

CIS4560 Term Project Tutorial

Authors: [Joey Tran](#); [Garrett Tice](#); [Brandon Ascencio](#); [Kyle Alfonso](#); [Michael Campos](#)

Instructor: [Jongwook Woo](#)

Date: 12/7/2020

Lab Tutorial

Michael Campos (mcampo61@calstatela.edu)

Garrett Tice (gtice@calstatela.edu)

Joey Tran (jtran66@calstatela.edu)

Brandon Ascencio (bascenc8@calstatela.edu)

Kyle Alfonso (kalfons@calstatela.edu)

12/7/2020

NYC 311 Service Request Data Analysis using Hive & Tableau

Objectives

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

- Download files into Apache Hadoop
- Create Hive tables to query data set
- Create queries to analyze the data
- Use Tableau to create visualization of analysis

Platform Spec

- Oracle BDCE
- CPU Speed: 2.20GHz
- # of CPU cores: 12 OCPUs
- # of nodes: 3
- Total Memory Size: 180 GB
- Storage Size: 957 GB

Prerequisites

- Tableau

Step 1: Downloading and Uploading Data Set

This step will be to download the provided data set for NYC 311 Service request into our Oracle server. From there we can upload it to the hadoop server for further analysis.

1. Open your computer's command prompt or terminal and SSH into oracle server.

```
ssh [your_username]@129.150.64.74
```

2. Change directory to a folder that can handle large files. For this example we will be using /dev/shm directory since our home folder may not have enough storage space.

```
$ cd /dev/shm
```

3. Download the data set from amazon S3 and extract.

```
$ wget  
https://s3-us-west-1.amazonaws.com/cis.group.project/311\_Service\_Requests.zip
```

```
Last login: Sun Dec  6 21:35:54 on ttys001
[joeytran@Joey's-MacBook-Pro ~ % ssh jtran66@129.150.64.74
-- WARNING -- This system is for the use of authorized users only. Individuals
using this computer system without authority or in excess of their authority
are subject to having all their activities on this system monitored and
recorded by system personnel. Anyone using this system expressly consents to
such monitoring and is advised that if such monitoring reveals possible
evidence of criminal activity system personnel may provide the evidence of such
monitoring to law enforcement officials.

[jtran66@129.150.64.74's password:
Last login: Mon Dec  7 05:36:01 2020 from 096-041-016-155.res.spectrum.com
[-bash-4.1$ cd /dev/shm
[-bash-4.1$ wget https://s3-us-west-1.amazonaws.com/cis.group.project/311_Service]
_Requests.zip
--2020-12-07 05:39:53-- https://s3-us-west-1.amazonaws.com/cis.group.project/31
1_Service_Requests.zip
Resolving s3-us-west-1.amazonaws.com... 52.219.116.200
Connecting to s3-us-west-1.amazonaws.com|52.219.116.200|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2270670635 (2.1G) [application/zip]
Saving to: "311_Service_Requests.zip"

54% [=====] 1,246,531,135 27.8M/s  eta 47s
```

```
$ unzip 311_Service_Requests.zip
```

4. Create folders in hadoop for data set and upload.

```
$ hdfs dfs -mkdir Services

$ hdfs dfs -put
311_Service_Requests_from_2010_to_Present.csv Services
```

5. Change file permissions so that it can be used. Be sure to include the . at the end.

```
$ hdfs dfs -chmod -R o+w .
```

6. Check to see if files are inside the Services folder.

```
$ hdfs dfs -ls Services
```

```
-bash-4.1$ hdfs dfs -chmod -R o+w .
[-bash-4.1$ hdfs dfs -ls Services
Found 1 items
-rw-r--rw-  2 jtran66 hdfs 13336588760 2020-12-07 05:47 Services/311_Service_Re
quests_from_2010_to_Present.csv
-bash-4.1$
```

Step 2: Create Hive Tables to Query

In this step, we will create Hive tables from our uploaded data set.

1. Type `beeline` to open the Hive interface.

```
$ beeline
```

2. Run the following command to access the class database. There is no password so press enter when prompted.

```
!connect
jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigdai-nov-bdcscse-2:
2181,bigdai-nov-bdcscse-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcscse_admin
put: `Services/311_Service_Requests_from_2010_to_Present.csv': File exists
-bash-4.1$ hdfs dfs -chmod -R o+w .
-bash-4.1$ hdfs dfs -ls Services
Found 2 items
-rw-r--rw- 2 jtran66      hdfs 13336588760 2020-12-07 05:47 Services/311_Service_Requests_from_2010_to_Present.csv
drwxr-xrwx - bdcscse_admin hdfs          0 2020-12-20 02:28 Services/complaints.txt
-bash-4.1$ beeline
WARNING: Use "yarn jar" to launch YARN applications.
Beeline version 1.2.1000.2.4.2.0-258 by Apache Hive
beeline> !connect jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigdai-nov-bdcscse-2:2181,bigdai-nov-bdcscse-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcscse_admin
Connecting to jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigdai-nov-bdcscse-2:2181,bigdai-nov-bdcscse-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive
Enter password for jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigdai-nov-bdcscse-2:2181,bigdai-nov-bdcscse-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive:
Connected to: Apache Hive (version 1.2.1000.2.4.2.0-258)
Driver: Hive JDBC (version 1.2.1000.2.4.2.0-258)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigdai-nov-bdcscse-2:2181,bigdai-nov-bdcscse-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive;
```

3. View and use your database.

```
beeline> show databases;
beeline> use [your_username];
```

```
[0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigdai-nov-bdcscse-2:2181,bigdai-nov-bdcscse-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive] beeline> show databases;
+-----+-----+
| database_name |
+-----+-----+
| amarro15      |
| apaul11       |
| arosali        |
| asekows        |
| asekowstest   |
| atran80       |
| bascenc8      |
| bmendo48      |
| bsangal        |
| default        |
| dkang10       |
| emanuki        |
| fchen26        |
| fmamagh2      |
| groupthree    |
| gtice          |
| hparekh2      |
| jcorte130     |
| jdiep5         |
| jkao           |
```

4. Copy and paste the following code to create a table name Service. The table created is from the CSV file using the Hive query. Be sure to change the highlighted red to your user name, so that it will save to the correct folder.

```

DROP TABLE IF EXISTS Services;
--create the hours table on space-separated building data
CREATE EXTERNAL TABLE IF NOT EXISTS Services(Unique_Key
string,
Created_Date date,
Closed_Date date,
Agency string,
Agency_Name string,
Complaint_Type string,
Descriptor string,
Location_Type string,
Incident_Zip string,
Incident_Address string,
Street_Name string,
Cross_Street_1 string,
Cross_Street_2 string,
Intersection_Street_1 string,
Intersection_Street_2 string,
Address_Type string,
City string,
Landmark string,
Facility_Type string,
Status string,
Due_Date date,
Resolution_Description string,
Resolution_Action_Updated_Date date,
Community_Board string,
BBL string,
Borough string,
X_Coordinate_State_Plane int,
Y_Coordinate_State_Plane int,
Open_Data_Channel_Type string,
Park_Facility_Name string,
Park_Borough string,
Vehicle_Type string,
Taxi_Company_Borough string,
Taxi_Pick_Up_Location string,
Bridge_Highway_Name string,
Bridge_Highway_Direction string,
Road_Ramp string,
Bridge_Highway_Segment string,
Latitude double,
Longitude double,
Location string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/bascenc8/Services'
TBLPROPERTIES ('skip.header.line.count'='1');

```

```

0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> X_Coordinate_State_Plane int,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Y_Coordinate_State_Plane int,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Open_Data_Channel_Type string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Park_Facility_Name string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Park_Borough string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Vehicle_Type string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Taxi_Company_Borough string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Taxi_Pick_Up_Location string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Bridge_Highway_Name string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Bridge_Highway_Direction string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Road_Ramp string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Bridge_Highway_Segment string,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Latitude double,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Longitude double,
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> Location string)
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> STORED AS TEXTFILE LOCATION '/user/r/jtran66/Services'
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> TBLPROPERTIES ('skip.header.line.count'='1');
No rows affected (0.213 seconds)
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> ■

```

5. We will then create a complaints table using specific columns from the data set and overwrite with a new Hive query.

```

DROP TABLE IF EXISTS Complaints;
CREATE EXTERNAL TABLE IF NOT EXISTS Complaints(agency_name
string, Complaint_type string, NumberofComplaints int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION
'/user/bascenc8/Services/complaints.txt';

INSERT OVERWRITE TABLE Complaints
select agency_name, complaint_type, count(complaint_type)
as NumberofComplaints from services group by agency_name,
complaint_type having count(complaint_type) > 5 order by
NumberofComplaints DESC;

```

```

INFO : Map 1: 2/2      Reducer 2: 124(+5)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 130(+4)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 133(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 138(+5)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 143(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 148(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 153(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 159(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 164(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 169(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 173(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 179(+5)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 183(+6)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 188(+5)/199  Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 194(+5)/199  Reducer 3: 0(+1)/1
INFO : Map 1: 2/2      Reducer 2: 199/199      Reducer 3: 0(+1)/1
INFO : Map 1: 2/2      Reducer 2: 199/199      Reducer 3: 1/1
INFO : Loading data to table jtran66.complaints from hdfs://mycluster/user/r/jtran66/Services/complaints.txt/.hive-staging_hive_2020-12-20-05-55_095_806660367
7787211131-2657/-ext-10000
INFO : Table jtran66.complaints stats: [numFiles=1, numRows=1696, totalSize=872
98, rawDataSize=85602]
No rows affected (114.22 seconds)
0: jdbc:hive2://bigdai-nov-bdcscce-1:2181,bigd> ■

```

6. Check your database tables to see if you have the following tables: Services, Complaints

```
beeline> show tables;
```

```
INFO : Map 1: 2/2      Reducer 2: 169(+6)/199 Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 173(+6)/199 Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 179(+5)/199 Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 183(+6)/199 Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 188(+5)/199 Reducer 3: 0/1
INFO : Map 1: 2/2      Reducer 2: 194(+5)/199 Reducer 3: 0(+1)/1
INFO : Map 1: 2/2      Reducer 2: 199/199      Reducer 3: 0(+1)/1
INFO : Map 1: 2/2      Reducer 2: 199/199      Reducer 3: 1/1
INFO : Loading data to table jtran66.complaints from hdfs://mycluster/user/jtran66/Services/complaints.txt.hive-staging_hive_2020-12-20_05-55_095_806660367
7787211131-2657/-ext-10000
INFO : Table jtran66.complaints stats: [numFiles=1, numRows=1696, totalSize=87298, rawDataSize=85602]
No rows affected (114.22 seconds)
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd>
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> show tables;
+-----+-----+
| tab_name | |
+-----+-----+
| complaints | |
| services   | |
+-----+-----+
2 rows selected (0.313 seconds)
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> ■
```

Step 3: Hive Query Analysis

This step will involve running queries from the created tables so that we can extrapolate useful information.

1. Let's view some data from the newly created table within beeline. Enter the following code to display 10 rows of data from the created complaints table.

```
SELECT *
FROM complaints
LIMIT 10;
```

```
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> SELECT *
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> FROM complaints
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> LIMIT 10;
+-----+-----+-----+-----+
| complaints.agency_name | complaints.complaint_type | complaints.numberofcomplaints | 
+-----+-----+-----+-----+
| New York City Police Department | Noise - Residential | 2189115 | 
| Department of Housing Preservation and Development | HEAT/HOT WATER | 1362946 | 
| New York City Police Department | Illegal Parking | 1091003 | 
| New York City Police Department | Blocked Driveway | 1025839 | 
| Department of Transportation | Street Condition | 1012885 | 
| Department of Transportation | Street Light Condition | 974690 | 
| Department of Housing Preservation and Development | HEATING | 887850 | 
| Department of Housing Preservation and Development | PLUMBING | 739135 | 
| Department of Environmental Protection | Water System | 694961 | 
| New York City Police Department | Noise - Street/Sidewalk | 674609 | 
+-----+-----+-----+-----+
10 rows selected (0.259 seconds)
```

2. The following query will contain 10 rows of data from the complaints table that belong to the New York City Police Department.

```
SELECT *
FROM complaints
WHERE agency_name
LIKE 'New York City Police Department'
LIMIT 10;
```

```
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> SELECT *
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> FROM complaints
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> WHERE agency_name
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> LIKE 'New York City Police Department'
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> LIMIT 10;
+-----+-----+-----+
|   complaints.agency_name |   complaints.complaint_type |   complaints.numberofcomplaints |
+-----+-----+-----+
| New York City Police Department | Noise - Residential | 2189115
| New York City Police Department | Illegal Parking | 1091003
| New York City Police Department | Blocked Driveway | 1025839
| New York City Police Department | Noise - Street/Sidewalk | 674609
| New York City Police Department | Noise - Commercial | 391100
| New York City Police Department | Noise - Vehicle | 294781
| New York City Police Department | Derelict Vehicle | 207810
| New York City Police Department | Non-Emergency Police Matter | 151408
| New York City Police Department | Abandoned Vehicle | 55362
| New York City Police Department | Animal Abuse | 52177
+-----+-----+-----+
10 rows selected (0.238 seconds)
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> ■
```

3. This query will build upon the last and find the number of complaints and complaint types that the New York City Police Department handled.

```
SELECT complaint_type, numberofcomplaints
FROM complaints
WHERE agency_name
LIKE 'New York City Police Department'
LIMIT 10;
```

```

0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> SELECT complaint_type, numberofcomplaints
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> FROM complaints
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> WHERE agency_name
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> LIKE 'New York City Police Department'
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> LIMIT 10;
+-----+-----+
|   complaint_type      | numberofcomplaints |
+-----+-----+
| Noise - Residential | 2189115          |
| Illegal Parking     | 1091003          |
| Blocked Driveway    | 1025839          |
| Noise - Street/Sidewalk | 674609          |
| Noise - Commercial  | 391100           |
| Noise - Vehicle     | 294781           |
| Derelict Vehicle    | 207810           |
| Non-Emergency Police Matter | 151408          |
| Abandoned Vehicle   | 55362            |
| Animal Abuse         | 52177            |
+-----+-----+
10 rows selected (0.229 seconds)
0: jdbc:hive2://bigdai-nov-bdcscse-1:2181,bigd> 

```

4. We can run the following query to see which agency handled the most complaints.

```

SELECT agency_name, COUNT(agency_name) AS
complaints_handled
FROM complaints
GROUP BY agency_name
ORDER BY complaints_handled DESC
LIMIT 10;

```

```

INFO : Map 1: 0/1 Reducer 2: 0/1 Reducer 3: 0/1
INFO : Map 1: 0(+1)/1 Reducer 2: 0/1 Reducer 3: 0/1
INFO : Map 1: 1/1 Reducer 2: 0(+1)/1 Reducer 3: 0/1
INFO : Map 1: 1/1 Reducer 2: 1/1 Reducer 3: 0(+1)/1
INFO : Map 1: 1/1 Reducer 2: 1/1 Reducer 3: 0/1
INFO : Map 1: 1/1 Reducer 2: 1/1 Reducer 3: 1/1
+-----+-----+
|           agency_name           | complaints_handled |
+-----+-----+
| Department of Health and Mental Hygiene | 40               |
| New York City Police Department        | 32               |
| Department of Buildings              | 30               |
| Department of Transportation        | 29               |
| Department of Housing Preservation and Development | 27               |
| Department of Environmental Protection | 18               |
| Department for the Aging             | 18               |
| Manhattan 02                        | 16               |
| BCC - Brooklyn South                | 16               |
| BCC - Bronx                         | 16               |
+-----+-----+
10 rows selected (13.393 seconds)

```

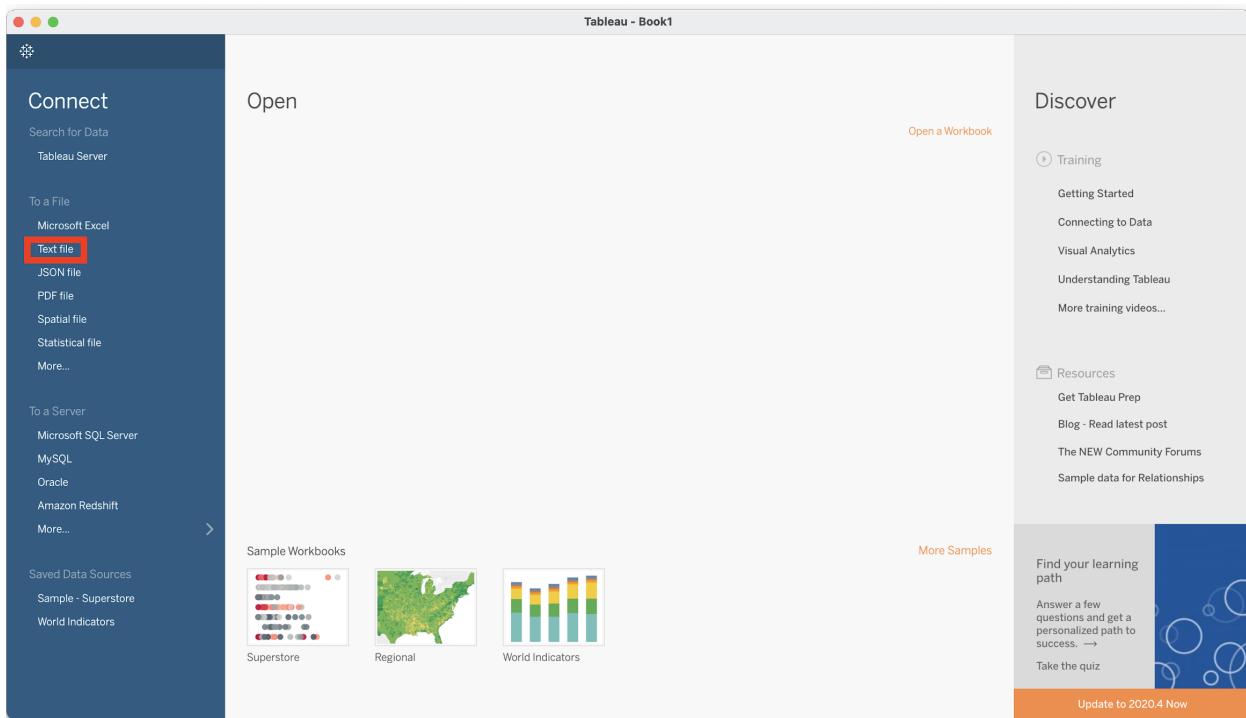
Step 4: Visualization In Tableau

This step involves importing our dataset into Tableau and analyzing our data through multiple visualizations .

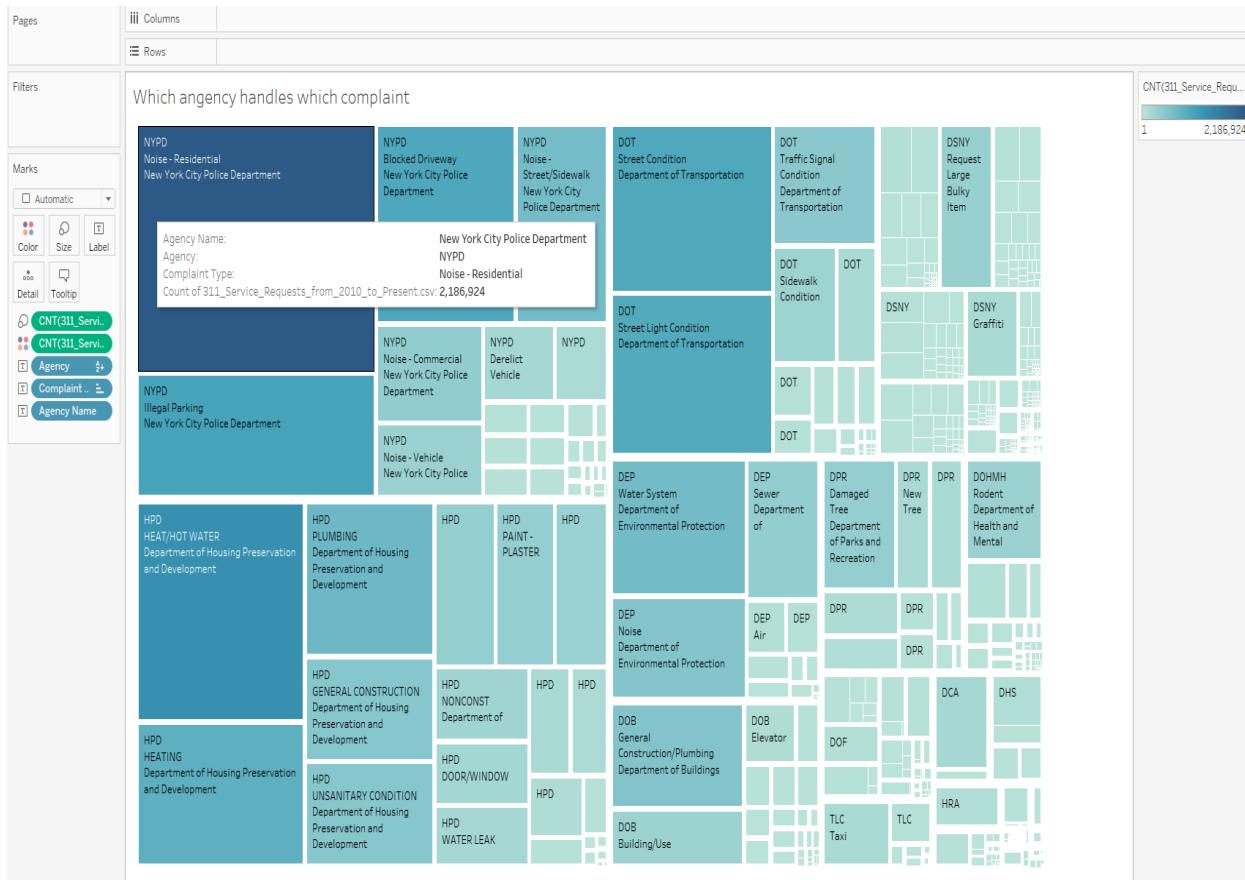
- At your PC with git bash, xterminal, or pscp.exe, you can remotely download the output file "311_Service_Requests_from_2010_to_Present.csv" to your PC to visualize it using Tableau.
NOTE: the following code has "." at the end.

```
scp
[your_username]@129.150.64.74:/dev/shm/311_Service_Request
s_from_2010_to_Present.csv .
```

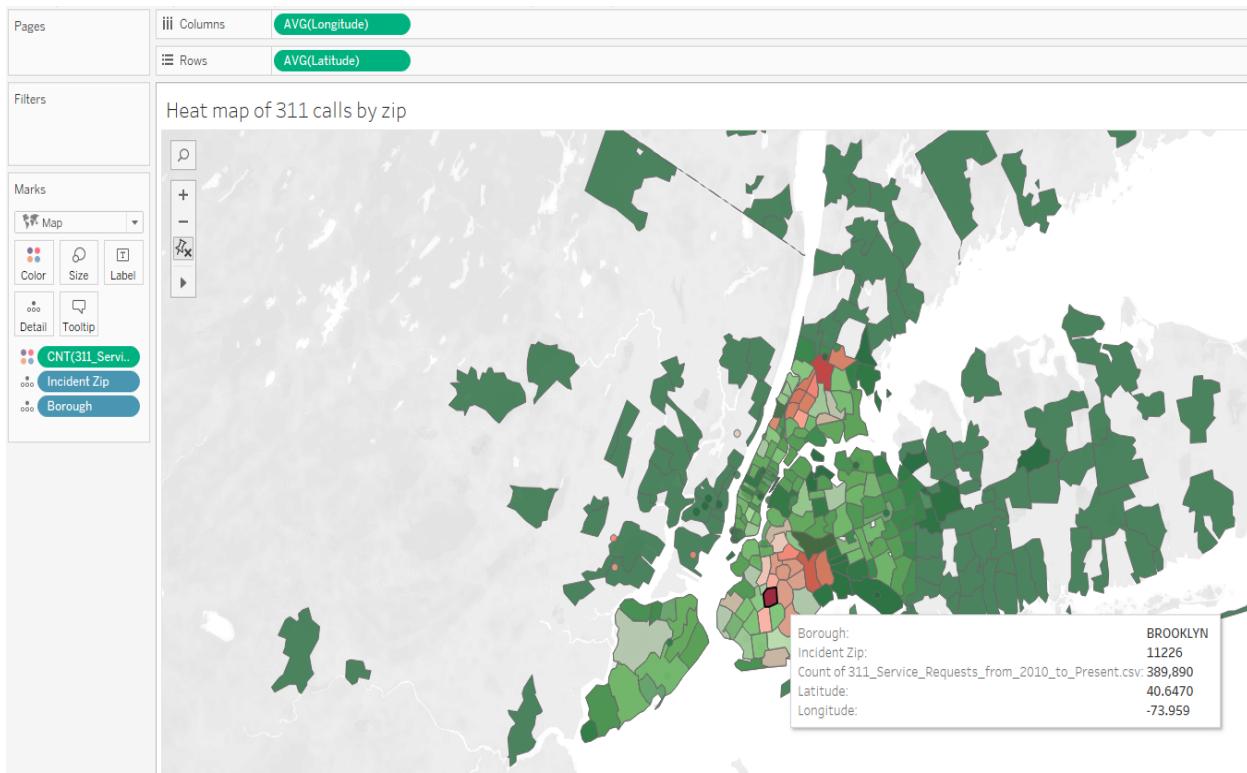
- Open your tableau on your local computer. Select Text File to open your 311_Service_Requests_from_2010_to_Present.csv.



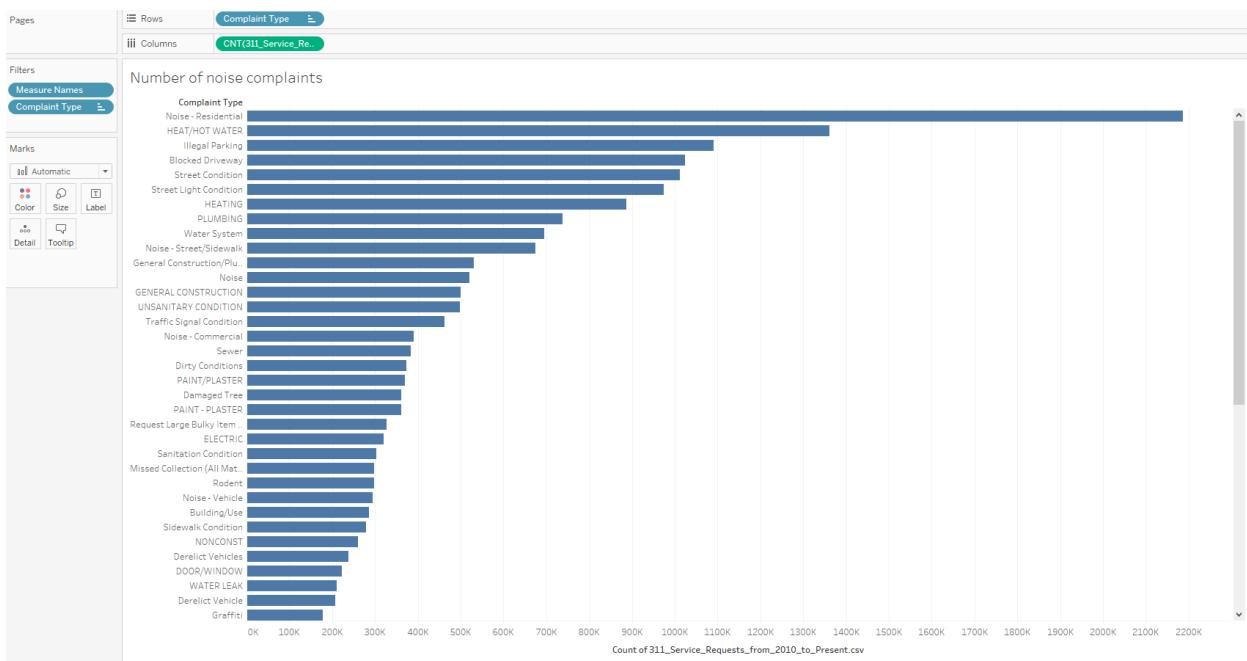
- Select sheet 1 to start working on our visualization.
In the tableau viewer drag and drop *agency* to the *columns* section and then *Count* to the *rows* section. Then make the marks set for count size and color by dragging the count to the appropriate mark icon. Drag and drop *agency*, *complaint type* and *agency name* into the labels icon



4. Make a new worksheet at the bottom of the tableau viewer. Assign the *Longitude* to the *columns* section and then the *latitude* to the *rows* section. Then import your marks on the map by dragging and dropping *count* to *color*, *Zip* to *detail* and *Borough* to *detail*. You should be able to get a heat map resembling the one below.



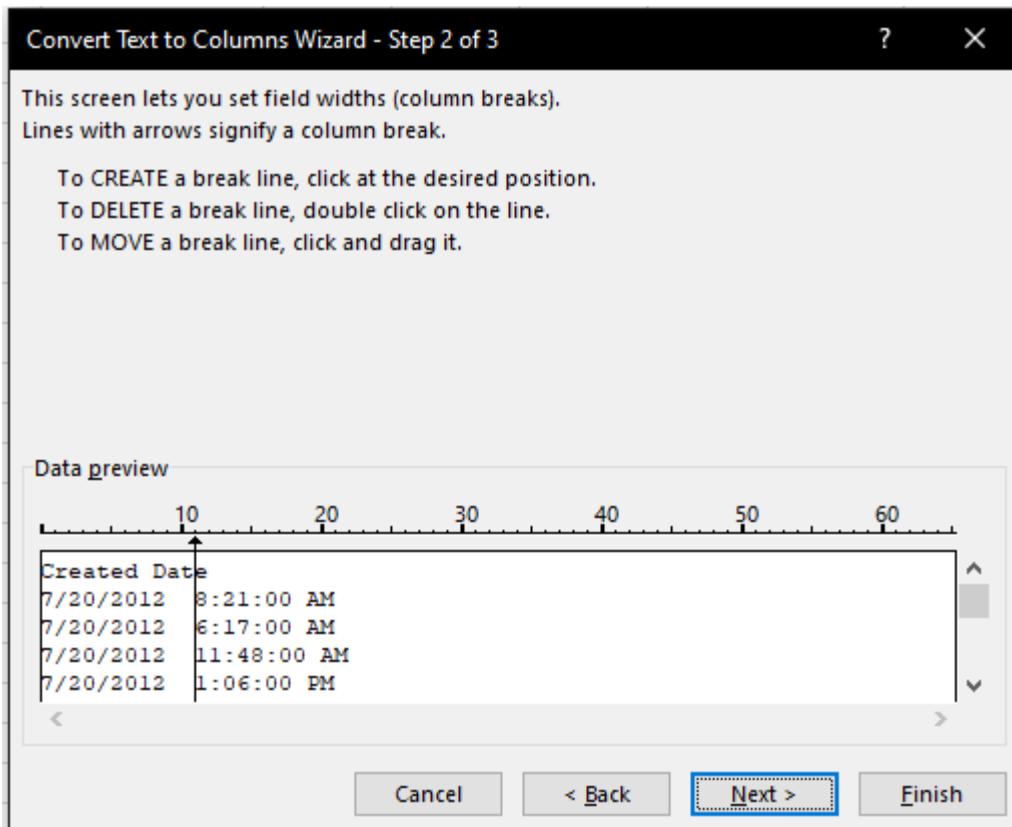
5. On another worksheet simply Drag and drop the *complaint* type into the *Rows* section and the *Count* into the *Columns* section. Then Filter the *complaint* type to a *descending* order. This can be done by clicking the *descending* icon located in the toolbar right below the “window” tab.



Step 5: Geo-temporal Visualization of sample data In Excel

A 3D Geo-Temporal Map will be created using Excel to present us with a timelapse between March 19, 2010 and November 21,2020 for residential noise complaints.

1. Open the `311_Service_Requests_from_2010_to_Present.csv` that you download previously in the step 4 section.
2. While the file is open, select File → Save As... and save the file as a `.xlsx` file type.
3. Close the Csv file, and open the `.XLSX` file you just saved.
4. Once the file is open, select the Closed Date column my clicking on the E right above the header. Right click on any of the records within that column, and select insert. This will create any empty column. Repeat this step one more time so there are two blank columns to the left of the Closed Date column
5. Highlight Created Date column by clicking on the B above the header. With this column highlighted, select the Data tap from the menu at the top and then click the Text to Columns option in the Data Tools section.
6. In the menu that appears, select the `Fixed_width` option and click Next. There will be a bar that you can drag in the next menu. Drag the bar to the right, so that it is touching the timestamp times. Click next, and at the next menu click Finish.

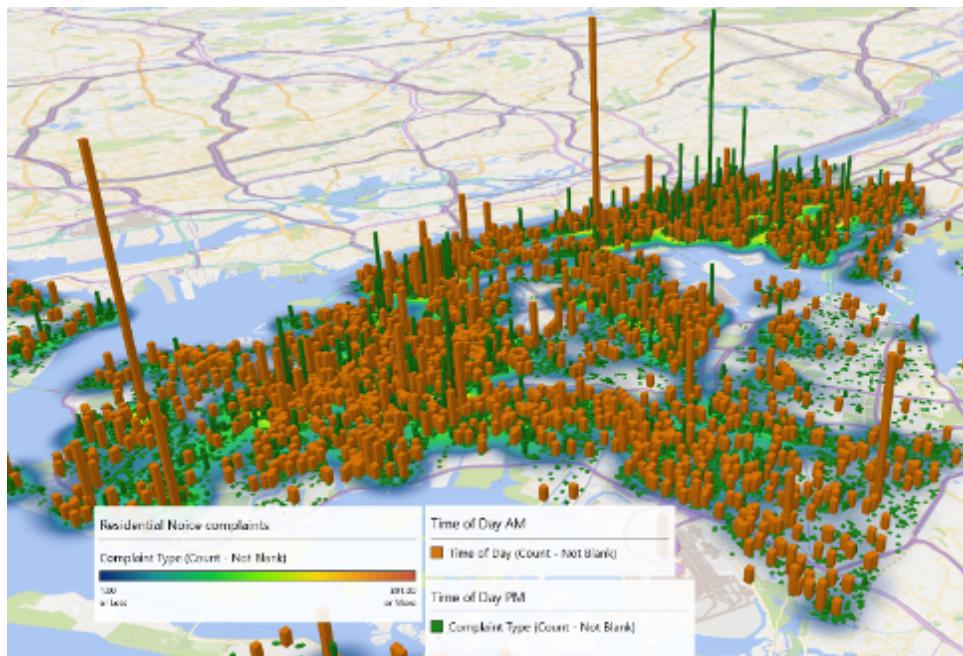


7. You should now see one of the blank columns now has data in it. Select this column that has been filled in by clicking on the C above the header. With this column selected, click the Text to Columns tool again in the top menu under the Data Tools section. Make sure the Delimited option is ticked and click next. Make sure Space is selected in the Delimiters section, and no others are selected before hitting next. Hit Finish when you get to the next menu.
8. In column B select the column, and right click the column to format the cells. Select Date as the category, and select the first option in the Type section. Click ok. Double click on the first cell of the column and change the header to Created Date.
9. In column C, select the column and right click to format the cells. Select the Time category, and the second option in the type section. Click ok. Change the header of the column to Created Time by double clicking on the first cell of the column.
10. Double click the first cell of column D to change the header to Active Hours.
11. Press **ctrl+S** to save your work.
12. Click on any cell within the worksheet, and hit **ctrl+a**. This will select all the cells in the sheet. With all the cells

selected, click on the Insert tab and select Table located just under it. Make sure the My table has headers checkbox is checked and hit ok.

13. Use ctrl+a again to select all the cells. While these are selected, navigate to the Insert tab in the top menu, and click on the 3D Maps menu that is located in the tours section. In the drop down menu, click open. Wait for the tour to open.
 14. Once the 3D Map screen opens, Look to the right section of the screen where there is an Add Layer button. Click the Add Layer button twice to add two more layers.
 15. Select the first layer. In the location field click the plus(+) sign. In the drop down menu, select latitude. Click the plus sign again and select Longitude from the drop down menu. Once both are added you will see drop down menus appear next to them. Click the drop down menu next to Latitude and select latitude from the menu. Do the same for longitude, except selecting longitude from the menu.
- Location**
-
16. Repeat the process in step 15 for layers 2 and 3.
17. On the first layer from the field list, drag the unique key to height and select count- Not Blank in the drop down menu that appears next to it. In the filter section select Active Hours and Complaint Type. For the active hours filtered, Check AM and uncheck PM. In the Complaint Type filter, search for Residential and select the first result. In the layer options, set the height to 141% and thickness to 32%.
18. For Layer 2, repeat step 17. Only this time, make sure that PM is checked and AM is not.
19. For Layer 3, change the visualisation to a heat map by selecting the second furthest right chart option under data. Select Unique Key as the Value and Count-Not Blank from the drop down menu to the side of it. Only add one filter, complaint type, to the layer. Making sure that the residential noise complaint is selected.
20. In the layers section, click the pencil icon to the right of each layer name and rename them as follows. Change Layer 1 to Time of Day AM, and Layer 2 to Time of Day PM. Change the name of Layer 3 to Residential Noise complaints concentration.

21. You should see a map similar to the one below. Colors may vary and you can change some of the layer options as you see fit.



References

-
1. <https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9>