



CIS4560 Term Project Tutorial



Authors: Rene Barillas, Ruben Becerra, Aaron Lopez, Brian Burwick, Erick Arevalo, Nazmul Khan

Instructor: [Jongwook Woo](#)

Date: 05/21/2023

Lab Tutorial

Rene Barillas, Ruben Becerra, Aaron Lopez, Brian Burwick, Erick Arevalo, Nazmul Khan
(rbarill2@calstatela.edu, rbecer18@calstatela.edu, alopez453@calstatela.edu, bburwic@calstatela.edu,
eareva25@calstatela.edu, nislamk@calstatela.edu)

05/21/2023

Analysis of Parking Tickets in New York City using Hadoop

Objectives

In this hands-on tutorial, you will learn how to:

- Download a publicly available dataset and upload it to Hadoop cluster
- Create hive tables to query data
- Create queries to analyze trends and gain better insight into the data
- Merge multiple files together and download files to your local computer
- Create Tempo-Spatial Visualizations using Excel 3D Maps

Platform Spec

- Oracle Linux Server
- CPU Speed: 1995 MHz
- # of CPU cores: 8
- # of nodes: 5
- Total Memory Size: 390.71 GB

Step 1: Download Data from Kaggle

1. Visit: https://www.kaggle.com/datasets/new-york-city/nyc-parking-tickets?select=Parking_Violations_Issued_-_Fiscal_Year_2017.csv
2. Make sure Fiscal Year 2017 is selected under the Data Explorer.
3. Download the data.

Search

CITY OF NEW YORK · UPDATED 3 YEARS AGO

252

New Notebook

Download (2 GB)

NYC Parking Tickets

42.3M Rows of Parking Ticket Data, Aug 2013-June 2017

Data Card

Code (15)

Discussion (0)

About Dataset

Context

The NYC Department of Finance collects data on every parking ticket issued in NYC (~10M per year!). This data is made publicly available to aid in ticket resolution and to guide policymakers.

Content

There are four files, covering Aug 2013-June 2017. The files are roughly organized by fiscal year (July 1 - June 30) with the exception of the initial dataset. The initial dataset also lacks 8 columns that are included in the other three datasets (although be warned that these additional data columns are used sparingly). See the dataset descriptions for exact details. Columns include information about the vehicle ticketed, the ticket issued, location, and time.

Acknowledgements

Data was produced by NYC Department of Finance. FY2018 data is found [here](#) with updates every third week of the month.

Inspiration

- When are tickets most likely to be issued? Any seasonality?
- Where are tickets most commonly issued?
- What are the most common years and types of cars to be ticketed?

View more

Usability

8.24

License

CC0: Public Domain

Expected update frequency

Not specified

Parking_Violations_Issued_-_Fiscal_Year_2017.csv

(2.09 GB)

Detail

Compact

Column

About this file

- Summons Number: Number
- Plate ID: Plain Text

10 of 43 columns

Data Explorer

Version 2 (8.97 GB)

Parking_Violations_Issued_-_F

Parking_Violations_Issued_-_F

Parking_Violations_Issued_-_F

Parking_Violations_Issued_-_F

Parking_Violations_Issued_-_Fiscal_Year_2017.csv (2.09 GB)

Automobiles and Vehicles

Law

Transportation

Travel

Government

Cities and Urban Areas

Step 2: Upload Data and Connect to Hadoop Cluster

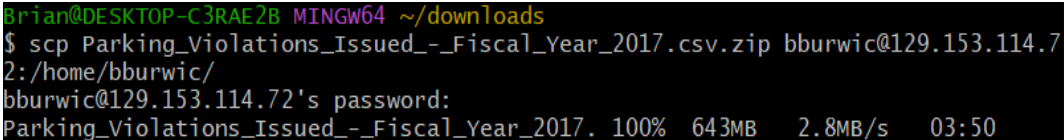
1. Open Git Bash and type in change your directory to downloads. (If you moved the downloaded file from the previous step change into that directory instead)

```
$ cd downloads
```

2. Upload the data to hdfs using the following **scp** command:

```
$ scp Parking_Violations_Issued_-_Fiscal_Year_2017.csv.zip
```

```
jwoo5@129.153.114.72:/home/jwoo5/
```



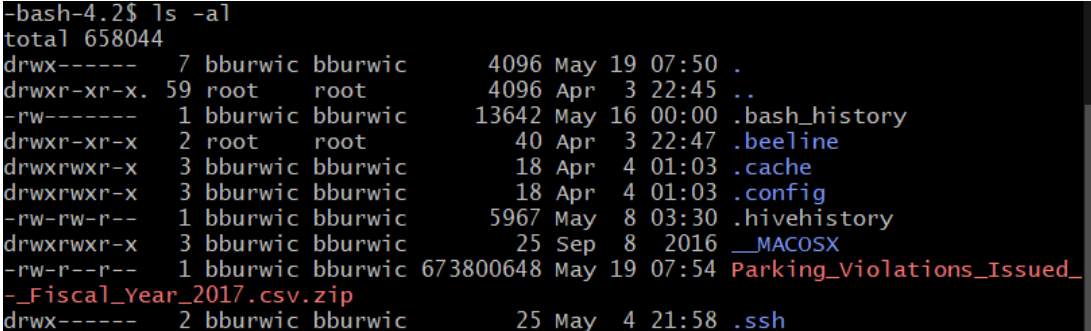
```
Brian@DESKTOP-C3RAE2B MINGW64 ~/downloads
$ scp Parking_Violations_Issued_-_Fiscal_Year_2017.csv.zip bburwic@129.153.114.72:/home/bburwic/
bburwic@129.153.114.72's password:
Parking_Violations_Issued_-_Fiscal_Year_2017. 100% 643MB 2.8MB/s 03:50
```

3. Use the ssh command to remotely connect to your Hadoop Cluster using the Git Bash terminal.

```
$ ssh jwoo5@129.153.114.72
```

4. Verify the file was uploaded using the ls command:

```
$ ls -al
```



```
-bash-4.2$ ls -al
total 658044
drwx----- 7 bburwic bburwic 4096 May 19 07:50 .
drwxr-xr-x. 59 root root 4096 Apr 3 22:45 ..
-rw----- 1 bburwic bburwic 13642 May 16 00:00 .bash_history
drwxr-xr-x 2 root root 40 Apr 3 22:47 .beeline
drwxrwxr-x 3 bburwic bburwic 18 Apr 4 01:03 .cache
drwxrwxr-x 3 bburwic bburwic 18 Apr 4 01:03 .config
-rw-rw-r-- 1 bburwic bburwic 5967 May 8 03:30 .hivehistory
drwxrwxr-x 3 bburwic bburwic 25 Sep 8 2016 __MACOSX
-rw-r--r-- 1 bburwic bburwic 673800648 May 19 07:54 Parking_Violations_Issued_-_Fiscal_Year_2017.csv.zip
drwx----- 2 bburwic bburwic 25 May 4 21:58 .ssh
```

5. Unzip the folder using the following code:

```
$ unzip Parking_Violations_Issued_-_Fiscal_Year_2017.csv.zip
```

6. Once the file is finished unzipping run the ls command again and verify the

Parking_Violations_Issued_-_Fiscal_Year_2017.csv shows at 2.09 GB.

```
$ ls -al
```

```
-bash-4.2$ unzip Parking_Violations_Issued_-_Fiscal_Year_2017.csv.zip
Archive:  Parking_Violations_Issued_-_Fiscal_Year_2017.csv.zip
  inflating: Parking_Violations_Issued_-_Fiscal_Year_2017.csv
-bash-4.2$ ls -al
total 2696048
drwx----- 7 bburwic bburwic      4096 May 19 07:59 .
drwxr-xr-x. 59 root    root        4096 Apr  3 22:45 ..
-rw----- 1 bburwic bburwic    13642 May 16 00:00 .bash_history
drwxr-xr-x  2 root    root         40 Apr  3 22:47 .beeline
drwxrwxr-x  3 bburwic bburwic      18 Apr  4 01:03 .cache
drwxrwxr-x  3 bburwic bburwic      18 Apr  4 01:03 .config
-rw-rw-r-- 1 bburwic bburwic     5967 May  8 03:30 .hivehistory
drwxrwxr-x  3 bburwic bburwic      25 Sep  8 2016 __MACOSX
-rw-rw-r-- 1 bburwic bburwic 2086913576 May 10 2020 Parking_Violations_Issued
_-_Fiscal_Year_2017.csv
-rw-r--r-- 1 bburwic bburwic 673800648 May 19 07:54 Parking_Violations_Issued
_-_Fiscal_Year_2017.csv.zip
drwx----- 2 bburwic bburwic      25 May  4 21:58 .ssh
```

7. Create the following directories in hdfs. We will be using them for future steps in the tutorial to store our data.

```
$ hdfs dfs -mkdir NYCTicketData
```

```
$ hdfs dfs -mkdir NYCTicketData/Violations
```

```
$ hdfs dfs -mkdir NYCTicketData/Vehicles
```

```
$ hdfs dfs -mkdir NYCTicketData/Locations
```

8. Use the put command to place the Parking_Violations_Issued_-_Fiscal_Year_2017.csv file into the NYCTicketData directory.

```
$ hdfs dfs -put Parking_Violations_Issued_-_Fiscal_Year_2017.csv NYCTicketData
```

9. Run the ls command one final time to verify the files were uploaded into the hdfs directory then you may proceed to the next step.

```
$ hdfs dfs -ls NYCTicketData
```

```
Found 4 items
drwxr-xr-x - bburwic hdfs      0 2023-05-19 08:11 NYCTicketData/Locations
-rw-r--r-- 3 bburwic hdfs 2086913576 2023-05-19 08:11 NYCTicketData/Parking_Vi
olations_Issued_-_Fiscal_Year_2017.csv
drwxr-xr-x - bburwic hdfs      0 2023-05-19 08:11 NYCTicketData/Vehicles
drwxr-xr-x - bburwic hdfs      0 2023-05-19 08:11 NYCTicketData/Violations
-bash-4.2$
```

Step 3: Creating Tables in Beeline

1. Open a new Git Bash terminal and ssh to your Hadoop cluster. Once connected type in beeline.

```
$ ssh jwoo5@129.153.114.72
```

```
$ beeline
```

2. Create a database for the NYC Ticket data and use the database.

```
hive: CREATE DATABASE if not exists NYCTickets;
```

```
hive: use NYCTickets;
```

3. Create a master data table to hold the original csv file. This table will be used to create future tables.

```
CREATE EXTERNAL TABLE IF NOT EXISTS 2017data(
```

```
summons_number BIGINT,
```

```
plate_id STRING,
```

```
registration_state STRING,
```

```
plate_type STRING,
```

```
issue_date STRING,
```

```
violation_code INT,
```

```
vehicle_body_type STRING,
```

```
vehicle_make STRING,
```

```
issuing_agency STRING,
```

```
street_code_1 INT,
```

```
street_code_2 INT,
```

```
street_code_3 INT,
```

```
vehicle_expiration_date INT,
```

violation_location STRING,
violation_precinct INT,
issuer_precinct INT,
issuer_code INT,
issuer_command STRING,
issuer_squad STRING,
violation_time STRING,
time_first_observed STRING,
violation_county STRING,
violation_in_front_of_or_opposite STRING,
house_number STRING,
street_name STRING,
intersecting_street STRING,
date_first_observed INT,
law_section INT,
sub_division STRING,
violation_legal_code STRING,
days_parking_in_effect STRING,
from_hours_in_effect STRING,
to_hours_in_effect STRING,
vehicle_color STRING,
unregistered_vehicle STRING,
vehicle_year INT,
meter_number STRING,

```

    feet_from_curb INT,

    violation_post_code STRING,

    violation_description STRING,

    no_standing_or_stopping_violation STRING,

    hydrant_violation STRING,

    double_parking_violation STRING

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/bburwic/NYCTicketData'

TBLPROPERTIES ('skip.header.line.count'='1');

```

4. Create the ViolationData, VehicleData, and LocationData tables and insert values from the master table into them using the following code. Note: The unix_timestamp function is formatting the date to be used properly as the master table has date formatted as string.

```

CREATE TABLE ViolationData

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION 'NYCTicketData/Violations'

AS

SELECT summons_number, TO_DATE(from_unixtime(unix_timestamp(issue_date,
'mm/dd/yyyy'), 'yyyy-mm-dd')) AS issue_date, violation_code, violation_precinct,
violation_county, CONCAT_WS(' ', house_number, street_name) AS address

FROM 2017data;

```

```
CREATE TABLE VehicleData

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION 'NYCTicketData/Vehicles'

AS

SELECT summons_number, plate_id, registration_state, plate_type, vehicle_body_type,

vehicle_make, vehicle_expiration_date, vehicle_year, vehicle_color,

unregistered_vehicle

FROM 2017data;
```

```
CREATE TABLE LocationData

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION 'NYCTicketData/Locations'

AS

SELECT summons_number, street_code_1, violation_precinct, violation_county,

house_number, street_name

FROM 2017data;
```

Step 4: Queries for Data Insight

To get an understanding of the amount of data we are working with and some specific results from the data we will run a handful of queries to get better insights into the data.

1. Run the following code to see the number of records in the 2017data ticket table.

```
SELECT COUNT(*) AS total_tickets FROM 2017data;
```


total_tickets
151242363

- The following code will show the top 10 violation codes issued among the entire dataset.

```
SELECT violation_code, COUNT(*) as num_violations
FROM ViolationData
WHERE violation_code IS NOT NULL
GROUP BY violation_code
ORDER BY num_violations DESC
LIMIT 10;
```

violation_code	num_violations
21	1528578
36	1400618
38	1062302
14	893496
20	618593
46	600009
37	596766
71	521309
40	519617
7	516405

- The following code will show the top 10 vehicle makes that received tickets.

```
SELECT vehicle_make, COUNT(*) as num_tickets
FROM VehicleData
WHERE vehicle_make REGEXP '^[^0-9]+$' AND vehicle_make IS NOT NULL
GROUP BY vehicle_make
ORDER BY num_tickets DESC
```

LIMIT 10;

vehicle_make	num_tickets
FORD	1280956
TOYOT	1211447
HONDA	1079237
NISSA	918590
CHEVR	714654
FRUEH	429155
ME/BE	389050
BMW	374926
DODGE	372125
JEEP	348144

4. The following code shows the top 10 streets with the most number of tickets issued.

```
SELECT street_name, COUNT(*) as num_tickets
```

```
FROM LocationData
```

```
WHERE street_name IS NOT NULL
```

```
GROUP BY street_name
```

```
ORDER BY num_tickets DESC
```

LIMIT 10;

street_name	num_tickets
Broadway	206157
3rd Ave	160279
Madison Ave	98853
5th Ave	90706
Lexington Ave	81198
2nd Ave	77738
1st Ave	72719
7th Ave	66541
8th Ave	57458
6th Ave	55903

5. The following code lists the top 10 counties that issued tickets.

```

SELECT violation_county, COUNT(*) AS num_tickets

FROM ViolationData

WHERE violation_county IS NOT NULL

GROUP BY violation_county

ORDER BY num_tickets DESC

LIMIT 10;

```

violation_county	num_tickets
NY	3433001
K	2218821
Q	1838979
BX	1362460
BK	880657
QN	674245
ST	137542
R	121416
MN	96298
	39597

6. The following code shows the top 25 most ticketed days in 2017.

```

SELECT issue_date, COUNT(*) as num_tickets

FROM ViolationData

WHERE issue_date IS NOT NULL

GROUP BY issue_date

ORDER BY num_tickets DESC

LIMIT 25;

```

issue_date	num_tickets
2016-09-16	46860
2016-09-27	46270
2016-10-07	45892
2016-10-06	45870
2016-10-11	45820
2017-03-21	45792
2017-03-02	45792
2017-05-11	45592
2017-03-23	45464
2016-09-15	45443
2016-10-13	45435
2016-09-29	45388
2017-05-02	45139
2016-11-10	45060
2017-06-02	45005
2016-10-20	44911
2017-05-23	44545
2017-03-09	44436
2016-11-17	44274
2017-06-15	44265
2016-09-26	44258
2017-06-09	44228
2017-03-30	44119
2016-09-08	44013
2017-05-01	43971

Step 5: Creating Tables to Analyze Bronx, Kings, & Queens

To get a better understanding of the counties of Bronx, Kings, and Queens we will create tables analyzing the first quarter of the year. Then we will download the data and prepare it for visualization for the next step.

1. The following code creates a table of violations for the year's first quarter for Bronx County. We concatenate the address with Bronx, New York, to make visualizations in Excel 3D Maps more accurate.

```
CREATE TABLE BronxViolations

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION 'NYCTicketData/BronxViolations'

AS

SELECT summons_number, issue_date, violation_code, violation_precinct,

    CONCAT(address, ' Bronx New York') AS address_new

FROM ViolationData

WHERE violation_county = 'BX' AND issue_date BETWEEN '2017-01-01' AND '2017-03-31';
```

```
CREATE TABLE KingsViolations

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION 'NYCTicketData/KingsViolations'

AS

SELECT summons_number, issue_date, violation_code, violation_precinct,

    CONCAT(address, ' Kings New York') AS address_new

FROM ViolationData

WHERE violation_county = 'K' AND issue_date BETWEEN '2017-01-01' AND '2017-03-31';
```

```
CREATE TABLE QueensViolations

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION 'NYCTicketData/QueensViolations'

AS

SELECT summons_number, issue_date, violation_code, violation_precinct,
```

```

CONCAT(address, ' Queens New York') AS address_new

FROM ViolationData

WHERE violation_county = 'Q' AND issue_date BETWEEN '2017-01-01' AND '2017-03-
31';

```

2. Once the new tables are created use the count function in the following code to see the number of tickets with the following code. If we have too many tickets, Excel may not function properly.

```

SELECT COUNT(*) AS total_tickets FROM BronxViolations;

SELECT COUNT(*) AS total_tickets FROM KingsViolations;

SELECT COUNT(*) AS total_tickets FROM QueensViolations;

```

Bronx:

Kings:

Queens:

total_tickets	total_tickets	total_tickets
336373	569769	464464

Step 6: Visualization

You can download data from the Bronx, Kings, and Queens tables to your local computer by following the below instructions.

1. Have at least two (2) Git Bash terminals open. One terminal should be connected to the Hadoop Cluster, not in beeline, and the second terminal will be local to manage the download to your computer.

2. Run the following code in hdfs to merge the multiple files stored within the Bronx, Kings, and Queens Violations directories into one CSV file each. Then verify the files exist with the ls command.

```
$ hdfs dfs -getmerge /user/bburwic/NYCTicketData/BronxViolations bronxviolations.csv
```

```
$ hdfs dfs -getmerge /user/bburwic/NYCTicketData/KingsViolations kingsviolations.csv
```

```
$ hdfs dfs -getmerge /user/bburwic/NYCTicketData/QueensViolations
```

```
queensviolations.csv
```

```
$ ls -al
```

```
-bash-4.2$ hdfs dfs -getmerge /user/bburwic/NYCTicketData/BronxViolations bronxviolations.csv
-bash-4.2$
-bash-4.2$ hdfs dfs -getmerge /user/bburwic/NYCTicketData/KingsViolations kingsviolations.csv
-bash-4.2$
-bash-4.2$ hdfs dfs -getmerge /user/bburwic/NYCTicketData/QueensViolations queensviolations.csv
-bash-4.2$
-bash-4.2$ ls -al
total 2776176
drwx----- 7 bburwic bburwic      4096 May 19 10:53 .
drwxr-xr-x. 59 root     root       4096 Apr  3 22:45 ..
-rw----- 1 bburwic bburwic    14158 May 19 10:47 .bash_history
drwxr-xr-x  2 root     root        40 Apr  3 22:47 .beeline
-rw-r--r-- 1 bburwic bburwic  20311790 May 19 10:53 bronxviolations.csv
-rw-r--r-- 1 bburwic bburwic   158696 May 19 10:53 .bronxviolations.csv.crc
drwxrwxr-x  3 bburwic bburwic     18 Apr  4 01:03 .cache
drwxrwxr-x  3 bburwic bburwic     18 Apr  4 01:03 .config
-rw-rw-r-- 1 bburwic bburwic    5967 May  8 03:30 .hivehistory
-rw-r--r-- 1 bburwic bburwic  32931059 May 19 10:53 kingsviolations.csv
-rw-r--r-- 1 bburwic bburwic   257284 May 19 10:53 .kingsviolations.csv.crc
drwxrwxr-x  3 bburwic bburwic     25 Sep  8 2016 __MACOSX
-rw-rw-r-- 1 bburwic bburwic 2086913576 May 10 2020 Parking_Violations_Issued
--Fiscal_Year_2017.csv
-rw-r--r-- 1 bburwic bburwic   673800648 May 19 07:54 Parking_Violations_Issued
--Fiscal_Year_2017.csv.zip
-rw-r--r-- 1 bburwic bburwic   28165283 May 19 10:53 queensviolations.csv
-rw-r--r-- 1 bburwic bburwic   220052 May 19 10:53 .queensviolations.csv.crc
drwx----- 2 bburwic bburwic     25 May  4 21:58 .ssh
-bash-4.2$
```

3. Run the following scp commands in your local Git Bash terminal to download the CSV files to your local computer. The files will be located under C:/Users/YourName.

```
scp bburwic@129.153.114.72:/home/bburwic/bronxviolations.csv bronxviolations.csv
```

```
scp bburwic@129.153.114.72:/home/bburwic/kingsviolations.csv kingsviolations.csv
```

scp bburwic@129.153.114.72:/home/bburwic/queensviolations.csv

queensviolations.csv

```
Brian@DESKTOP-C3RAE2B MINGW64 ~
$ scp bburwic@129.153.114.72:/home/bburwic/bronxviolations.csv bronxviolations.csv
bburwic@129.153.114.72's password:
bronxviolations.csv                                100%  19MB  21.5MB/s   00:00

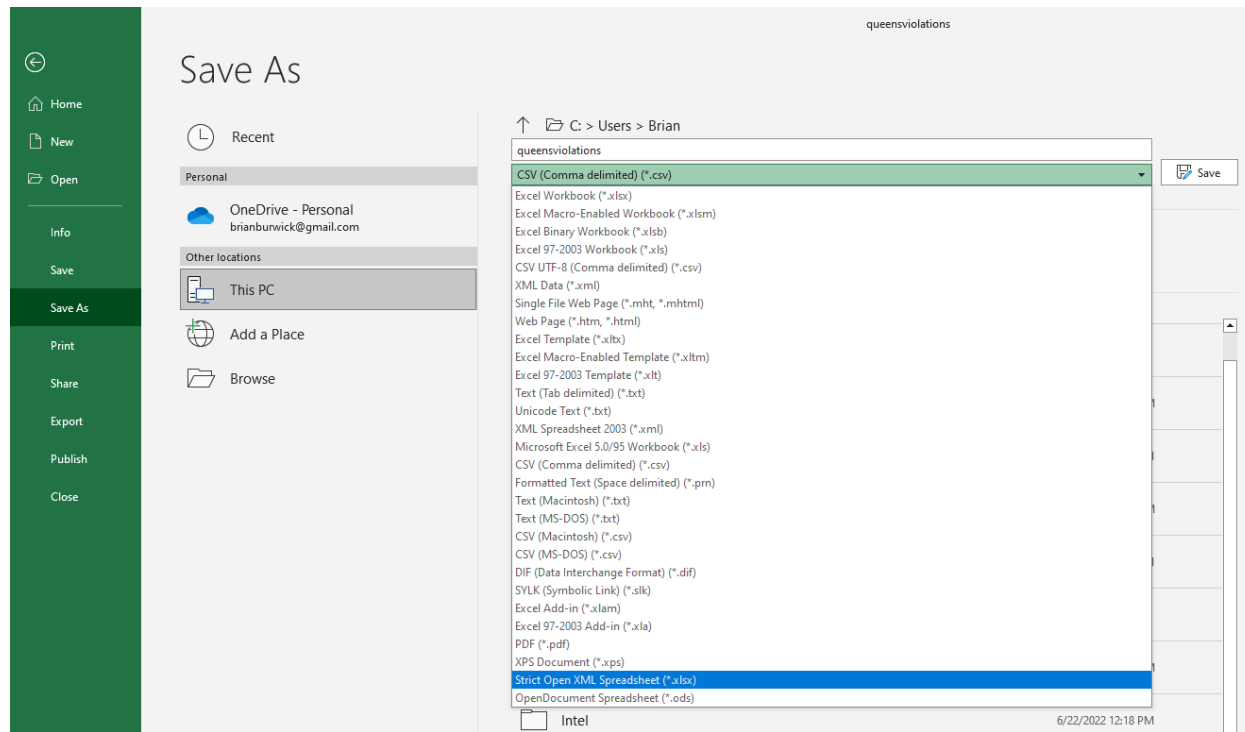
Brian@DESKTOP-C3RAE2B MINGW64 ~
$ scp bburwic@129.153.114.72:/home/bburwic/kingsviolations.csv kingsviolations.csv
bburwic@129.153.114.72's password:
kingsviolations.csv                                100%  31MB  25.5MB/s   00:01

Brian@DESKTOP-C3RAE2B MINGW64 ~
$ scp bburwic@129.153.114.72:/home/bburwic/queensviolations.csv queensviolations.csv
bburwic@129.153.114.72's password:
queensviolations.csv                                100%  27MB  25.3MB/s   00:01
```

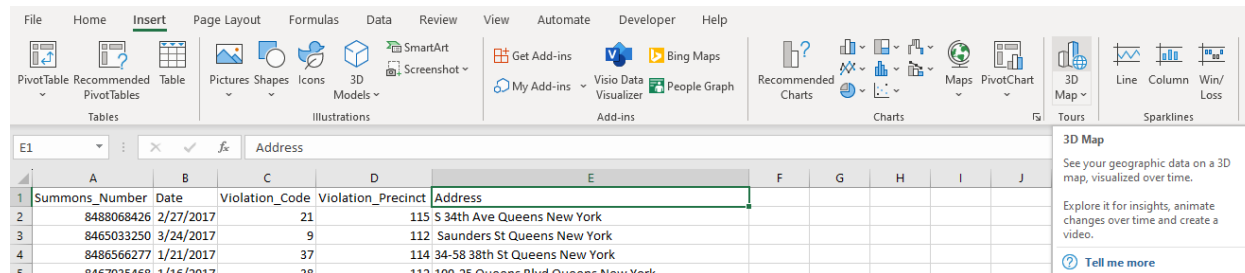
4. Open each CSV file and insert a row at the very top of the spreadsheet and place the following in the fields as headers: Summons_Number, Date, Violation_Code, Violation_Precinct, Address.

	A	B	C	D	E
1	Summons_Number	Date	Violation_Code	Violation_Precinct	Address
2	8382538265	2/14/2017	82	49	902 Brady Ave Bronx New York
3	1417521417	1/3/2017	21	46	1882 JEROME AVENUE Bronx New York
4	1420521895	3/2/2017	21	44	901 WALTON AVE Bronx New York
5	1414028696	1/9/2017	71	47	3714 HARPER AVE Bronx New York
6	4631056253	2/27/2017	36	0	NB SOUTHERN BLVD @ E Bronx New York
7	8141049379	2/8/2017	37	46	2031 Grand Concourse Bronx New York
8	1420243172	3/20/2017	21	48	2419 BATHGATE AVE Bronx New York
9	8474669730	2/2/2017	21	49	2526 Bronx Park East Bronx New York
10	8523351899	3/25/2017	20	52	3535 Wayne Ave Bronx New York
11	8034556804	2/3/2017	38	48	627 E Tremont Ave Bronx New York
12	5095088940	1/21/2017	7	0	GRAND CONCOURSE (N/B Bronx New York
13	8379523763	2/14/2017	14	50	3444 Bailey Pl Bronx New York
14	7057576043	2/16/2017	19	43	2200 E Tremont Ave Bronx New York
15	4630428604	2/7/2017	36	0	SB BAILEY AVE @ W 22 Bronx New York
16	8528955655	3/29/2017	38	45	3752 E Tremont Ave Bronx New York
17	8524054610	3/20/2017	21	49	N Radcliff Ave Bronx New York
18	1414039220	3/3/2017	46	43	1268 CASTLE HILL Bronx New York
19	8289056199	2/7/2017	38	49	S Roberts Ave Bronx New York
20	7601562876	1/19/2017	21	49	2526 Bronx Park East Bronx New York

5. Once you have added the header rows for each of the CSV files save each one as a Strict Open XML Spreadsheet .xlsx file. This enables the data to be used in Excel 3D Maps.



- After saving as an Open XML Spreadsheet go to the insert tab at the top of excel and click on 3D Map.



- On the right side of the 3D Map Tour select the following values for visualization. Click the drop-down arrow on the right of respective fields for additional options.

Layer 1

Data

Location

Address
Full Address

+ Add Field

Height

Violation_Code (Count - Distinct)

+ Add Field

Category

Violation_Code

Time

Date (Day)

Filters

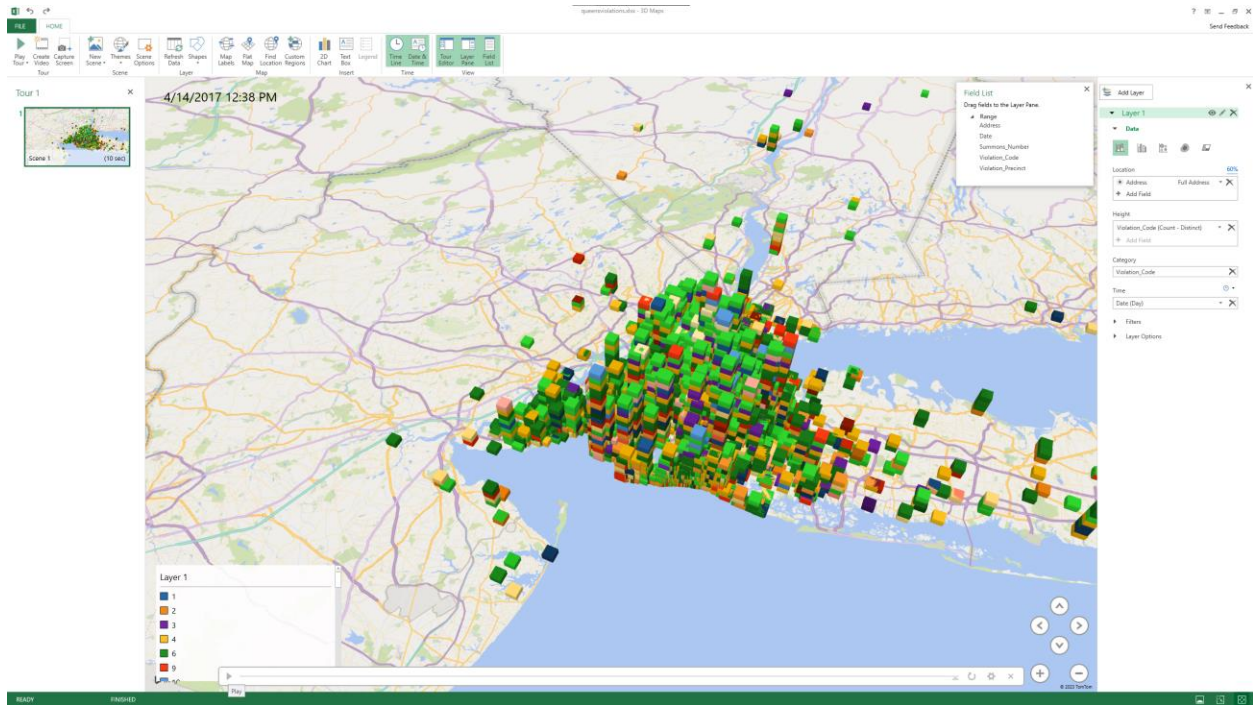
Layer Options

- Wait patiently for Excel to process the addresses for visualization. This can take 10-20 minutes. You can check the progress on the bottom left corner of the Excel screen.

READY

FINISHED

- Once all locations are processed, press the play button on the bottom of the screen and move the map around to see the spread of tickets over the year's first quarter for the respective county.



10. Repeat steps 6-9 to visualize each county if you wish.

References

- [1] Ackerman, Samuel S. Red Zone, Blue Zone: Discovering Parking Ticket Trends in New York City, Accessed 8 May 2023.
https://newyorkparkingticket.com/wp-content/uploads/2016/11/NYC-Parking-Ticket-Report_parking_Samuel_Ackerman5.pdf
- [2] Ginzburg, Steven. “Data Visualizing New York City’s Parking Violation.” Data Science Blog, 1 May 2016, nycdatascience.com/blog/student-works/data-visualizing-new-york-citys-parking-violation/.
- [3] York, City of New. “NYC Parking Tickets.” Kaggle, CITY OF NEW YORK, 10 May 2020, <https://www.kaggle.com/datasets/new-york-city/nyc-parking-tickets>.
- [4] Burwick, Brian GitHub Project Repository
<https://github.com/BrianBurwick/NYCTicketData2017>