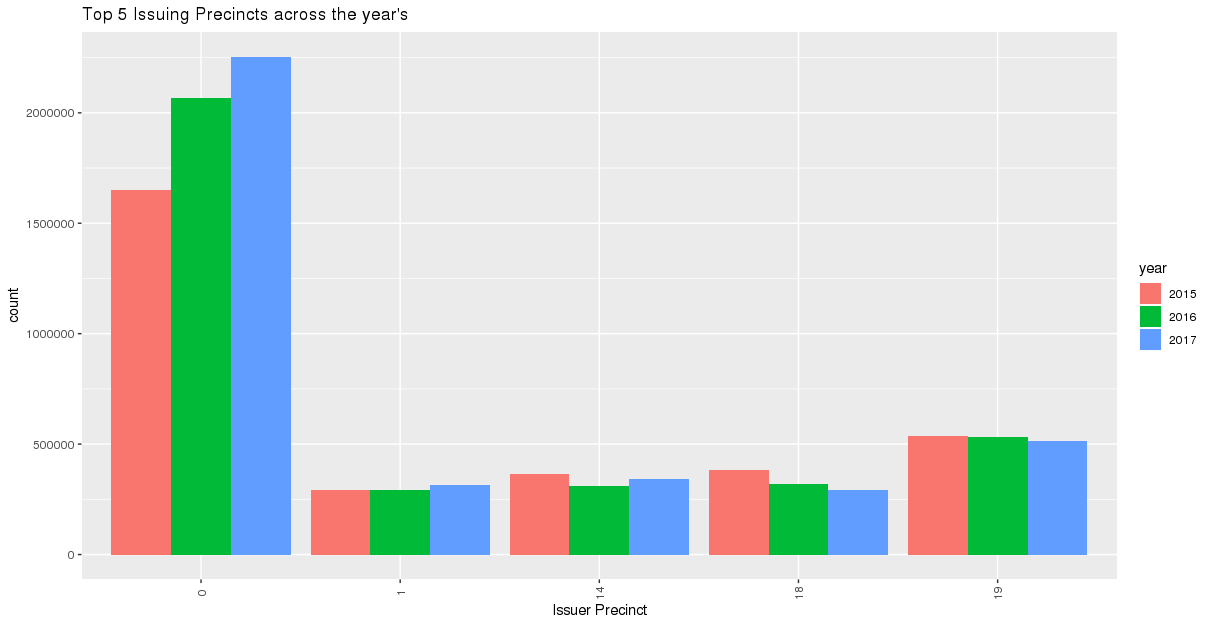
1. A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of:
   1. Violating Precincts (this is the precinct of the zone where the violation occurred)

**Answer -** **Plot:** The top 5 precincts which have more violations are ‘0’, ‘19’, ‘18’, ‘14’ & ‘1’ across the 3 years.



* 1. Issuing Precincts (this is the precinct that issued the ticket)

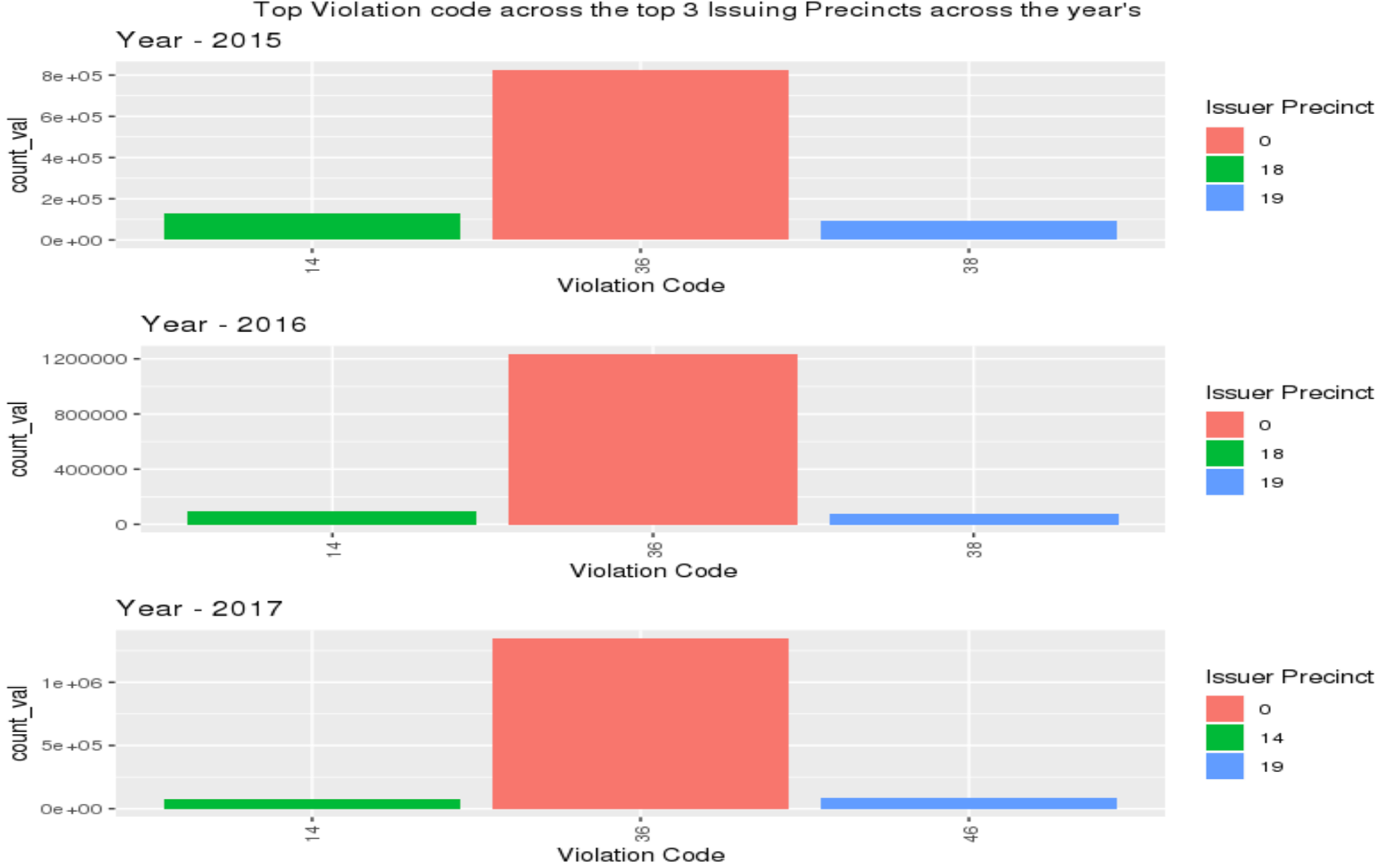
**Answer -** **Plot:** The top 5 issuing precincts which have more violations are ‘0’, ‘19’, ‘18’, ‘14’ & ‘1’ across the 3 years.



1. Find the violation code frequency across 3 precincts which have issued the most number of tickets - do these precinct zones have an exceptionally high frequency of certain violation codes? Are these codes common across precincts?

**Answer -** **Plot:**

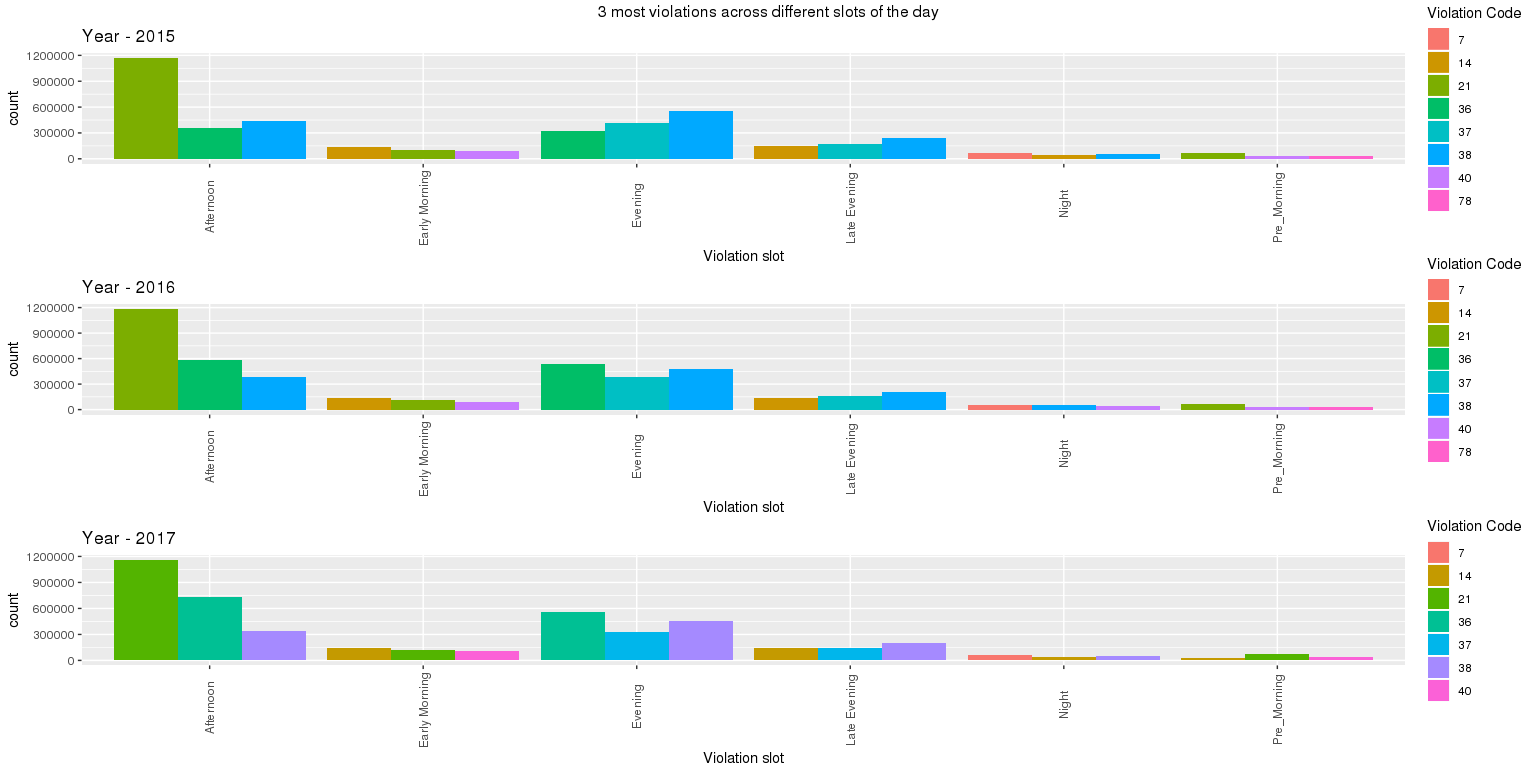
* We could see that violation codes ‘14’, ‘36’ & ‘38’ for year 2015 and 2016
* Violation codes ‘14’, ‘36’ & ‘46’.
* It is clear from the below plot that not all the high frequency violation codes are common across precincts and years.
* But one thing is clear that ‘36’ violation is common across the year 2015 , 2016 and 2017 and ‘0’ is the issuer precinct for it.



1. You’d want to find out the properties of parking violations across different times of the day:
   1. The Violation Time field is specified in a strange format. Find a way to make this into a time attribute that you can use to divide into groups.
   2. Find a way to deal with missing values, if any.
   3. Divide 24 hours into 6 equal discrete bins of time. The intervals you choose are at your discretion. For each of these groups, find the 3 most commonly occurring violations
   4. Now, try another direction. For the 3 most commonly occurring violation codes, find the most common times of day (in terms of the bins from the previous part)

5c. Following graph shows the 3 most common violations across the different slots of the day

* ‘21’ violation code happens most in the afternoon slot of the day and this is consistently happening across all the years 2015 , 2016 and 2017
* ‘21’ , ‘36’ and ‘38’ violation code has happened in afternoon slot of the day and this is consistently happening across all the years 2015 , 2016 and 2017
* ‘21’ violation code happens in ‘Afternoon’ , Early Morning and Pre Morning slots of the day across all the years 2015 , 2016 and 2017.

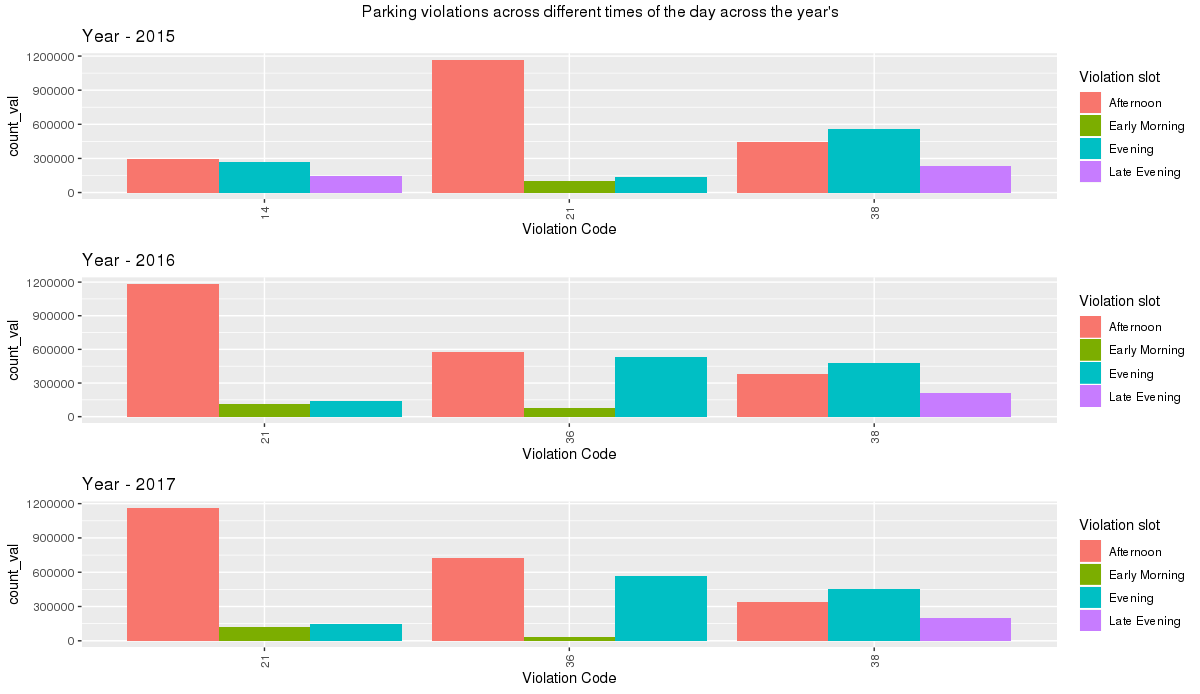


5d. For the 3 most commonly occurring violation codes, find the most common times of day (in terms of

the bins from the previous part)

**Answer -** **Plot:** From the below plots we can see the below:

* Violation code ‘21’ occurs more during the afternoon time slot
* Lowest number of violations occur during Early Morning and Late Evening time slot



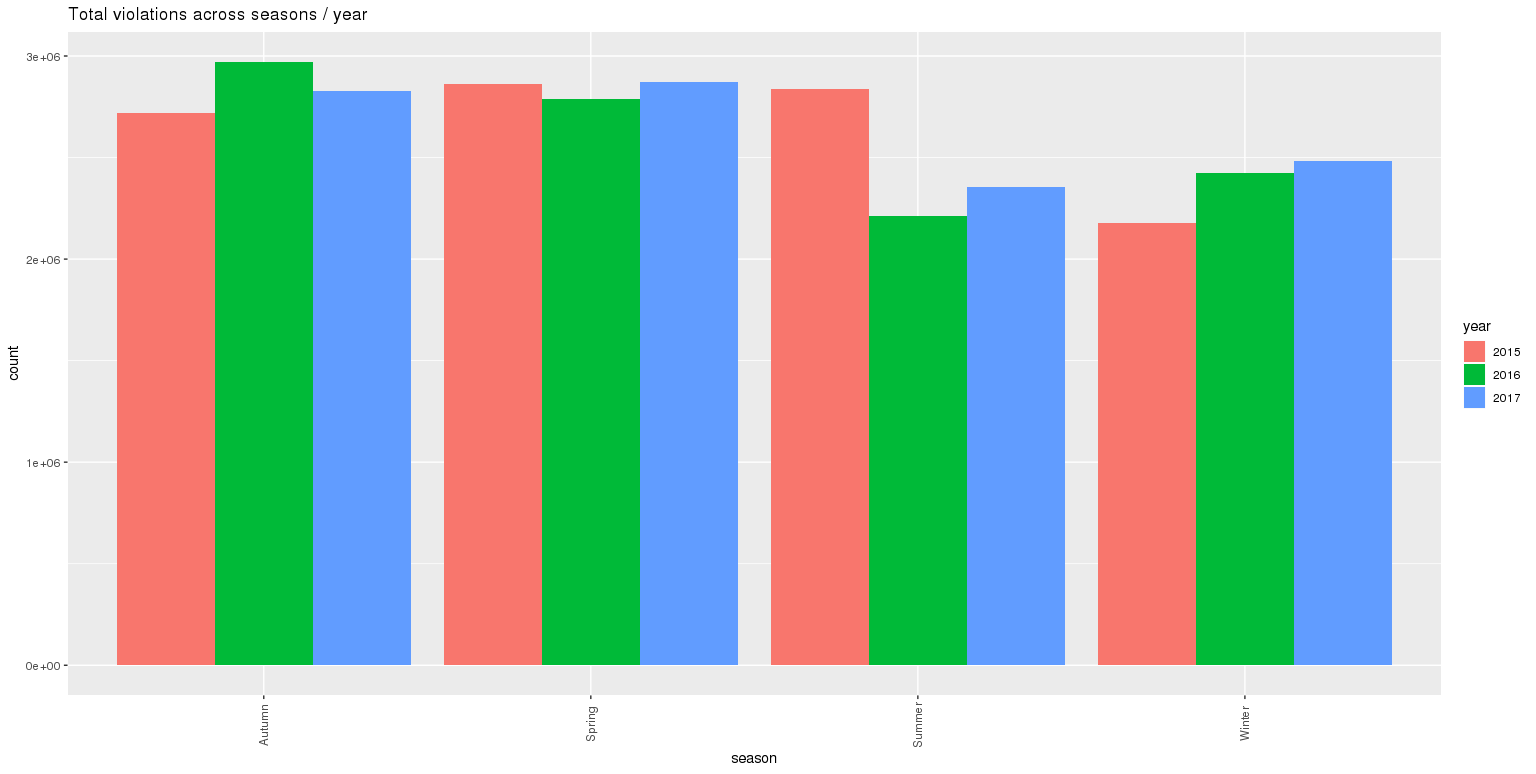
1. Let’s try and find some seasonality in this data

* First, divide the year into some number of seasons, and find frequencies of tickets for each season.
* Then, find the 3 most common violations for each of these season

6a – plot

**Answer –** **Plot**

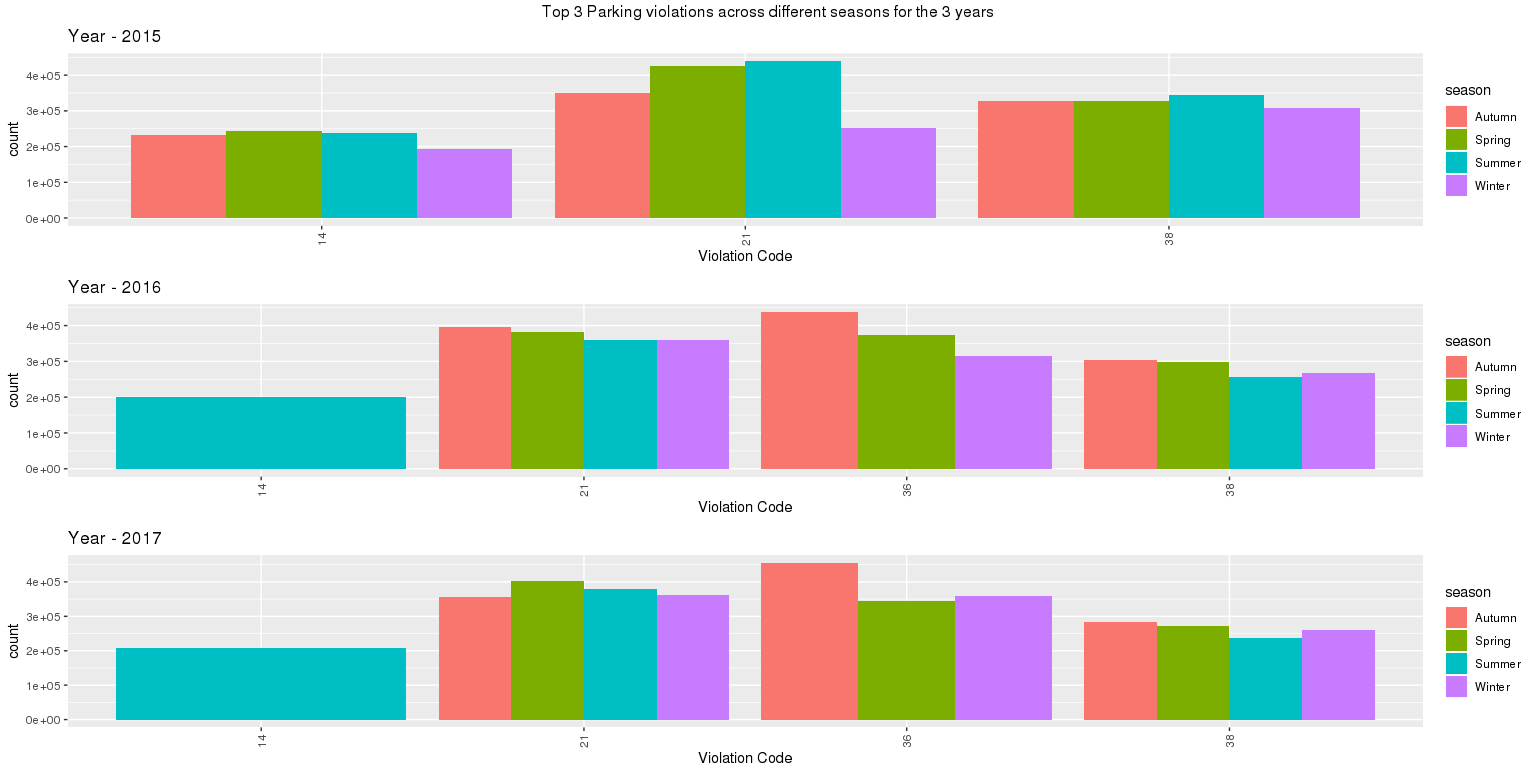
* **For the year 2016 and 2017 violation have gone in the summer season compared to the other seasons.**
* **Overall violations is lowest in winter season.**
* **Overall violations is highest in spring season & autumn season**



6b.

**Answer -** **Plot:** From the plot, we could see that

* Violation codes ‘21’ and ‘38’ is common across all the seasons and all the 3 years.
* However violation codes ‘14’ and ‘36’ had different behavior across seasons and the years.



1. The fines collected from all the parking violation constitute a revenue source for the NYC police department. Let’s take an example of estimating that for the 3 most commonly occurring codes.

* Find total occurrences of the 3 most common violation codes
* Then, search the internet for NYC parking violation code fines. You will find a website (on the nyc.gov URL) that lists these fines. They’re divided into two categories, one for the highest-density locations of the city, the other for the rest of the city. For simplicity, take an average of the two.
* Using this information, find the total amount collected for all of the fines. State the code which has the highest total collection.
* What can you intuitively infer from these findings?

**Answer:**

Following are average fine identified for the top 3 most common violation codes from the NYC website.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year** | **Violation code** | **Definition** | **Manhattan**  **96th St. & below** | **All Other Areas** | **Average** |
| 2015 | 21 | Street Cleaning - No parking where parking is not allowed by sign, street marking or traffic control device. | $65 | $45 | $55 |
| 38 | Muni Meter - Failing to show a receipt or tag in the windshield. Drivers get a 5-minute grace period past the expired time on Muni-Meter receipts. | $65 | $35 | $50 |
| 14 | General No Standing: Standing or parking where standing is not allowed by sign, street marking or; traffic control device. | $115 | $115 | $115 |
| 2016 | 21 | Street Cleaning - No parking where parking is not allowed by sign, street marking or traffic control device. | $65 | $45 | $55 |
| 36 | Exceeding the posted speed limit in or near a designated school zone. | $50 | $50 | $50 |
| 38 | Muni Meter - Failing to show a receipt or tag in the windshield. Drivers get a 5-minute grace period past the expired time on Muni-Meter receipts. | $65 | $35 | $50 |
| 2017 | 21 | Street Cleaning - No parking where parking is not allowed by sign, street marking or traffic control device. | $65 | $45 | $55 |
| 36 | Exceeding the posted speed limit in or near a designated school zone. | $50 | $50 | $50 |
| 38 | Muni Meter - Failing to show a receipt or tag in the windshield. Drivers get a 5-minute grace period past the expired time on Muni-Meter receipts. | $65 | $35 | $50 |

**Answer -** **Plot:** From the plot,

* we could see violation code ‘14’ on year 2015 had the highest fine collection and this code this not in the top 3 for 2016 and 2017.
* Also we could see violation code ‘21’ the common violation code which provides the next highest fine collection and this violation code is common across all the 3 years.

