




3/10/2019

NYC Parking Tickets

An Exploratory Analysis

- 
1. Pritha Banerjee
 2. Priyanka Kapoor
 3. Piyush Baid
 4. Rajarshi Ghoshal

1. INTRODUCTION

New York City is a thriving metropolis. Just like most other metros that size, one of the biggest problems its citizens face is parking. The classic combination of a huge number of cars and cramped geography is the exact recipe that leads to a huge number of parking tickets.

In an attempt to scientifically analyze this phenomenon, the NYC Police Department has collected data for parking tickets. Out of these, the data files for multiple years are publicly available on Kaggle. We will try and perform some exploratory analysis on a part of this data. Spark will allow us to analyze the full files at high speeds, as opposed to taking a series of random samples that will approximate the population. For the scope of this analysis, we wish to analyze the parking tickets over the year 2017. The purpose of this case study is to conduct an exploratory data analysis that helps you understand the data.

1.1 Data

The data structure understanding is taken from: <https://www.kaggle.com/new-york-city/nyc-parking-tickets/data> .

For our analysis we will be considering the dataset of fiscal year 2017. The data for this case study has been placed in HDFS at the following path:-

```
'/common_folder/nyc_parking/Parking_Violations_Issued_-_Fiscal_Year_2017.csv'
```

1.2 Methodology

We have done our analysis in 3 steps:

1. Understanding the data
2. Cleaning the data
3. Examining the data
4. Aggregation Task

2. UNDERSTANDING THE DATA

Dimensions of dataset:

- There are 10 variables and 10803028 rows in the dataset.
- Details of the 10 variables are as follows:
 - Summons Number: num
 - Plate ID: chr
 - Registration State: chr
 - Issue Date: POSIXct
 - Violation Code: int
 - Vehicle Body Type: chr
 - Vehicle Make: chr
 - Violation Precinct: int
 - Issuer Precinct: int
 - Violation Time: chr

3. CLEANING THE DATA

1. Renaming the columns: The column names contain white space. For the ease of our analysis, we have replaced white spaces with “_”.
2. Checking Null Values: In order to facilitate smooth analysis, we checked for the null values. However, there were no null values present in the dataset.
3. Checking the range of Issue Date: Since we only need data from year 2017 for our analysis, we checked for non-conformities in the dataset and removed them.

year(Issue_Date)	count
2017	5431918
2016	5368391
2018	1057
2019	472
2015	419
2000	185

4. EXAMINING THE DATA

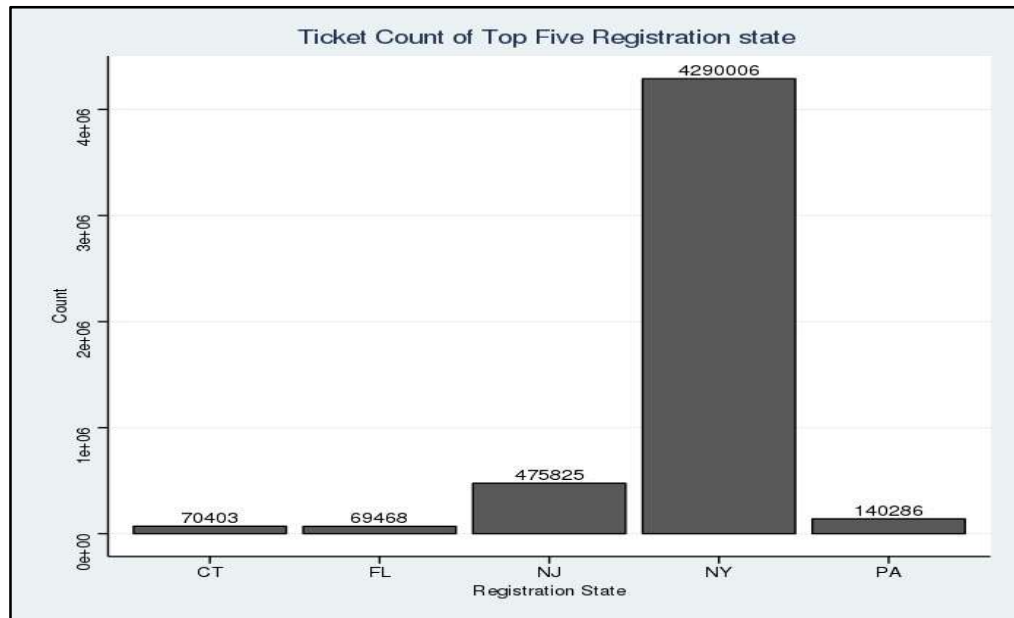
1. Find the total number of tickets for the year 2017
We did EDA on Issue_date and found the number of tickets for the year 2017 were **5431918**.
2. Find out the number of unique states from where the cars that got parking tickets came from.
 - There is a numeric entry '99' in the column which should be corrected. We strategized that we will replace it with the state having maximum entries.
 - The total number of states are **65**.
 - We found the top 6 states where maximum tickets are issued.

Registration State	count
NY	4273951
NJ	475825
PA	140286
CT	70403
FL	69468
IN	45525

- Since NY State has the maximum count, we replaced the erroneous entry '99' with 'NY'.
- Therefore, we got total **64** unique states. NY is the state where maximum tickets have been issued.

Registration State	count
NY	4290006
NJ	475825
PA	140286
CT	70403
FL	69468
IN	45525

- Plot for the top 5 states where the maximum tickets were issued.



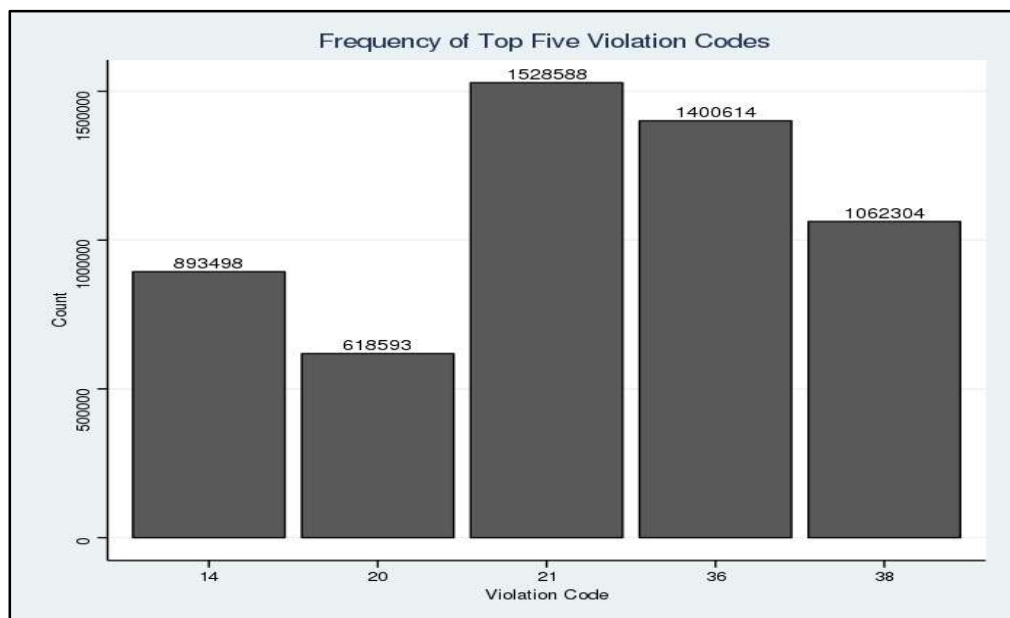
5. AGGREGATION TASKS

- How often does each violation code occur? Display the frequency of the top five violation codes.

- Results

Violation_Code	count
21	768087
36	662765
38	542079
14	476664
20	319646

- Plot



- Violation code 21 occurs the maximum number of times.

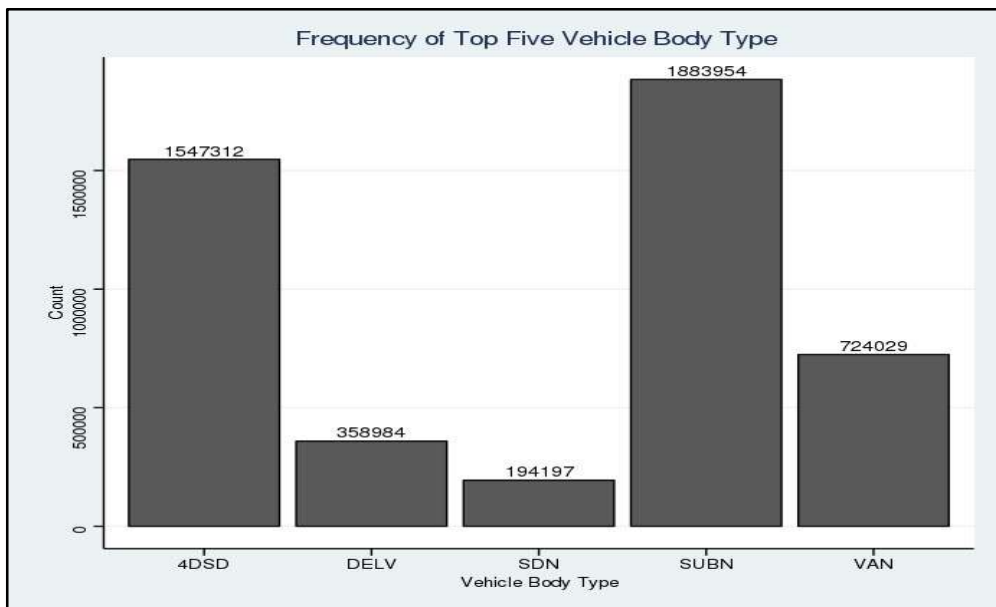
2. How often does each 'vehicle body type' get a parking ticket? How about the 'vehicle make'?

Vehicle body type

- Results

Vehicle_Body_Type	count
SUBN	1883954
4DSD	1547312
VAN	724029
DELV	358984
SDN	194197

- Plot



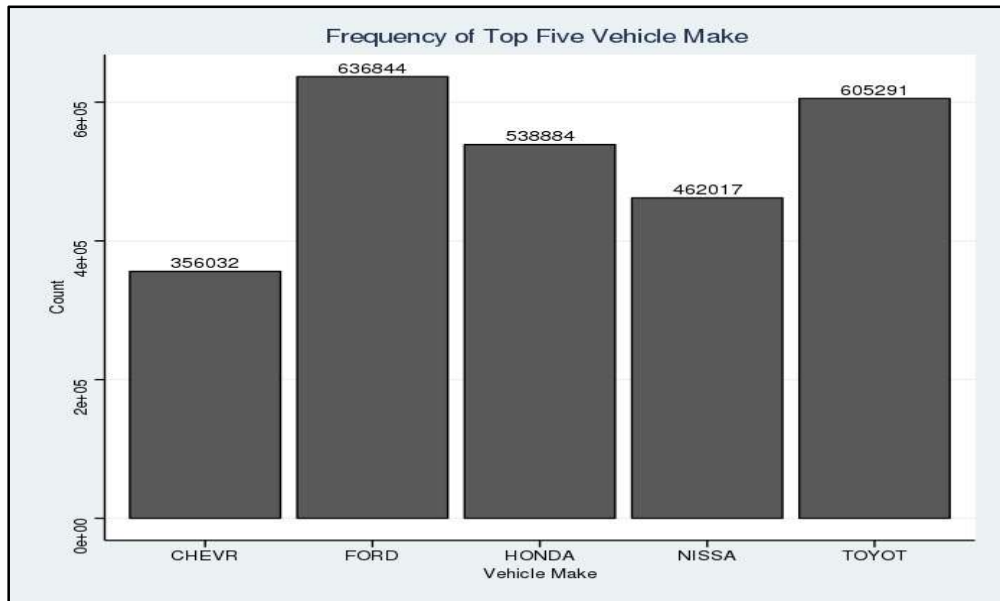
- SUBN vehicle body type is the one that gets the most parking tickets.

Vehicle Make

- Results

Vehicle_Make	count
FORD	636844
TOYOT	605291
HONDA	538884
NISSA	462017
CHEVR	356032

- Plot



3. A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequency of tickets for each of the following:

Violation Precinct (this is the precinct of the zone where the violation occurred).

Using this, can you make any insights for parking violations in any specific areas of the city?

Issuer Precinct (this is the precinct that issued the ticket). Here you would have noticed that the dataframe has 'Violating Precinct' or 'Issuing Precinct' as '0'. These are the erroneous entries. Hence, provide the record for five correct precincts. (Hint: Print top six entries after sorting)

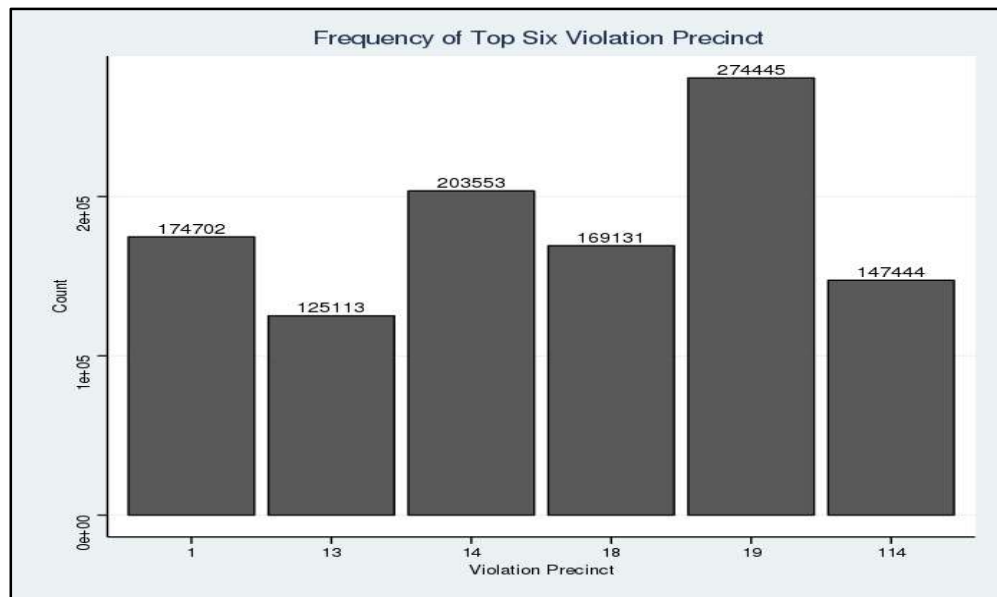
Violation Precinct

- Deleting rows with '0' as entry since these are erroneous entries. 925596 rows are removed for analysis.
- Results: Most violation occurred in Zone 19 followed by zone 14. Least frequent violation zones are 183,126,673,918,613 and 806.

Violation_Precinct	count
19	274445
14	203553
1	174702
18	169131
114	147444
13	125113

Violation_Precinct	count
183	1
126	1
673	1
918	1
613	1
806	1

- Plot

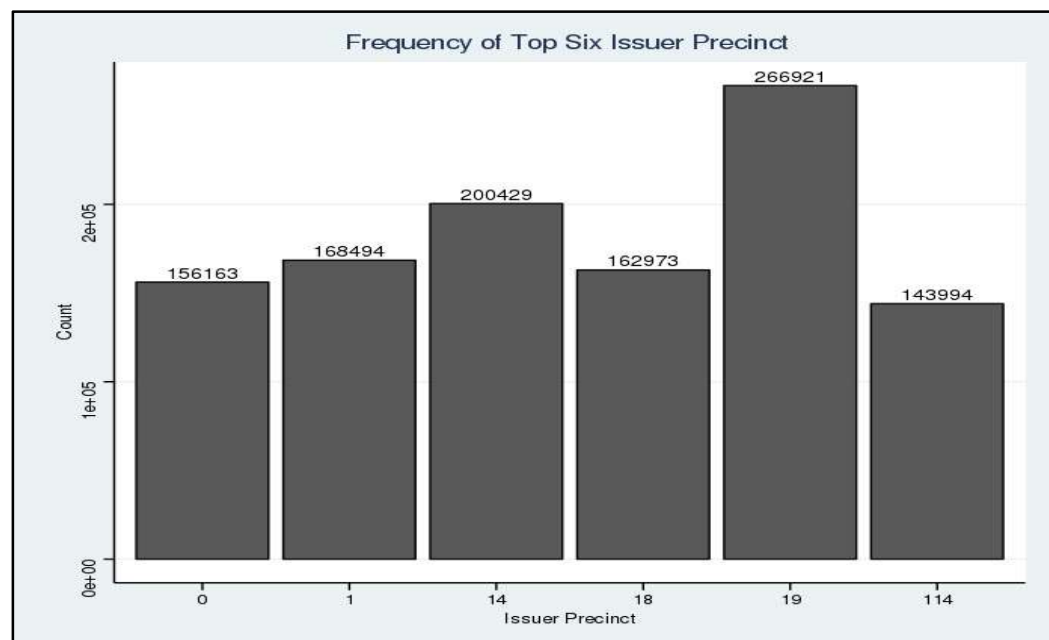


Issuer Precinct

- Deleting rows with '0' as entry since these are erroneous entries. **15613** rows are removed for analysis.
- Results: Most violation occurred in Zone 19 followed by zone 14.

Violation_Precinct	count
19	266921
14	200429
1	168494
18	162973
114	143994

- Plot



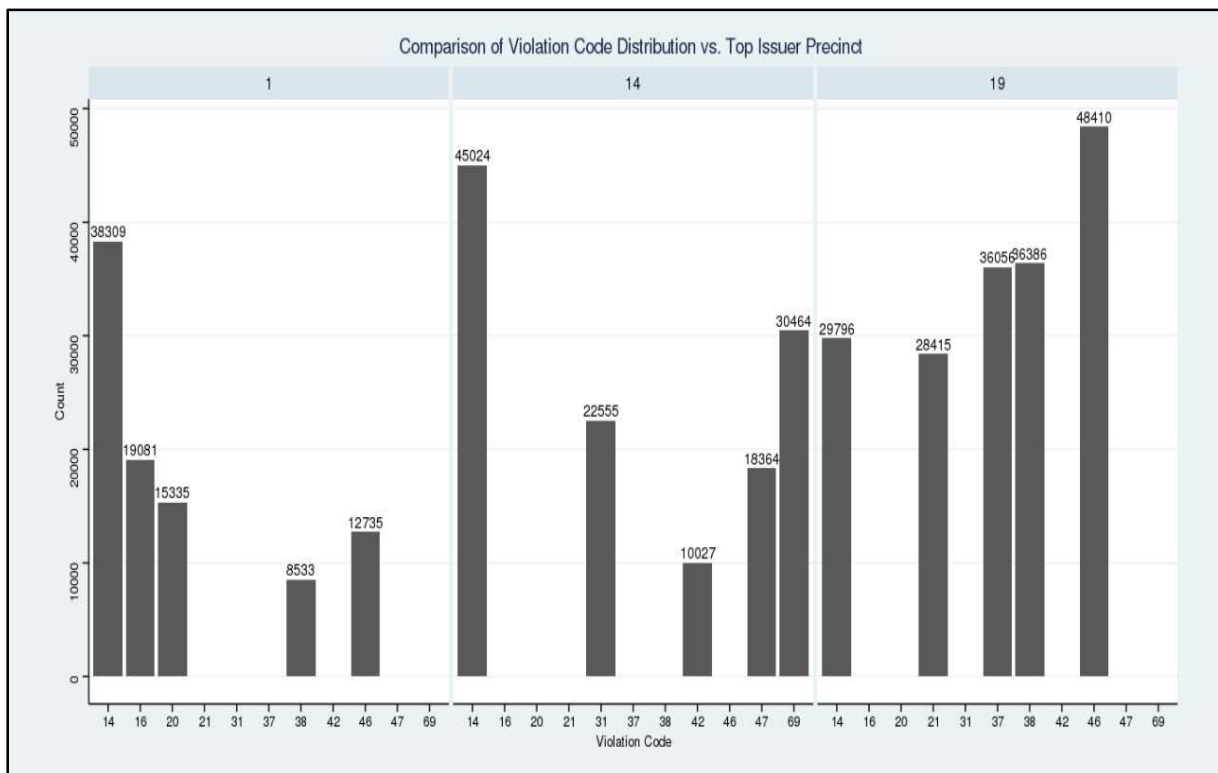
- So, it can be concluded that violation occurred and issued in the same zones mostly

4. Find the violation code frequency across three 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?

- The most number of tickets are issued in (Top3) precincts 19, 14 and 1. Hence, we found the violation codes for these three precincts:

Violation_Code	Count_of_Tickets	Issuer_Precinct
46	48410	19
38	36386	19
37	36056	19
14	29796	19
21	28415	19
14	45024	14
69	30464	14
31	22555	14
47	18364	14
42	10027	14
14	38309	1
16	19081	1
20	15335	1
46	12735	1
38	8533	1

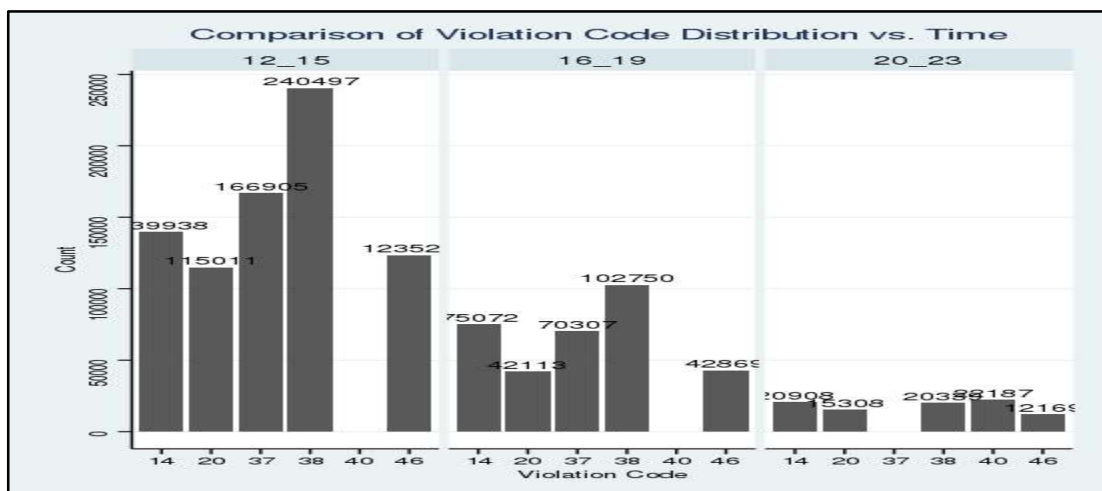
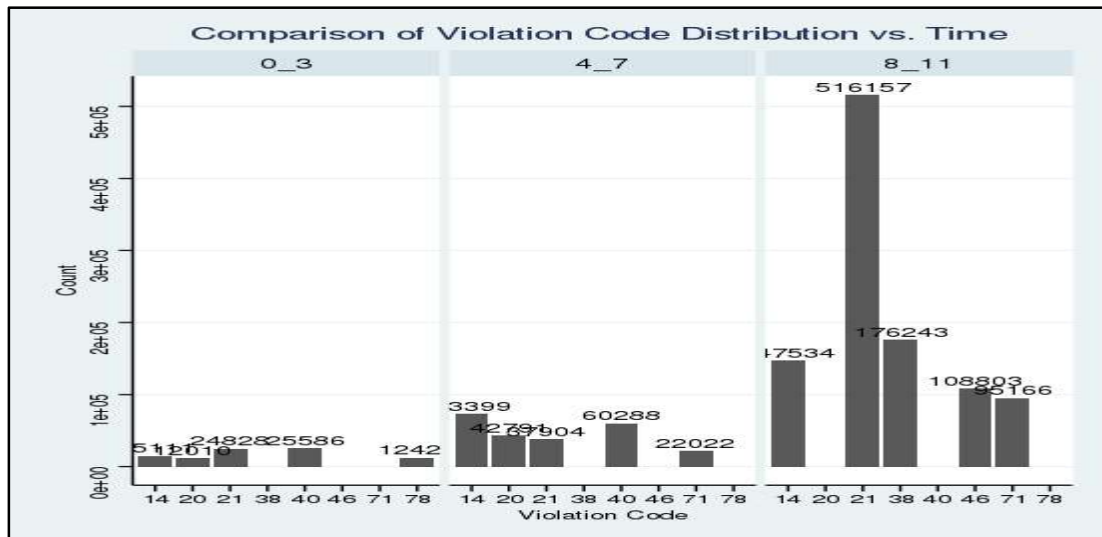
- Plot



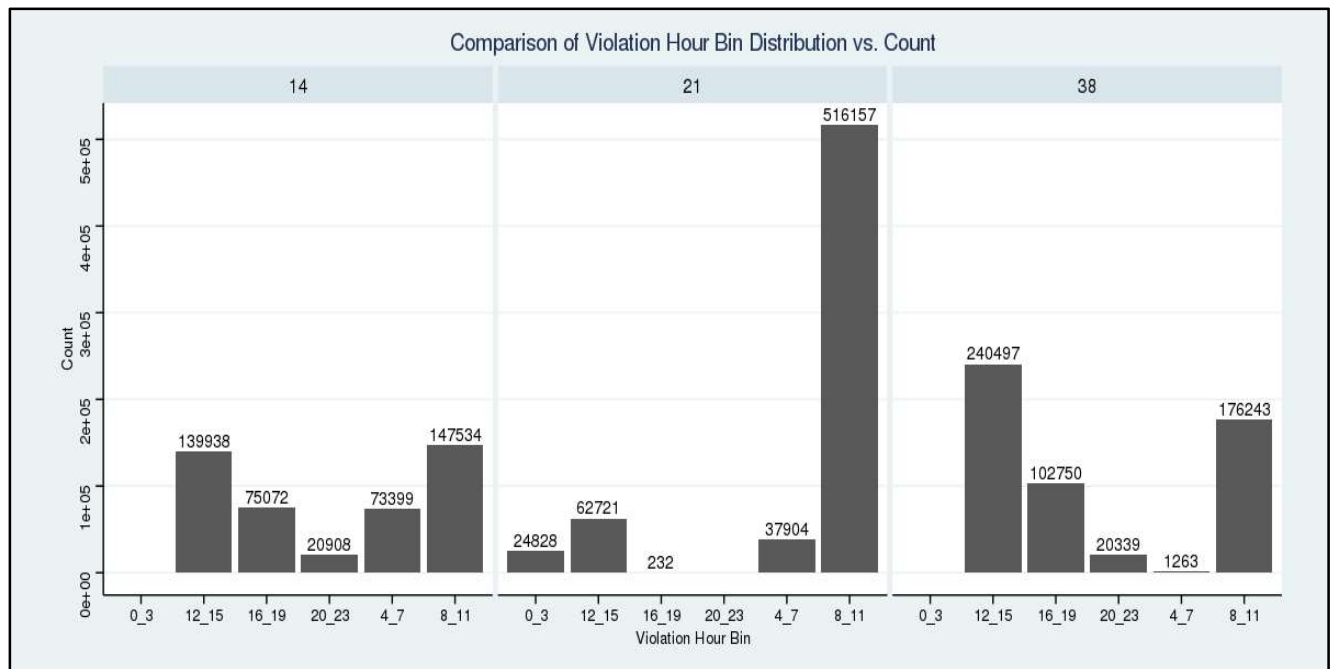
- Issuer_Precinct- 19 has an exceptionally high frequency of Violations ($48410+36386+36056=120852$)
- Violation Code- 14 is the highest ($29796+45024+38309=113129$) and also the common among the Issuer Precincts having the highest frequency of Violation.

5. You'd want to find out the properties of parking violations across different times of the day:

- Find a way to deal with missing values, if any.
 - 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.
 - Divide 24 hours into six equal discrete bins of time. The intervals you choose are at your discretion. For each of these groups, find the three most commonly occurring violations.
 - Now, try another direction. For the three most commonly occurring violation codes, find the most common time of the day (in terms of the bins from the previous part)
- Converting the **Violation Time** into correct timestamp format:
 - Adding a column with constant "M" values
 - Joining the 2 columns "Violation_Time" & "M" to complete the Meridian(AM/PM) part of the column.
 - Extracting Violation Hour, Violation Minute and Part of Day.
 - We've observed that there are records that have both 00xxAM as well as 12xxAM. Therefore we will replace all 00xxAM with 12xxAM.
 - Removing erroneous values for Violation Hour column (greater than 12) and Violation Minute column (greater than 59) and assigning NA to those redundant values.
 - Concatenating the components (Violation Hour, Violation Minute and Part of Day) into a standardized Violation Time and converting into a TimeStamp format.
 - Dividing 24 hours into six equal discrete bins of time (**0 to 3, 4 to 7, 8 to 11, 12 to 15, 16 to 19, 20 to 23**) and finding the three most commonly occurring violations for the same.



- **Top 3 Violation Codes found were 21, 38 and 14.** The most common time of the day for the respective Violation Codes are calculated.



6. Let's try and find some seasonality in this data

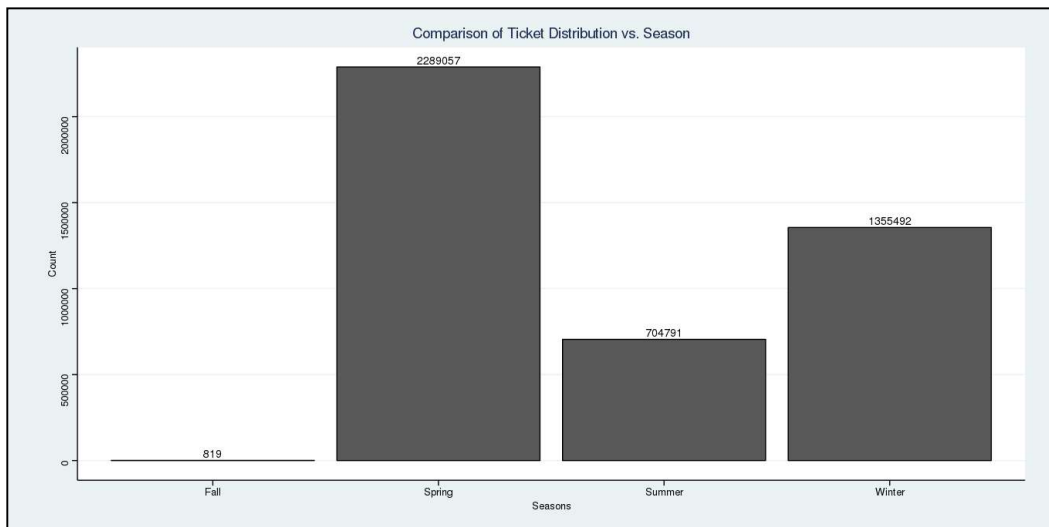
- First, divide the year into some number of seasons, and find frequencies of tickets for each season. (Hint: Use Issue Date to segregate into seasons)
- Then, find the three most common violations for each of these seasons.

- **Seasons of the year:**

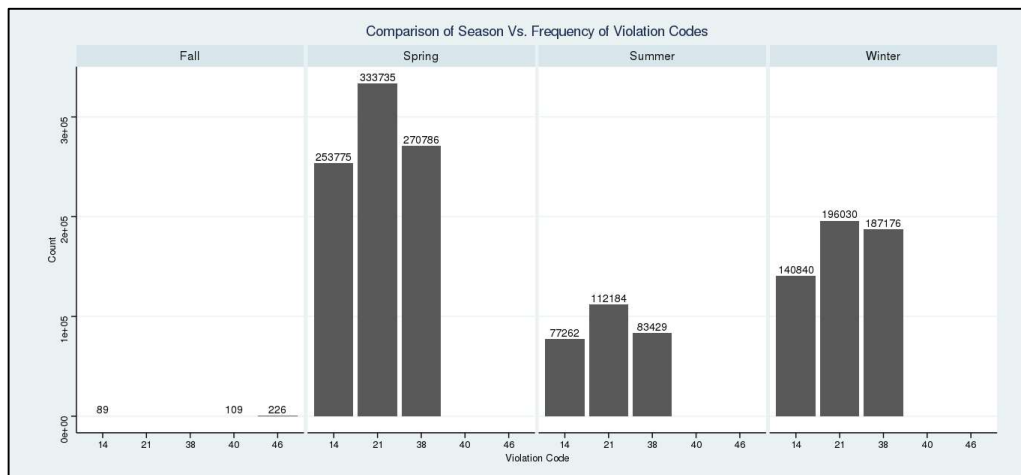
- Dividing the year into 4 seasons: **Winter, Spring, Summer, Fall.**
- Frequencies of tickets for each Season is calculated.

Seasons	Number of Tickets
Spring	2289057
Winter	1355492
Summer	704791
Fall	819

- Plot



- Three most common violations for each of these seasons are calculated.
Maximum violation is registered in **Spring season** for **Violation Code 21**



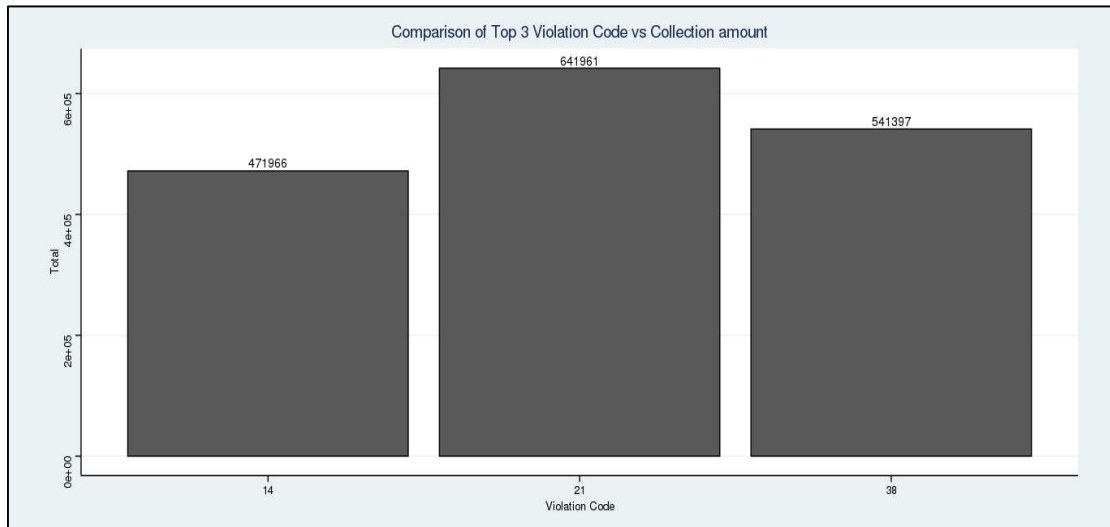
- The fines collected from all the parking violation constitute a revenue source for the NYC police department. Let's take an example of estimating that for the three most commonly occurring codes.
 - Find total occurrences of the three most common violation codes
 - Then, visit the website: <http://www1.nyc.gov/site/finance/vehicles/services-violation-codes.page>. It lists the fines associated with different violation codes. 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 the three violation codes with maximum tickets. State the code which has the highest total collection.
 - What can you intuitively infer from these findings?

Part A: Total occurrences of the three most common violation codes

- Result

Violation Code	Number of Tickets
21	641961
38	541397
14	471966

- Plot

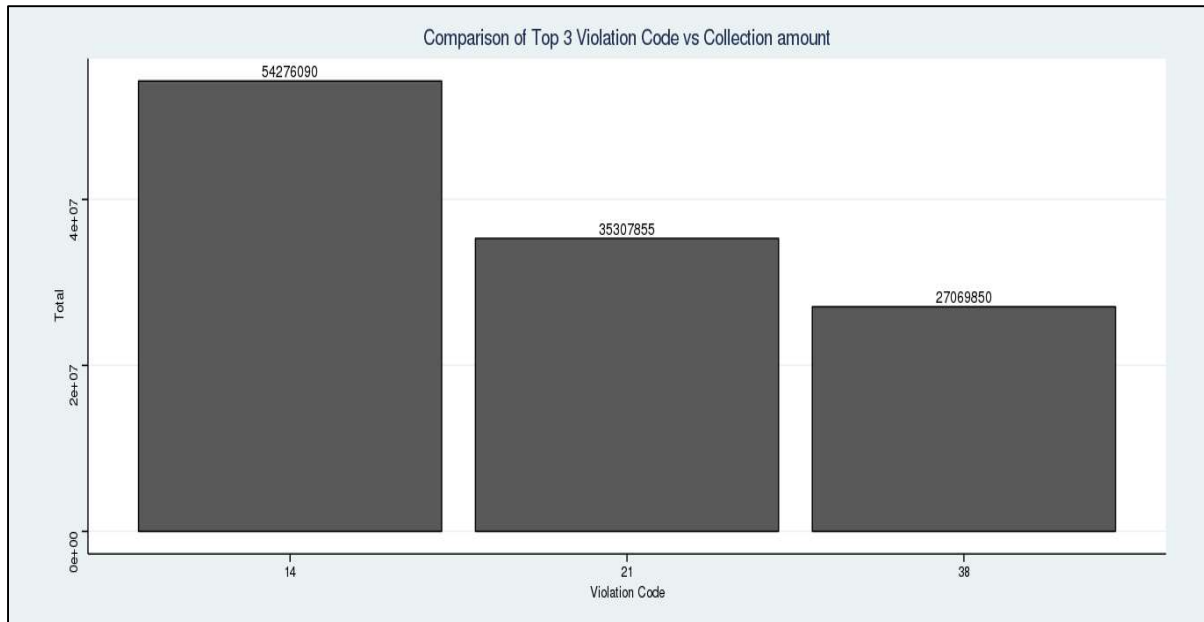


Part B: find the total amount collected for the three violation codes with maximum tickets. State the code which has the highest total collection

- Result

Violation Code	Average Fine amount (in \$)	Total Collection
21	55	35307855
38	50	27069850
14	115	54276090
Total Collection (in \$)	116653795	

- Plot



- **Insight:** The Violation code **14** has the highest total collection even though it has the lowest frequency in the top 3 violation codes that got the maximum number of tickets. This could be because of high average fine amount of \$115 for Violation code 14. After Violation code 14, 21 has the highest total collection which has the second highest average amount of \$55.