MSCI 6110: Big Data Management and Analytics
Midpoint Project Report
Group 2
Aarron Lebow, Jinghui Li, Tom Maples, Santhosh Raj Murugesan

Summary

Every year, millions of parking citations are issued in New York City. The data collected from each ticket could contribute to valuable information and knowledge gained. Trends in years and months, clusters of vehicle types, locations, issuing precincts, and violation descriptions are just a few of the many facets. Potential stakeholders include the offices of New York City—budgeting, finance, department of transportation, police—auto manufacturers, rideshare services, and potentially a searchable public database educating people on parking trends.

Data

The dataset that we have used for NYC parking violation analysis consists of approximately 42 million observations from New York City from August 2013 - June 2017. The data has been collected from New York City Department of Finance and made publicly available on opendata.cityofnewyork.us. Each observation comprises of 51 attributes pertaining to each individual ticket. All attributes are listed in the appendix

Methods

A temporary table was defined, and data from fiscal years 2014 – 2017 was loaded from the hive data file system. In order to reduce computing costs that stem from the size of the data, the analysis was refined to 17 features: summons number, plater ID, violation code, violation location, violation precinct, issuer precinct, issuer command, issuer squad, street name, vehicle color, vehicle make, vehicle body type, vehicle year, violation description, year, month, day, and hour. These features were then used to populate a more manageable pivot table by query. This table was dynamically partitioned by year and month.

Challenges

Some of the challenges that we faced were converting the date and time to required format and storing it in our hive tables. For example, the time was in the format of 0212p, 0124a. We had to strip off ‘a’ or ‘p’ and covert it to a 24-hour format and use it for analytics. And the date was in the format of mm/dd/yyyy but hive requires the date be in the format yyyy-mm-dd. So, we had run some scripts on the data to convert the date into required format. We are also planning to find some fine details such as month number, day of week, if it’s a weekend or not and stored it in the tables so we will be able to predict how many violations happen during the weekends and weekdays. We have also which month of

the year and the years the parking violations are at its peak. The violation description was missing in many observations, but we populated it using information obtained from the Department of Finance Website

https://www1.nyc.gov/site/finance/vehicles/services-violation-codes.page

Analysis

Registration State

The data was grouped by Registration State and the number of tickets issued per state was counted and divided by the total number of tickets. The majority of tickets were issued to the vehicles registered in New York, followed by New Jersey and Pennsylvania.

NY	73%
NJ	8%
PA	3%

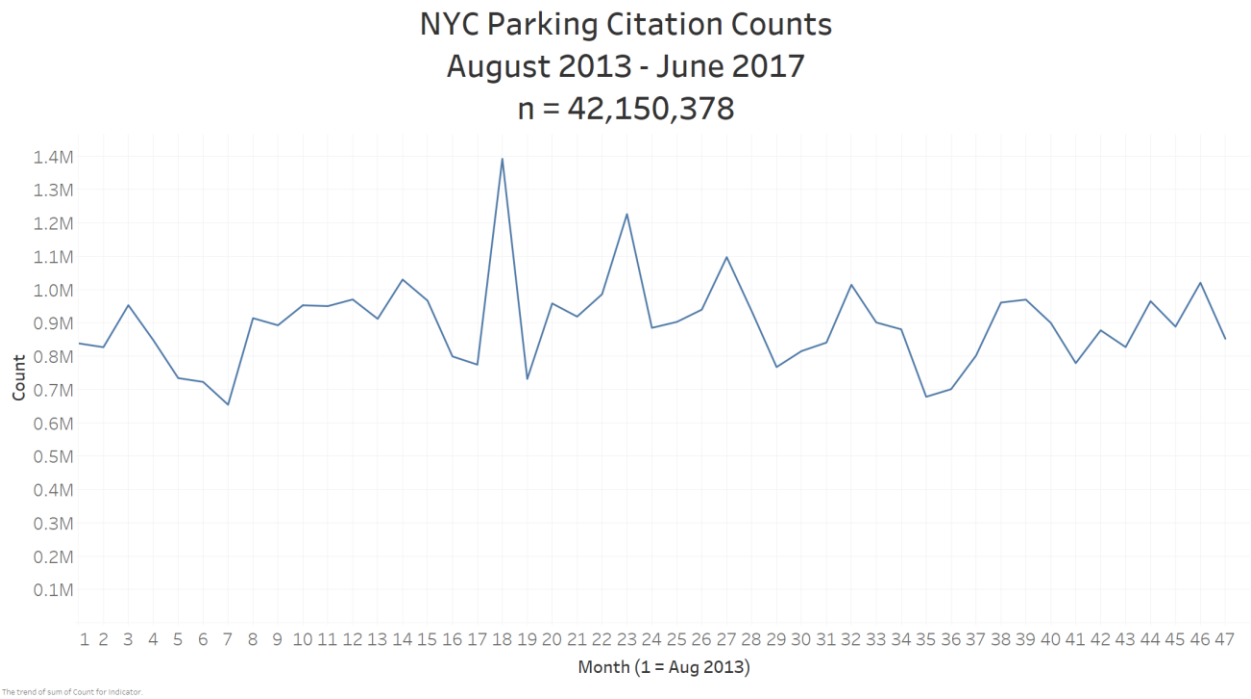
Plate Type

The data was grouped by plate type and the number of tickets issued was counted for each plate type and divided by the total number of tickets. The majority of tickets were issued to the passenger plate type, followed by commercial vehicles.

PAS	70%
COM	21%

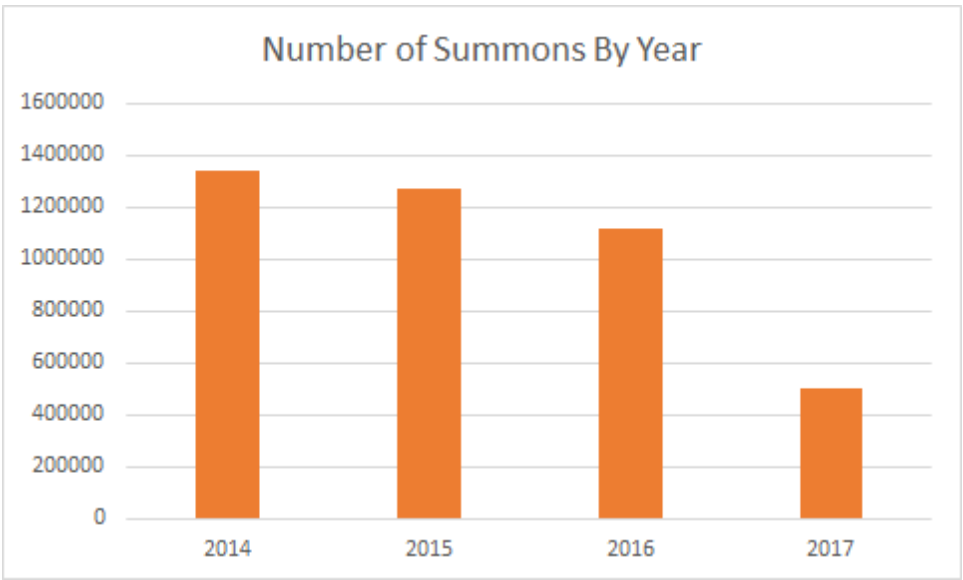
Citation Counts by Month

Below is a timeline of counts between August 2013 and June 2017. The sharp spike in January 2015 (18) is most striking, especially considering the low counts in December 2014 and February 2015. There doesn't appear to be any true cyclical trend in the timeline, except that September and October appear to have consistently higher counts relative to local trends (2-3, 14-15, 26-27, 38-39). January 2014 appears to have approximately 725,000 citations compared to January 2015 with almost 1.4M citations—nearly double. The count for January 2016 (30) is approximately 820,000. There is a visible downward trend from January 2015. Further exploration of this time period centered around January 2015 is needed. Examining patterns and clusters of citations in the months of September and October would be prudent.



Summons per Year

The total number of summons was counted by year. Excluding the sharp decline in 2017, which only contains 6 months of data, the number of summons slightly decreased from 2014 to 2016

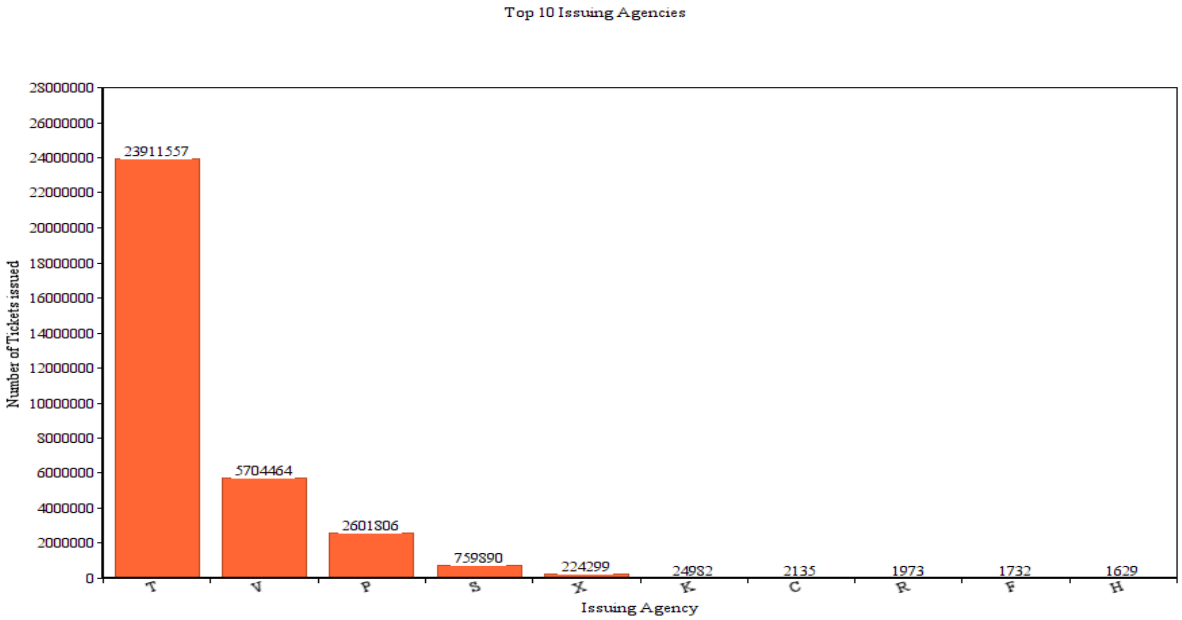


Street

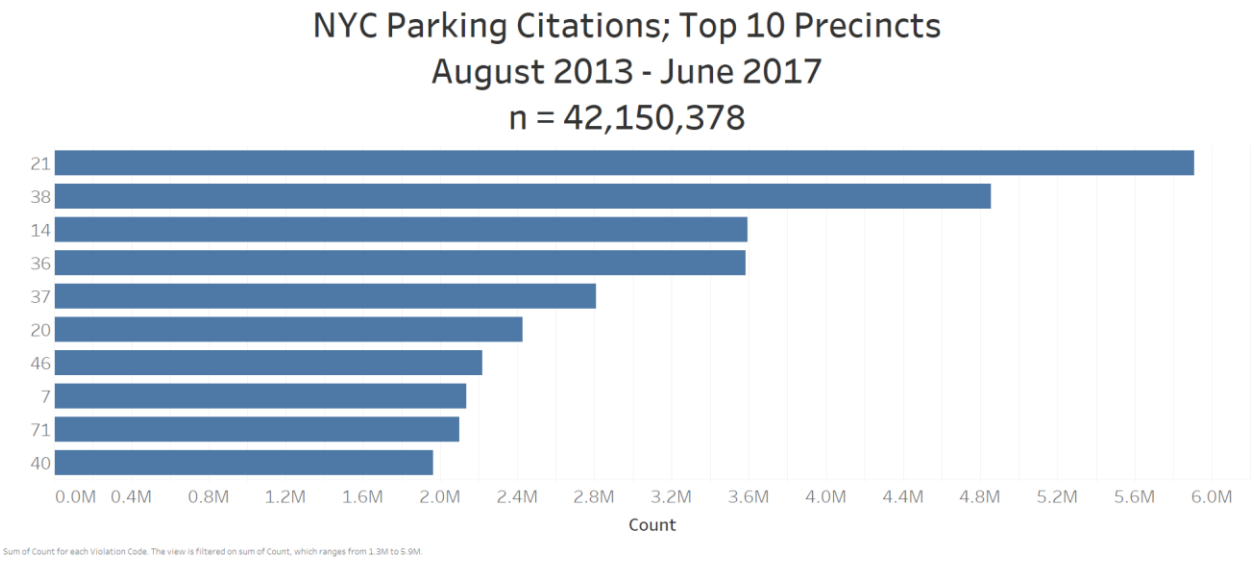
Manhattan has the most number of issuances. And Broadway has the maximum number of tickets. Since Broadway is the longest street in the city, we will study the violation location to locate where on Broadway the tickets were issued.

Issuing Agency

From the result, Agent T issued majority of tickets. It is about 80% of total issued tickets.



Type of violations



There are about 100 type codes, so the chart shows the top 10. They account for about 70% of all tickets.

Rank	Violation Code	DESC
1	21	Street Cleaning
2	38	Parking Meter: Parking in excess of the allowed time
3	14	General No Standing
4	36	Exceeding the posted speed limit in or near a designated school zone.
5	37	Parking Meter: Failing to show a receipt or tag in the windshield. Drivers get a 5-minute grace period past the expired time on parking meter receipts.
6	20	General No Parking
7	46	Standing or parking on the roadway side of a vehicle stopped, standing or parked at the curb
8	7	Vehicles photographed going through a red light at an intersection
9	71	Standing or parking a vehicle without showing a current New York inspection sticker.
10	40	Stopping, standing or parking closer than 15 feet of a fire hydrant.

Analysis Plan

The goal of the analysis will be to look for groupings among the parking violations. Features from the parking violations dataset will be clustered in order to look for these groupings of violations. The groups will then be characterized. Some initial features of interest are location, car make, car model, car color, date, time, registration state, and issuing precinct. Location of violations over time will be visualized by creating choropleth maps. Similarly, choropleths can potentially be used to visually explore the clusters.

Appendix

Original variables from CSV files

1. Summons Number 2. Plate ID 3. Registration State 4. Plate Type 5. Issue Date 6. Violation Code 7. Vehicle Body Type 8. Vehicle Make 9. Issuing Agency 10. Street Code1 11. Street Code2 12. Street Code3 13. Vehicle Expiration Date 14. Violation Location 15. Violation Precinct 16. Issuer Precinct 17. Issuer Code 18. Issuer Command 19. Issuer Squad 20. Violation Time 21. Time First Observed 22. Violation County 23. Violation In Front Of Or Opposite 24. House Number 25. Street Name 26. Intersecting Street	27. Date First Observed 28. Law Section 29. Sub Division 30. Violation Legal Code 31. Days Parking In Effect 32. From Hours In Effect 33. To Hours In Effect 34. Vehicle Color 35. Unregistered Vehicle? 36. Vehicle Year 37. Meter Number 38. Feet From Curb 39. Violation Post Code 40. Violation Description 41. No Standing or Stopping Violation 42. Hydrant Violation 43. Double Parking Violation 44. Latitude 45. Longitude 46. Community Board 47. Community Council 48. Census Tract 49. BIN 50. BBL 51. NTA

SQL Code

```
--- Load CSV files into Hadoop File System
hdfs dfs -put /pylon5/cc5phlp/ever930/data/project/NYC_Parking_Citations/Parking_Violations_Issued_-
_Fiscal_Year_2014__August_2013___June_2014_.csv
hdfs dfs -put /pylon5/cc5phlp/ever930/data/project/NYC_Parking_Citations/Parking_Violations_Issued_-_Fiscal_Year_2015.csv
hdfs dfs -put /pylon5/cc5phlp/ever930/data/project/NYC_Parking_Citations/Parking_Violations_Issued_-_Fiscal_Year_2016.csv
hdfs dfs -put /pylon5/cc5phlp/ever930/data/project/NYC_Parking_Citations/Parking_Violations_Issued_-_Fiscal_Year_2017.csv

--- Create temporary table
create table if not exists nyc_parking_violations_temp
(summons_number int,
plate_ID varchar(10),
registration_state char(2),
plate_type varchar(3),
issue_date string,
violation_code int,
vehicle_body_type varchar(10),
vehicle_make varchar(10),
issuing_agency char(1),
street_code1 int,
street_code2 int,
street_code3 int,
vehicle_expiration_date int,
violation_location int,
violation_precinct int,
issuer_precinct int,
issuer_code int,
issuer_command varchar(10),
issuer_squad varchar(10),
violation_time varchar(10),
time_first_observed varchar(10),
violation_county char(5),
violation_in_front_of_or_opposite char(1),
house_number varchar(10),
street_name varchar(50),
intersecting_street varchar(50),
date_first_observed int,
law_section int,
sub_division varchar(2),
violation_legal_code varchar(1),
days_parking_in_effect varchar(10),
from_hours_in_effect varchar(10),
to_hours_in_effect varchar(10),
vehicle_color char(5),
unregistered_vehicle int,
vehicle_year int,
meter_number varchar(10),
feet_from_curb int,
violation_post_code varchar(5),
violation_description varchar(50),
no_standing_or_stopping_violation boolean,
hydrant_violation boolean,
```

```
double_parking_violation boolean,
latitude boolean,
longitude boolean,
community_board boolean,
community_council boolean,
census_tract boolean,
BIN boolean,
BBL boolean,
NTA boolean)
partitioned by(
year int,
month int,
day int,
hour int,
violation_code int,
issuer_precinct int
)
row format delimited
fields terminated by ','
lines terminated by '\n'
stored as textfile
tblproperties ("skip.header.line.count"="1");

--- Load data into temporary table
load data inpath "Parking_Violations_Issued_-_Fiscal_Year_2014__August_2013___June_2014_.csv" into table
nyc_parking_violations_temp;
load data inpath "Parking_Violations_Issued_-_Fiscal_Year_2015.csv" into table nyc_parking_violations_temp;
load data inpath "Parking_Violations_Issued_-_Fiscal_Year_2016.csv" into table nyc_parking_violations_temp;
load data inpath "Parking_Violations_Issued_-_Fiscal_Year_2017.csv" into table nyc_parking_violations_temp;

--- Create smaller partitioned pivot table for analysis
create table if not exists nyc_parking_violations
(summons_number int,
plate_ID varchar(10),
violation_code int,
violation_location int,
violation_precinct int,
issuer_precinct int,
issuer_command varchar(10),
issuer_squad varchar(10),
street_name varchar(50),
vehicle_color char(5),
vehicle_make varchar(10),
vehicle_body_type varchar(10),
vehicle_year int,
violation_description varchar(50),
day int,
hour int
)
partitioned by(
year int,
month int
)
row format delimited
```

```
fields terminated by ','
lines terminated by '\n'
stored as textfile;

--- Code for dynamic partitioning
set hive.exec.dynamic.partition =  TRUE;
set hive.exec.dynamic.partition.mode = nonstrict;
set hive.exec.max.dynamic.partitions = 3000;
set hive.exec.max.dynamic.partitions.pernode = 3000;

--- Load data into smaller partitioned pivot table
insert overwrite table nyc_parking_violations
partition(
year,
month
)
select
summons_number,
plate_ID,
violation_code,
violation_location,
violation_precinct,
issuer_precinct,
issuer_command,
issuer_squad,
street_name,
vehicle_color,
vehicle_make,
vehicle_body_type,
vehicle_year,
violation_description,
day(to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy')))),
(case
when (violation_time regexp '[0-1][0-9][0-9][0-9][A-Z]') and (substring(violation_time,5,5) == 'A') and
(substring(violation_time,1,2) == '12') then cast('0' as int)
when (violation_time regexp '[0-1][0-9][0-9][0-9][A-Z]') and (substring(violation_time,5,5) == 'P') and
(substring(violation_time,1,2) == '12') then cast(substring(violation_time,1,2) as int)
when (violation_time regexp '[0-1][0-9][0-9][0-9][A-Z]') and (substring(violation_time,5,5) == 'P') and
(substring(violation_time,1,2) != '12') then cast(substring(violation_time,1,2) as int) + 12
when (violation_time regexp '[0-1][0-9][0-9][0-9][A-Z]') and (substring(violation_time,5,5) == 'A') and
(substring(violation_time,1,2) != '12') then cast(substring(violation_time,1,2) as int)
end),
year(to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy')))),
month(to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy'))))
from nyc_parking_violations_temp
where to_date(from_unixtime(unix_timestamp(issue_date, 'MM/dd/yyyy'))) between '2013-08-01' and '2017-06-30';
```