# CIS4560 Term Project Tutorial


**Authors: Francisco Cortés, Arturo Pena, Kevin Anaya, Luigie Olmos, Robert Saravia**

**Instructor: Jongwook Woo**

**Date: 05/05/2021**

# Lab Tutorial

fcortes6@calststela.edu, kanaya8@calstatela.edu, lolmos11@calstatela.edu, rsaravi3@calstatela.edu, apena69@calstatela.edu

05/10/2021

# Parking Violations Issued- FY 2020 using Hadoop and Tableau

---

## Objectives

**List what your objectives are.** In this hands-on lab, you will learn how to:

- Get data from website and upload to Hadoop
- Create directory for the file
- Use Hive to create tables
- SQL commands to perform the analysis.
- Visualization

## Platform Spec

- Hadoop / Python / Spark
- CPU Speed: 2000HMz
- # of CPU cores: 48
- # of nodes: 3
- Total Memory Size: 182gb

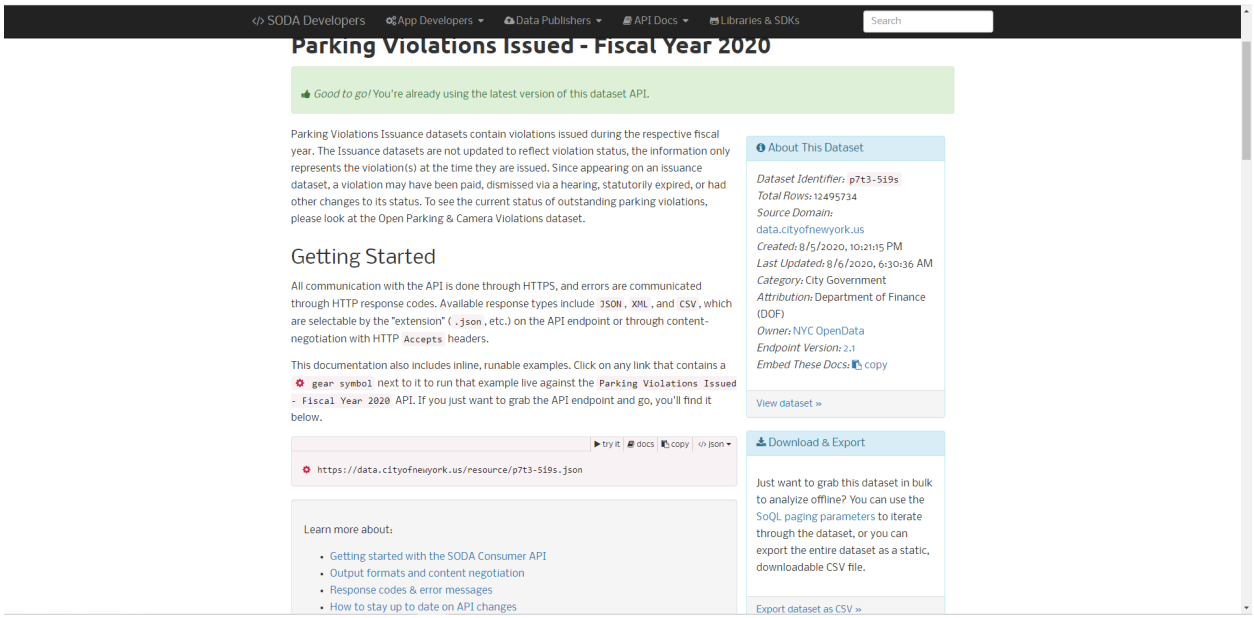
# Step 1: Get data manually by Downloading from the website

**This step is to retrieve the file.**

1. Go to https://data.cityofnewyork.us/City-Government/Parking-Violations-Issued-Fiscal-Year-2020/p7t3-5i9s

2. Once there, you will click on the three dots (...) on your upper right-hand side corner.

   Then, select API Docs, which it will take to the following image. Lastly, click on the *Export dataset as CSV* to download the file.



## Step 2: Upload the Parking Violations Issued to Hadoop

**We will begin to upload and extract the zip file Hadoop file System.**

1. The file will be located in your download folder. However, you need to move the file to the following path: *This PC > Documents*, and zip the file. To zip, *right click on the file > send to > Compressed (zipped) folder*.
2. To upload the file, open a shell terminal as Git Bash, Minty, and run following the scp command

   scp C:/Users/**fcort**/Documents/Parking_Violations_Issued_Fiscal_Year_2020.zip **fcortes6**@220.116.230.21:/home/**fcortes6**

   **Don't forget to replace fcort & fcortes6 with your computer account and hadoop account name. Close the terminal, once upload is complete.**

```
$ scp C:/Users/fcort/Documents/Parking_Violations_Issued_Fiscal_Year_2020.zip fcortes6@220.116.230.21:/home/fcortes6
fcortes6@220.116.230.21's password:
Parking_Violations_Issued_Fiscal_Year_2020.zip                100%  418MB   1.2MB/s   05:44
```

3. Open a new Shell terminal such as Git Bash, Minty, and run the ssh command to connect to the Hadoop cluster.
4. To connect to Hadoop Cluster use:

   ssh **fcortes6**@220.116.230.21 **(Don't forget to replace fcortes6 to your account name)**

5. Enter your password **(Should be the same as your username)**

```
$ ssh fcortes6@220.116.230.21
fcortes6@220.116.230.21's password:
Last login: Tue Apr 20 10:52:08 2021 from cpe-172-113-212-142.socal.res.rr.com
```

6. Once logged into your cluster, you will want to check your Hadoop cluster to ensure the file has uploaded successfully. To do so, enter the following:

   ls

```
-bash-4.2$ ls
Parking_Violations_Issued_Fiscal_Year_2020.zip  movie.java       pig_1616462952727.log  pig_1618277890114.log
__MACOSX                                        moviegenre.java  pig_1616464183682.log  ratings_2012.txt
genre.java                                      movierating.java pig_1617671860825.log  ratings_2013.txt
labPigETL                                       occupation.java  pig_1617677842479.log  user.java
```

7. Once the file is uploaded, you will need to unzip it. Since the file was uploaded as a .zip file.

   To unzip the file, enter the following:

```
unzip Parking_Violations_Issued_Fiscal_Year_2020.zip
```

```
-bash-4.2$ unzip Parking_Violations_Issued_Fiscal_Year_2020.zip
Archive:  Parking_Violations_Issued_Fiscal_Year_2020.zip
  inflating: Parking_Violations_Issued_Fiscal_Year_2020.csv
```

8. Once the file is fully uncompressed, we can now start to transfer it over to your HDFS.

   Before we begin transferring it over, we need to create a folder to save it in. To do so, enter the following:

```
hdfs dfs -mkdir ParkingViolationsIssued20
```

9. To make sure, that the directory was created, we will now list the directories by using

```
hdfs dfs -ls
```

```
-bash-4.2$ hdfs dfs -ls
Found 10 items
drwx------   - fcortes6 hdfs          0 2021-04-20 21:00 .Trash
drwxr-xrwx   - fcortes6 hdfs          0 2021-03-16 10:38 .hiveJars
drwx------   - fcortes6 hdfs          0 2021-04-20 12:46 .staging
drwxr-xr-x   - fcortes6 hdfs          0 2021-04-23 07:07 ParkingViolationsIssued20
-rw-r--r--   3 fcortes6 hdfs       2043 2021-03-23 10:27 drivers.csv
drwxr-xrwx   - fcortes6 hdfs          0 2021-03-16 11:14 dualcore
drwxr-xr-x   - fcortes6 hdfs          0 2021-04-06 12:02 output
drwxr-xrwx   - fcortes6 hdfs          0 2021-03-16 11:14 ratings
drwxr-xr-x   - fcortes6 hdfs          0 2021-04-06 10:38 tmp
-rw-r--r--   3 fcortes6 hdfs    2272077 2021-03-23 10:28 truck_event_text_partition.csv
```

10. Now that we know that our directory has been created, we can now begin to move the dataset into the ParkingViolationIssued20 by using the -put command:

```
hdfs dfs -put Parking_Violations_Issued_Fiscal_Year_2020.csv ParkingViolationsIssued20/
```

11. We can then check to ensure the file transferred over to the correct directory: **hdfs dfs –ls ParkingViolationsIssued20** the output will show the directory along with the fully unzipped .csv inside of it.

```
-bash-4.2$ hdfs dfs -put Parking_Violations_Issued_Fiscal_Year_2020.csv ParkingViolationsIssued20/
-bash-4.2$ hdfs dfs -ls /user/fcortes6/ParkingViolationsIssued20
Found 1 items
-rw-r--r--   3 fcortes6 hdfs 2321997751 2021-04-23 13:29 /user/fcortes6/ParkingViolationsIssued20/Parki
ng_Violations_Issued_Fiscal_Year_2020.csv
```

12. Now that the dataset is uploaded, we can now move onto the next part, which is creating the tables and queries in hive.

## Step 3: Create the tables in Hive

**This step will allow us to create tables from the columns**

1. Open another terminal to run **hive**. Then, use **your** database of Hive as follows:

```
0: jdbc:hive2://bigdata3.iscu.ac.kr:2181,bigd> use fcortes6;
```

2. Once connected, you will need to create a table using the following Hive Command.

**Copy and paste the following into Hadoop**

```
DROP TABLE IF EXISTS Parking_Violations_Issued;

---Create Table Parking_Violations_Issued
CREATE EXTERNAL TABLE IF NOT EXISTS Parking_Violations_Issued(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_Code1 INT, Street_Code2 INT, Street_Code3 INT, Vehicle_Expiration_Date
BIGINT, Violation_Location INT, Violation_Precinct INT, Issuer_Precinct INT, Issuer_Code
BIGINT, Issuer_Command STRING, Issuer_Squad BIGINT, 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 BIGINT, 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 INT, 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 ',' LOCATION
'/user/fcortes6/ParkingViolationsIssued20/' TBLPROPERTIES ('skip.header.line.count'='1');
```

3. Then, in the hive shell, you need to check if the table “Parking_Violations_Issued” is shown:

```
0: jdbc:hive2://bigdata3.iscu.ac.kr:2181,bigd> show tables;
```

4. If it becomes successful, you will see the following:

```
+----------------------------+--+
|          tab_name          |
+----------------------------+--+
| parking_violations_issued  |
+----------------------------+--+
```

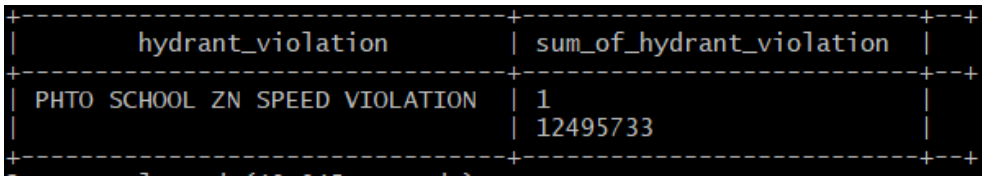
# Step 4: Creating Queries and Viewing their Output

**This step we are creating queries and outputting files for visualization**

1. Now that the tables have been created, we can then run our queries. This query will tell us how many tickets were given for parking next to a fire hydrant.
2. Enter the following into your hive:

```
select hydrant_violation, COUNT(hydrant_violation) AS Sum_Of_Hydrant_Violation from parking_violations_issued Group By hydrant_violation Order By Sum_Of_Hydrant_Violation;
```

3. It will display the result something like follows:



| hydrant_violation | sum_of_hydrant_violation |
|---|---|
| PHTO SCHOOL ZN SPEED VIOLATION | 1 |
| | 12495733 |

4. This next query will show which NY county had the most parking citations in descending order.
5. Enter the following into your hive:

```
select violation_county, COUNT(violation_county) AS Sum_of_Violation_County from parking_violations_issued Group By violation_county Order By Sum_of_Violation_County Desc LIMIT 13;
```

6. It will display the result something like follows:

| violation_county | sum_of_violation_county |
|---|---|
| NY | 3260490 |
| K | 1889613 |
| BX | 1722022 |
| Q | 1697621 |
| BK | 1572434 |
| QN | 1488229 |
| MN | 385590 |
| ST | 313313 |
| R | 125529 |
| | 40887 |
| QNS | 2 |
| KINGS | 1 |
| P | 1 |

7. This following query will show us which violation codes were most violated in descending order.
8. Enter the following into your hive:

```
select violation_code, COUNT(violation_code) AS Sum_Of_Violation_Code from
```

```
parking_violations_issued Group By violation_code Order By Sum_Of_Violation_Code Desc LIMIT 13;
```

**9.** It will display the result something like follows:

| violation_code | sum_of_violation_code |
|---|---|
| 36 | 3809496 |
| 21 | 1420067 |
| 38 | 970184 |
| 14 | 863756 |
| 20 | 617593 |
| 40 | 519504 |
| 46 | 476743 |
| 71 | 395889 |
| 37 | 370030 |
| 7 | 363537 |
| 19 | 260171 |
| 5 | 256434 |
| 70 | 236541 |

**10.** Lastly, The following will show which Issuing Agency gave the most parking tickets in the city of New York.

**11.** Enter the following into your hive:

```
select issuing_agency, count(issuing_agency) AS Sum_Of_Issuing_Agency from parking_violations_issued Group By issuing_agency Order By Sum_Of_Issuing_Agency Desc LIMIT 13;
```

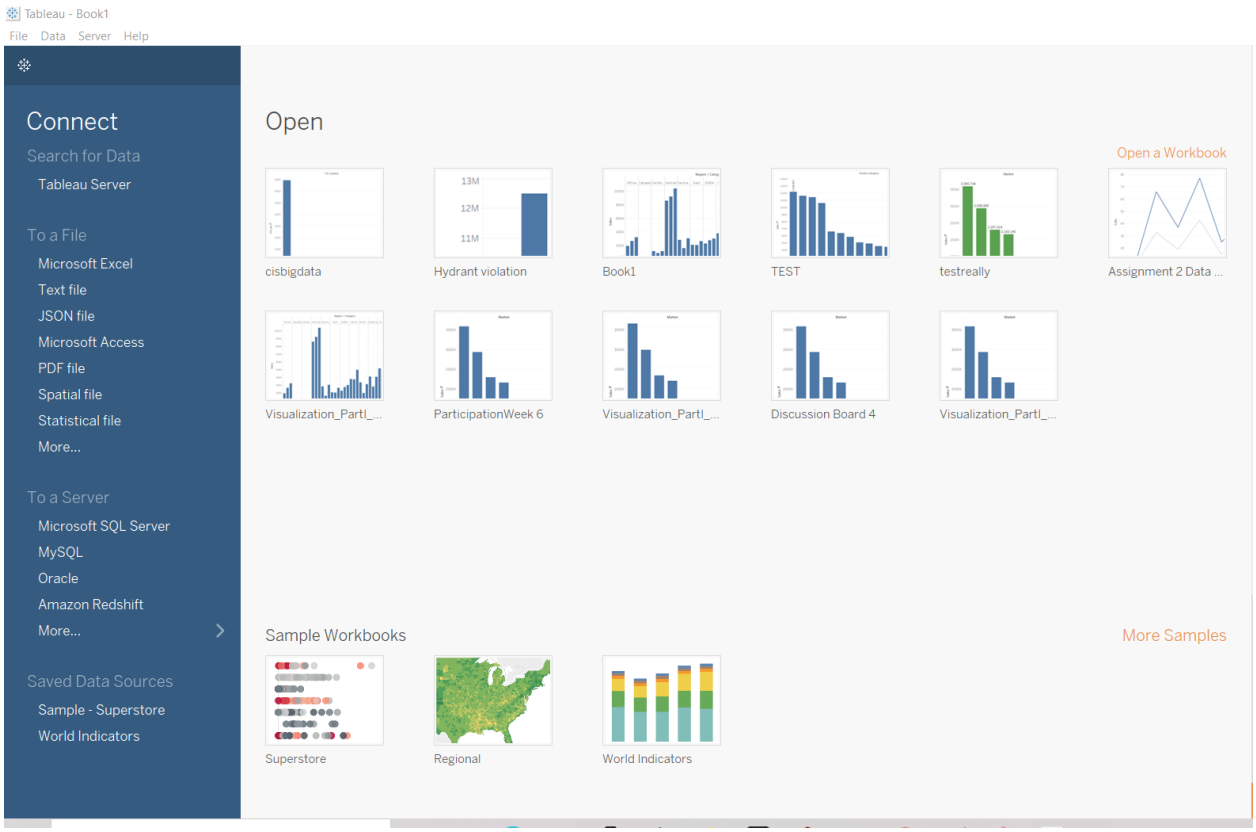
**12.** It will display the result something like follows:

| issuing_agency | sum_of_issuing_agency |
|---|---|
| T | 7244161 |
| V | 4453098 |
| P | 448096 |
| S | 317920 |
| X | 20911 |
| K | 7992 |
| A | 1370 |
| F | 430 |
| N | 400 |
| C | 316 |
| H | 295 |
| U | 203 |
| 1 | 200 |

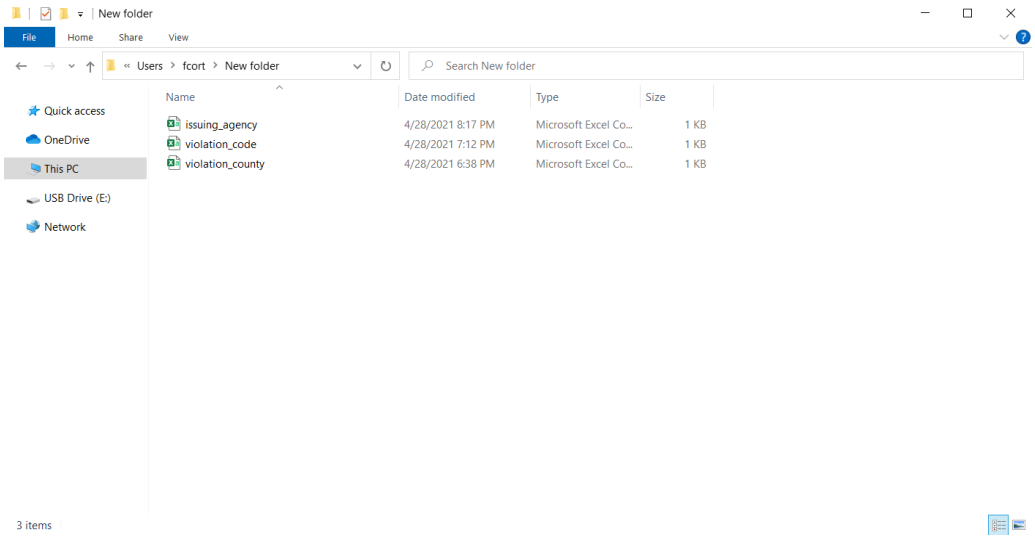
# Step 5: Visualization

### Using Tableau for Visualization of the CSV outputs

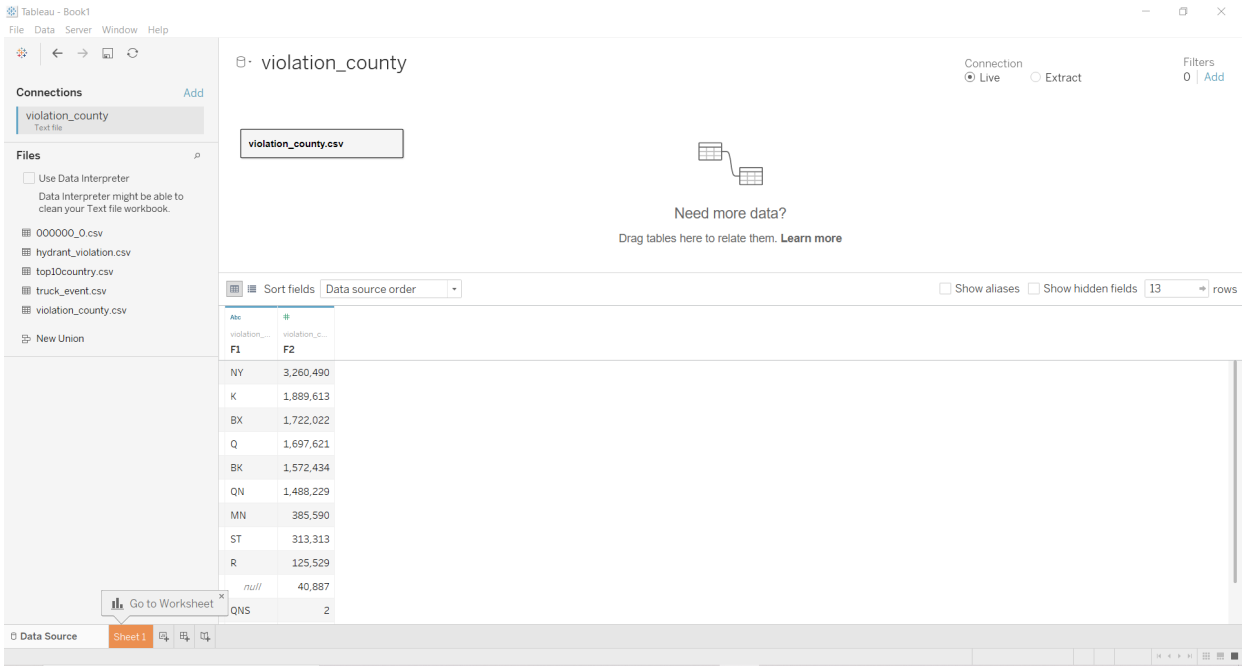
1. Open Tableau to open file download from HDFS:
    a. Select **Text File** under **To a File** to work on a file.




2. Select file download from HDFS.



3. Select **Sheet 1** on the bottom left of the screen to start working on graphs.




**For the NY county that receive the most parking tickets in a bar chart**

1. In hive write the following command to save the Violation County

```
insert overwrite directory '/user/fcortes6/' row format delimited fields terminated by ',' select violation_county, COUNT(violation_county) AS Sum_of_Violation_County from parking_violations_issued Group By violation_county Order By Sum_of_Violation_County Desc LIMIT 13;
```

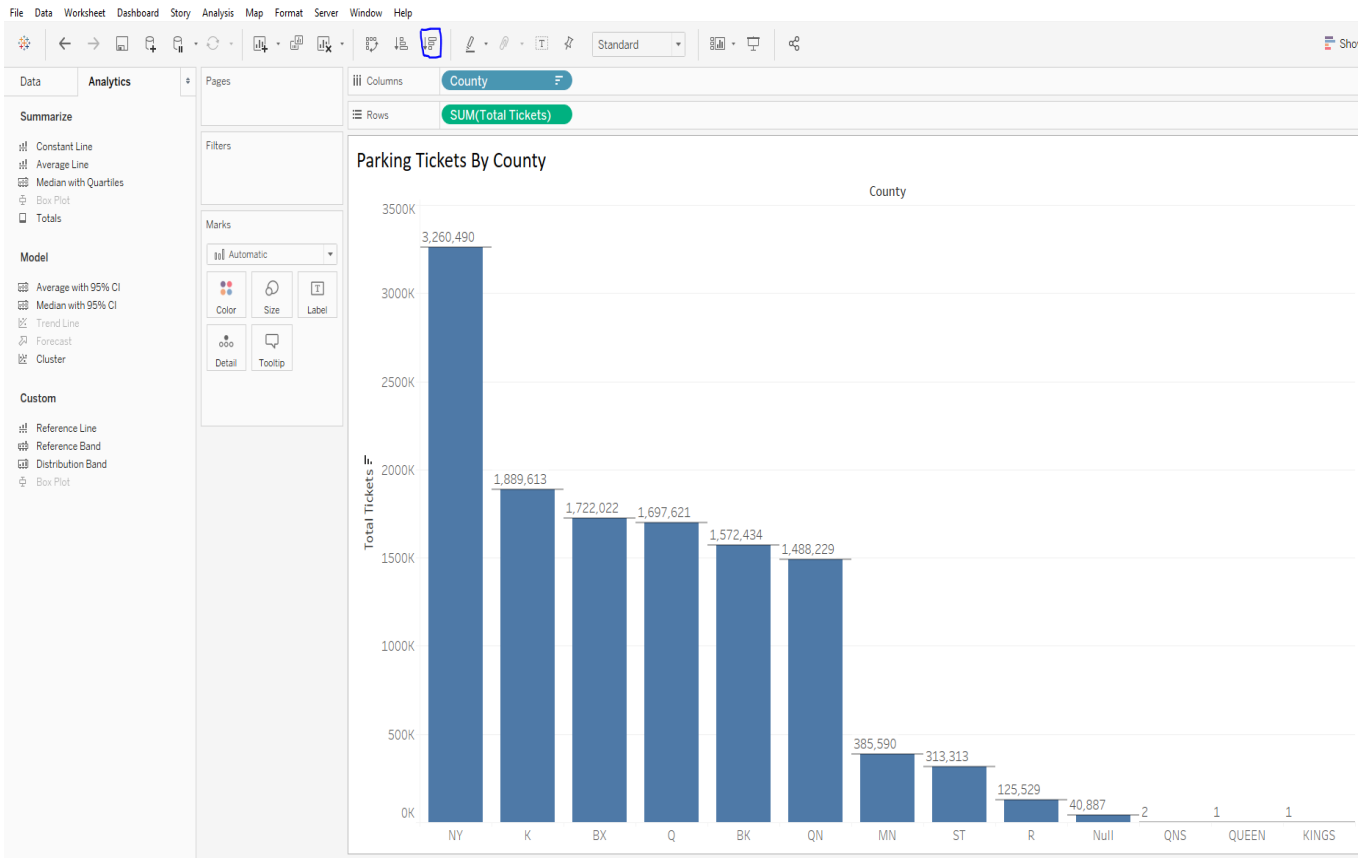
2. Download it by saving it as violation_county and enter the following:

```
hdfs dfs -get 000000_0 violation_county
```

3. Download file using scp:

```
scp fcortes6@220.116.230.21:/home/fcortes6/violation_county violation_county.csv
```

4. Open file on Tableau. Move **County** to columns and **Total Tickets** to rows. Sort by descending order button shown with blue square.




**For which Parking Violation Code is the most violated on a Treemaps.**

1. In hive write the following command to save violation_code :

```
insert overwrite directory '/user/fcortes6/' row format delimited fields terminated by ',' select violation_code, COUNT(violation_code) AS Sum_Of_Violation_Code from parking_violations_issued Group By violation_code Order By Sum_Of_Violation_Code Desc LIMIT 13;
```

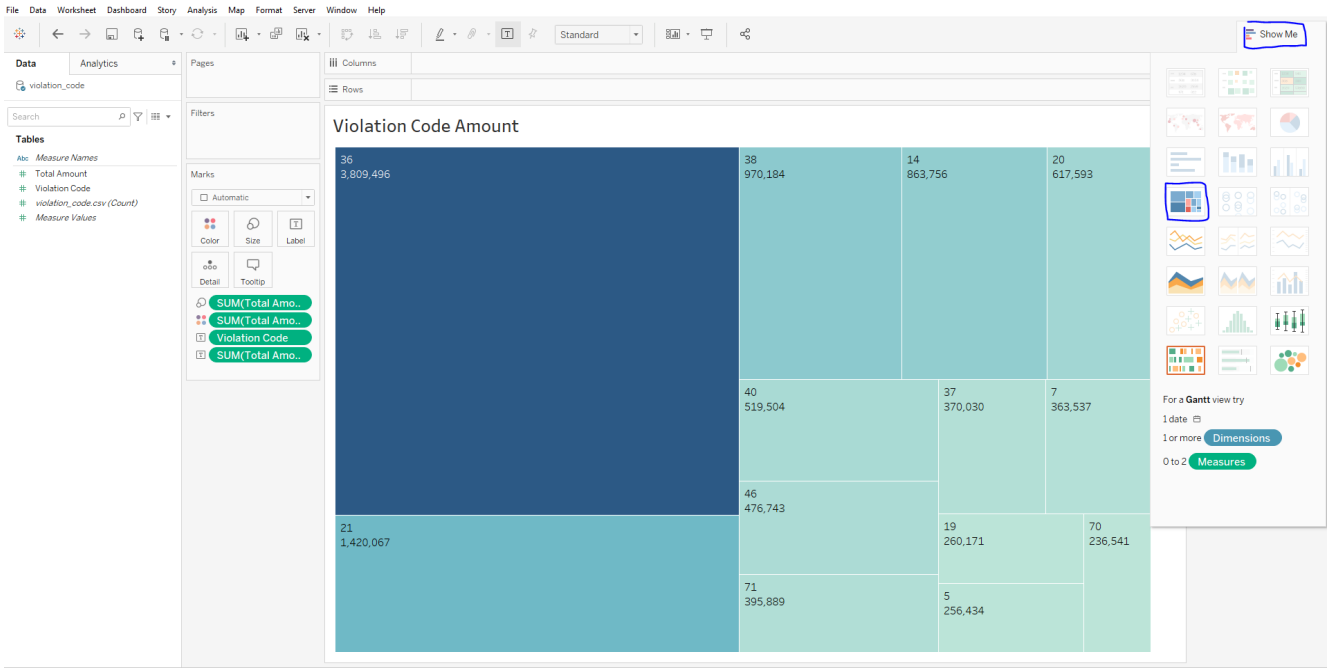
2. Download it by saving it as violation_code and enter the following:

```
hdfs dfs -get 000000_0 violation_code
```

3. Download file using scp:

```
scp fcortes6@220.116.230.21:/home/fcortes6/violation_code violation_code.csv
```

4. Open file on Tableau. Move **Violation Code** to columns and **Total Amount** to rows. Select **Treemaps** on the top right corner of the page marked with blue square




**For which parking issuing agency give the most tickets**

1. In hive write the following command to save **Issuing_Agency**:

```
insert overwrite directory '/user/fcortes6/' row format delimited fields terminated by ',' select issuing_agency, count(issuing_agency) AS Sum_Of_Issuing_Agency from parking_violations_issued Group By issuing_agency Order By Sum_Of_Issuing_Agency Desc LIMIT 13;
```

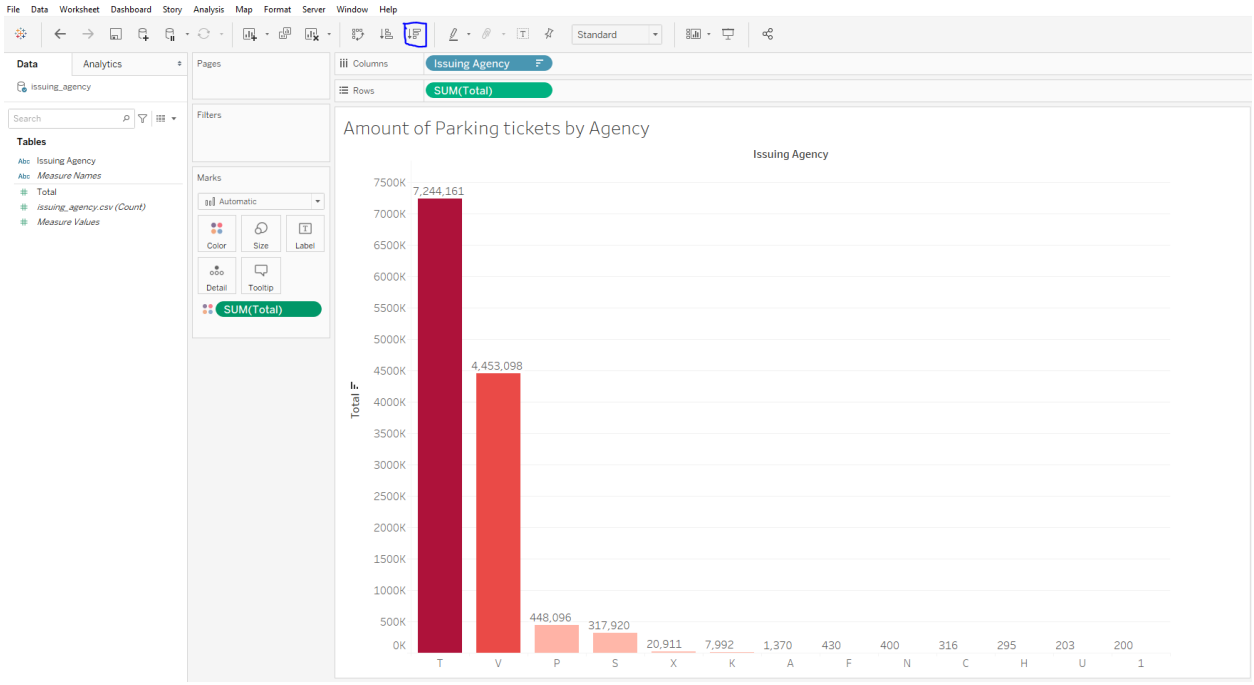
2. Download it by saving it as **Issuing_Agency** and enter the following:

```
hdfs dfs -get 000000_0 issuing_agency
```

3. Download the file using scp:

```
scp fcortes6@220.116.230.21:/home/fcortes6/issuing_agency issuing_agency.csv
```

4. Open file on Tableau. Move **Issuing Agency** and **Total** to rows. Sort by descending order button shown with blue square.




# Step 6: Geo Spatial Mapping

**For which STATE OF PLATE REGISTRATION is given the most parking tickets**

1. In hive write the following command to save **Registration_State:**

```
insert overwrite directory '/user/fcortes6/' row format delimited fields terminated by ',' select Registration_State, count(Registration_State) AS Sum_Of_Registration_State from parking_violations_issued GROUP By Registration_State Order By Sum_of_Registration_State Desc;
```

2. Download it by saving it as **Registration_State** and enter the following:

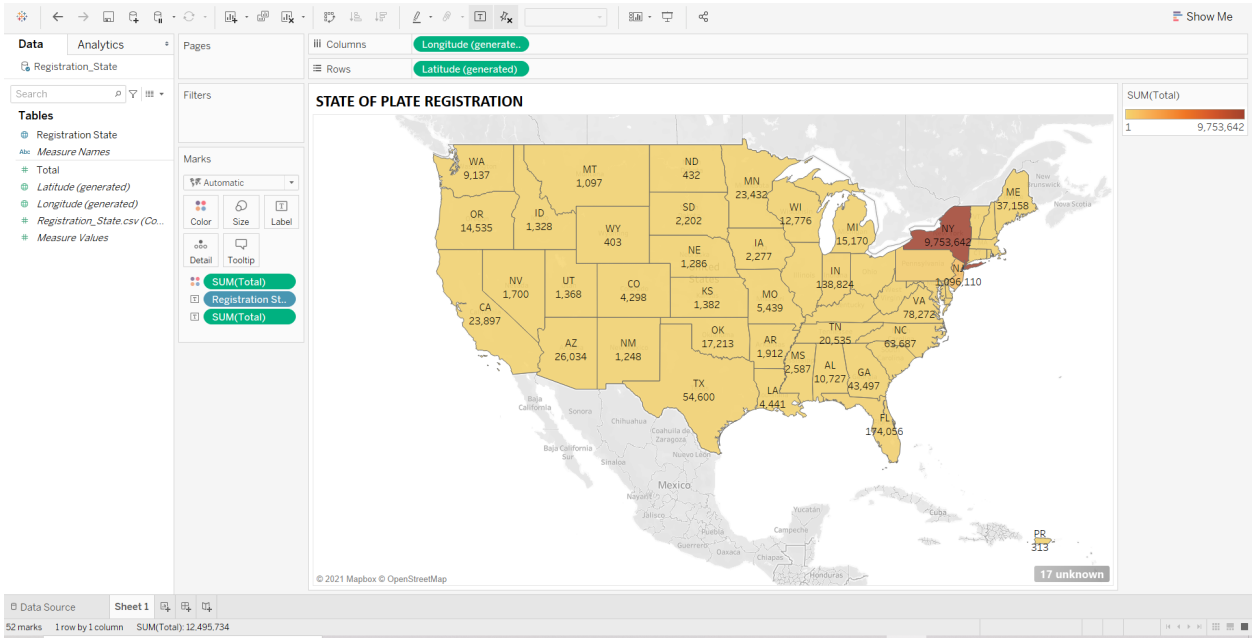
```
hdfs dfs -get 000000_0 Registration_State
```

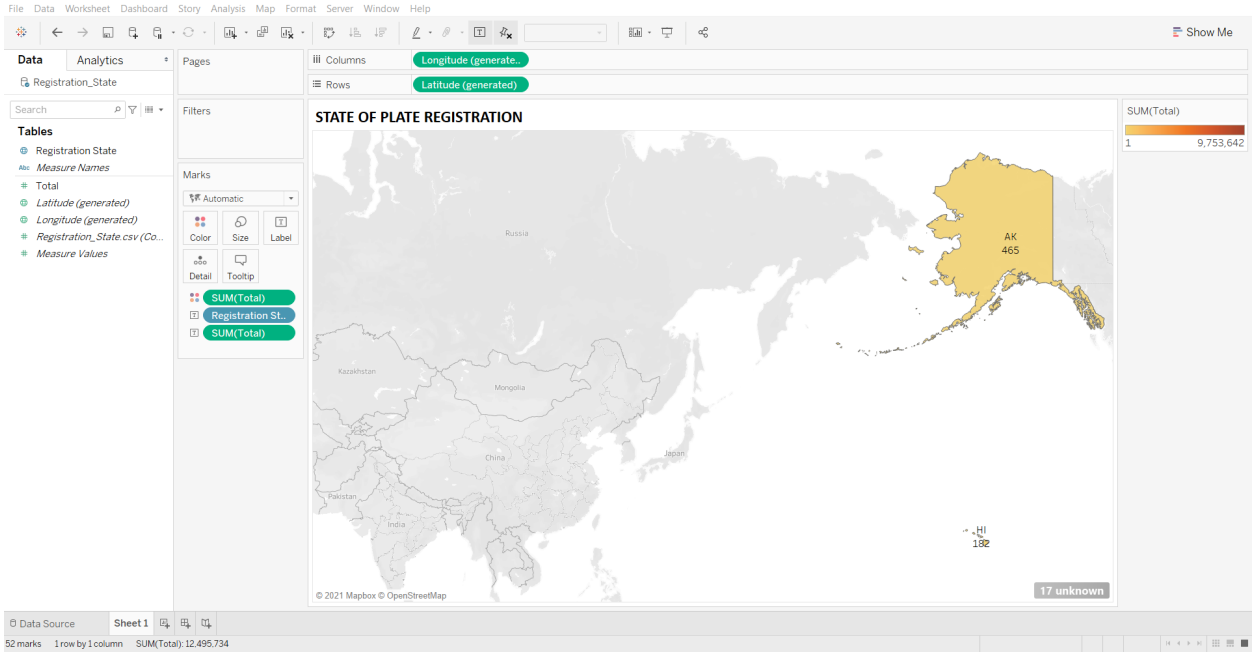
3. Download the file using scp:

```
scp fcortes6@220.116.230.21:/home/fcortes6/Registration_State Registration_State.csv
```

4. Open file on Tableau. Move **Registration State** and **Total** to rows. You can select **Show Me** and its Geo Map to display the total amount of tickets given to state of plate

registration outside of New York. In order to make geo data mapped into the map. the dropbox of **Registration State** dimension should have:  geographic role > State/Province



# References


1. https://data.cityofnewyork.us/City-Government/Parking-Violations-Issued-Fiscal-Year-2020/p7t3-5i9s
2. https://hadoop.apache.org/
3. https://help.tableau.com/current/pro/desktop/en-us/shortcut.htm


**This is the end of the lab.**