[**C:\Users\jwoo5\AppData\Local\Temp\templateTermTutorial.html**](http://www.calstatela.edu/centers/hipic) **CIS5200 Term Project Tutorial** 

California State University, Los Angeles

**Authors:   Jane Liang, Weiqiang He, Yiqing Qian**

**Instructor: [Jongwook Woo](https://www.linkedin.com/in/jongwook-woo-7081a85)**

**Date: 12/15/2019**

**Lab Tutorial**

Jane Liang ([bliang12@calstatela.edu](mailto:bliang12@calstatela.edu))

Weiqiang He ([whe8@calstatela.edu](mailto:whe8@calstatela.edu))

Yiqing Qian ([yqian6@calstatela.edu](mailto:yqian6@calstatela.edu))

12/15/2019

**311 Service Requests Analysis in New York City**

**Objective:**

This tutorial is about the analysis of 311 Service Requests in New York City Area using Hadoop Cluster in AWS and visualizations using tableau and Excel. You will gain working knowledge about using SCP to upload data file into Hadoop distributed file system in AWS and visualizing the analyzed data in tableau and Excel.

The original data set comprising the details of 311 Service Request in New York City was downloaded from a trusted source (https://data.cityofnewyork.us). This data set has a comprehensive and detailed records of complaint cases filed with 311 service from Jan 2010 to Oct 2019. The data set has 41 columns which mainly include Unique Key, Created Date, Closed Date, Agency, Agency Name, Complaint Type, Descriptor, Location Type, Incident Zip, Incident Address City, Landmark, Facility Type, Status, Due Date, Resolution Description, Resolution Action, Borough, Open Data Channel Type, Latitude, and Longitude etc.

This analysis will give us ideas about the case numbers filed each year and which year has the most cases, the average days to close a case by each year, the most common complaint types, the most common channels used to file a complaint case, and how cases are distributed geographically etc.

**Technologies Used:**

* SCP
* Amazon AWS
* Hadoop distributed file system
* Hive
* Tableau / Excel

**Pre-requisites:**

**Setup Tableau with the License:**

If you have the Tableau software with active license you are good to go. Else you can get the Tableau for students from the below link,

<http://www.tableau.com/academic/students>

You can use your institution ID to register and get free tableau desktop license for 1-year.

**Platform Specification**

The Account accessing Hadoop Cluster on AWS was provided by professor in class. The platform specification are shown below:

|  |  |
| --- | --- |
| Cluster IP: | 34.221.40.43 |
| Number of Nodes in Cluster: | 3 (m3.xlarge) |
| Release Label : | Emr-5.27.0 |
| Hadoop Distribution: | Amazon 2.8.5 |
| Applications: | Hive 2.3.5 |
| Core Number of CPU: | 2 Cores (4 vCPU) |
| CPU Speed: | 2.5 GHz |
| Memory | 15 GiB |
| Instance Storage | 2 x 40 GB |

**Step 1: Import Data Into AWS Cluster**

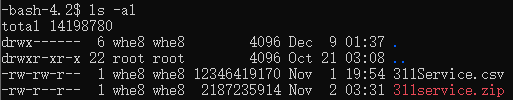
* The download link for the dataset used for our project is given below:

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

* Total Size : 11.6GB
* This dataset is in a single CSV file. Since it is big, before we upload to the AWS cluster, we zip it into 311Service.zip, then use SCP to upload the data to Cluster with IP 34.221.40.43 by the following command in windows command shell:   
  Scp 311Service.zip [whe8@34.221.40.43:/home/whe8/](about:blank)
* Then we logged in the cluster by the following command in the windows command shell:

ssh whe8@34.221.40.43

* After that we unzip the file to 311Service.csv in the cluster.

unzip 311Service.zip  


* Next step, we created a folder CIS5200Project in Hadoop distributed file system to hold the data file by below command:  
   hdfs dfs -mkdir CIS5200Project
* Then we copied the data file to Hadoop distributed file system by below command:  
  hdfs dfs -put 311Service.csv /user/whe8/CIS5200Project/
* After that, we displayed the file to confirm the data file is there.

hdfs dfs -ls /user/whe8/CIS5200Project/



**Step 2: Analyze data in Hive**

We logged in Hive application by:

beeline -u jdbc:hive2://localhost:10000/default -n whe8

In order to analyze the data in Hive, we need to first create an external table to link to the data file we have in the hadoop distributed file system.

USE WHE8;

DROP TABLE IF EXISTS Service;

CREATE EXTERNAL TABLE IF NOT EXISTS Service (

UniqueKey STRING,

CreatedDate STRING,

ClosedDate STRING,

Agency STRING,

AgencyName STRING,

ComplaintType STRING,

Descriptor STRING,

LocationType STRING,

IncidentZip STRING,

IncidentAddress STRING,

StreetName STRING,

CrossStreet1 STRING,

CrossStreet2 STRING,

IntersectionStreet1 STRING,

IntersectionStreet2 STRING,

AddressType STRING,

City STRING,

Landmark STRING,

FacilityType STRING,

Status STRING,

DueDate STRING,

ResolutionDescription STRING,

ResolutionActionUpdatedDate STRING,

CommunityBoard STRING,

BBL STRING,

Borough STRING,

XCoordinate STRING,

YCoordinate STRING,

OpenDataChannelType STRING,

ParkFacilityName STRING,

ParkBorough STRING,

VehicleType STRING,

TaxiCompanyBorough STRING,

TaxiPickUpLocation STRING,

BridgeHighwayName STRING,

BridgeHighwayDirection STRING,

RoadRamp STRING,

BridgeHighwaySegment STRING,

Latitude STRING,

Longitude STRING,

Location STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

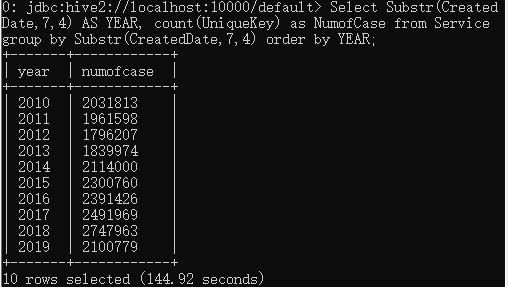
STORED AS TEXTFILE LOCATION '/user/whe8/CIS5200Project'

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

1. **NUMBER OF CASES BY YEAR THROUGH 2010 TO 2019**

Select Substr(CreatedDate,7,4) AS YEAR, count(UniqueKey) as NumofCase from Service group by Substr(CreatedDate,7,4) order by YEAR;

**OUTPUT**



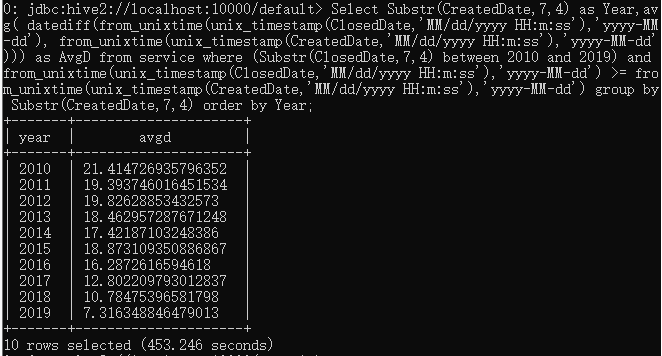
1. **THE AVERAGE NUMBER OF DAYS TO CLOSE A CASE BY YEAR**

The date time format in the original data file cannot be recognized by Hive, therefore two functions were used to convert the date time format to the one which can be recognized.

In addition, the data field of “ClosedDate” is not very clear, a significant part of them are either empty or in the year of 1900, therefore we specified two conditions to exclude those invalid data.

Select Substr(CreatedDate,7,4) as Year,avg( datediff(from\_unixtime(unix\_timestamp(ClosedDate,'MM/dd/yyyy HH:m:ss'),'yyyy-MM-dd'), from\_unixtime(unix\_timestamp(CreatedDate,'MM/dd/yyyy HH:m:ss'),'yyyy-MM-dd'))) as AvgD from service where (Substr(ClosedDate,7,4) between 2010 and 2019) and  from\_unixtime(unix\_timestamp(ClosedDate,'MM/dd/yyyy HH:m:ss'),'yyyy-MM-dd') >= from\_unixtime(unix\_timestamp(CreatedDate,'MM/dd/yyyy HH:m:ss'),'yyyy-MM-dd') group by Substr(CreatedDate,7,4) order by Year;

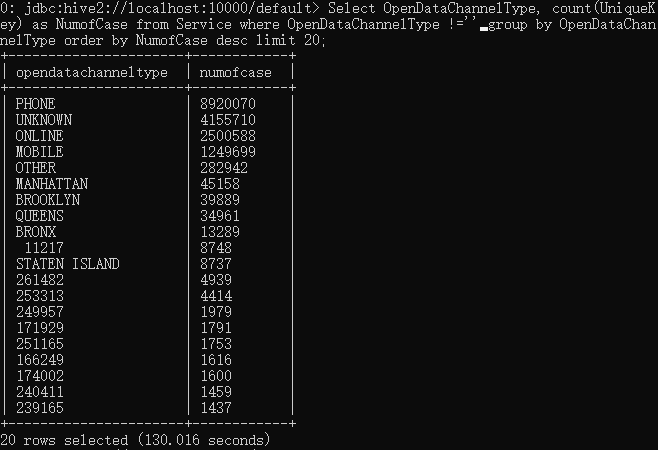
**OUTPUT**



**3)NUMBER OF CASE BY COMPLAINT CHANNEL (TOP 20)**

Select OpenDataChannelType, count(UniqueKey) as NumofCase from Service where OpenDataChannelType !='' group by OpenDataChannelType order by NumofCase desc limit 20;

**OUTPUT**

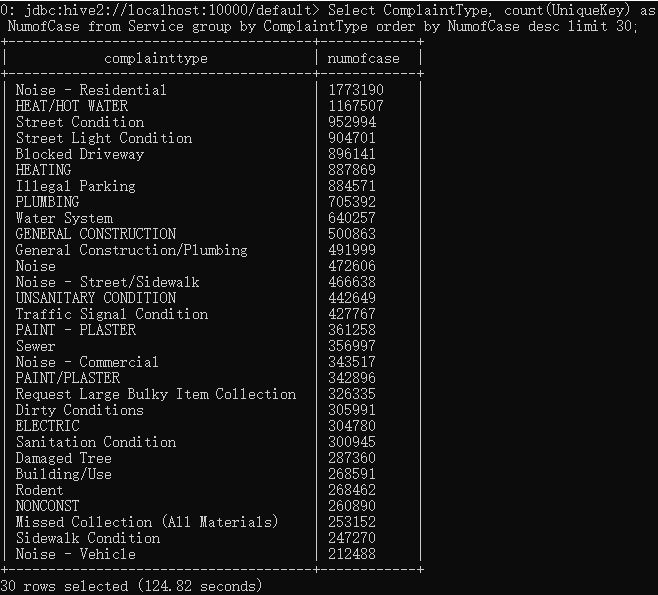


As per the above query result, only PHONE, ONLINE, AND MOBILE are identified, and all others are treated as invalid or unspecified in our analysis and visualization in the later part of this tutorial.

**4-1)NUMBER OF CASE BY COMPLAINT TYPE (TOP 30)**

Select ComplaintType, count(UniqueKey) as NumofCase from Service group by ComplaintType order by NumofCase desc limit 30;

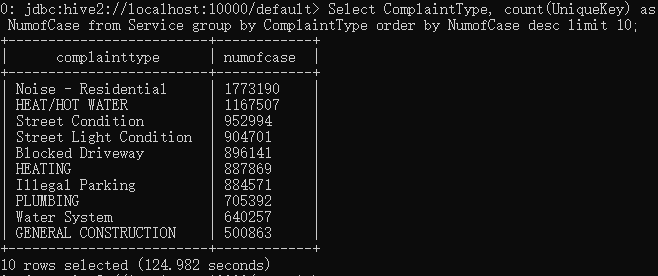
**OUTPUT**



**4-2)NUMBER OF CASE BY COMPLAINT TYPE (TOP 10)**

Select ComplaintType, count(UniqueKey) as NumofCase from Service group by ComplaintType order by NumofCase desc limit 10;

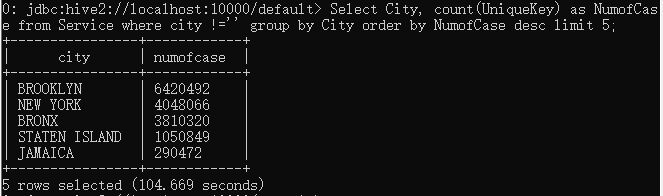
**OUTPUT**



**5)NUMBER OF CASE BY CITY(TOP 5)**

Select City, count(UniqueKey) as NumofCase from Service where city !='' group by City order by NumofCase desc limit 5;

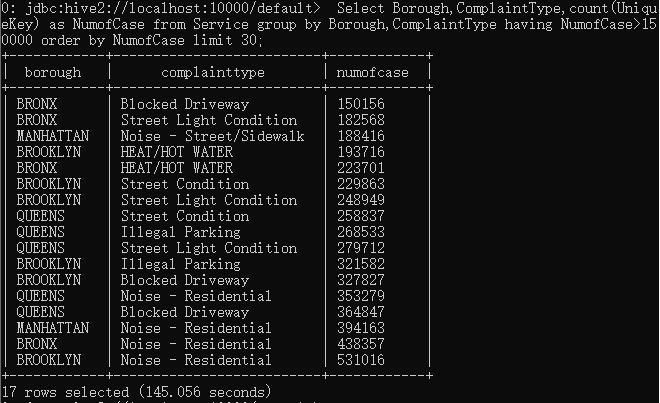
**OUTPUT**



**6)NUMBER OF CASES BY BOROUGH AND COMPLAINT TYPE WITH CASE NUMBER OVER 150,000 (TOP 30)**

Select Borough,ComplaintType,count(UniqueKey) as NumofCase from Service group by Borough,ComplaintType having NumofCase>150000 order by NumofCase limit 30;

**OUTPUT**



**7)EXTRACT ALL THE CASE REPORTED ON JULY 4,2019 WITH LATITUDE AND LONGITUDE VALUES.**

We want to see the case distribution geographically with time series throughout the day on July 4,2019. Some records do not have either latitude or longitude, therefore we use a where condition to exclude those invalid records.

There is a lot of rows on July 4,2019, therefore we use the following query to write the query result to a text file:

INSERT OVERWRITE DIRECTORY '/user/whe8/test' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' Select \* from Service where Substr(CreatedDate,1,10)='07/04/2019' and Latitude !='' and Longitude !='';

After that, we open a new command shell, and logged into the cluster to display the result file by command:

hdfs dfs -ls /user/whe8/test

**OUTPUT**



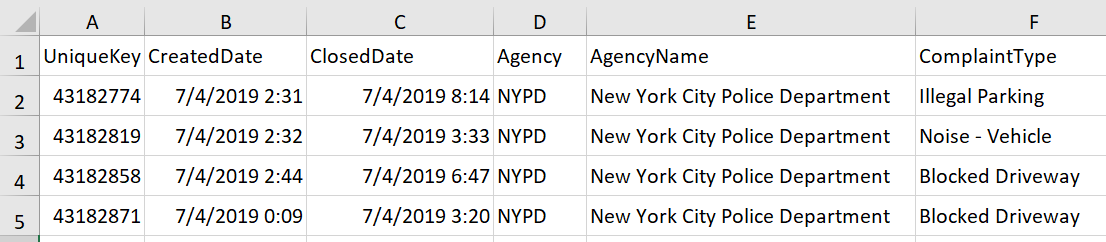
Then we copied the file to local drive of the cluster by the below command:

hdfs dfs -copyToLocal /user/whe8/test/000000\_0 Resultset0704

After that, we use SCP to copy it to the local computer by below command:

Scp whe8@34.221.40.43:/home/whe8/Resultset0704 CIS0704.csv

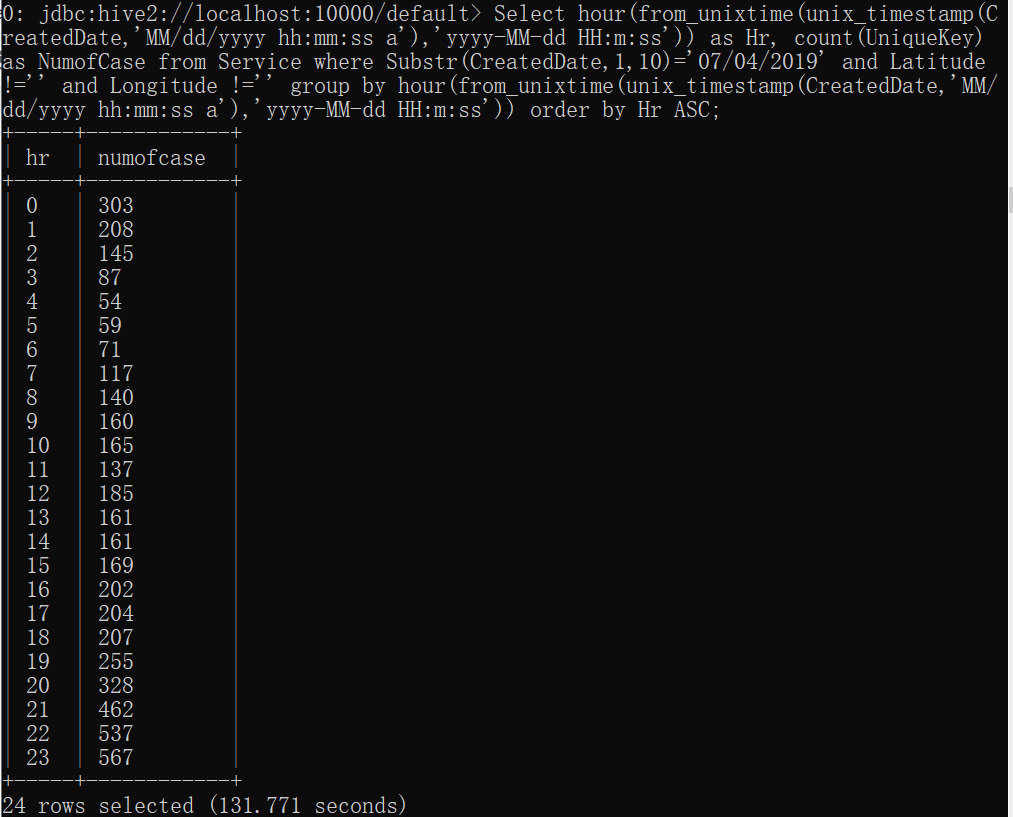
We opened CIS0704.csv in local computer, added back the column headers to this file, and then saved it as an Excel file in order to generate the 3D map with time in Excel. Please refer to the following screen shot for part of this query result in Excel format:



**8)THE NUMBER OF CASE BY HOUR ON JULY 4,2019 WITH BOTH LATITUDE AND LONGITUDE VALUES.**

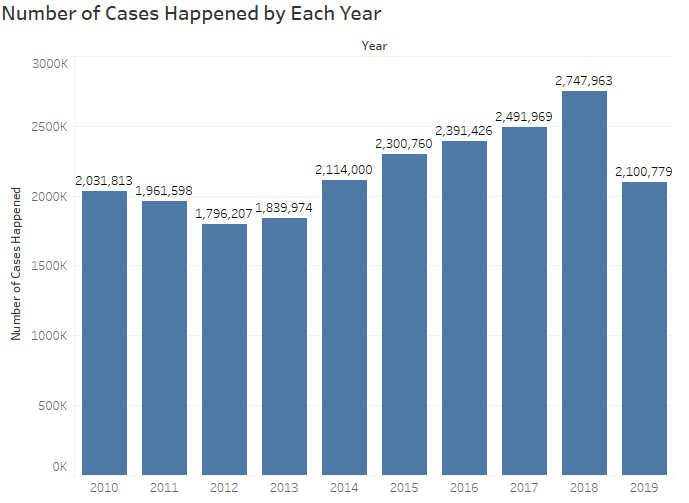
Select hour(from\_unixtime(unix\_timestamp(CreatedDate,'MM/dd/yyyy hh:mm:ss a'),'yyyy-MM-dd HH:m:ss')) as Hr, count(UniqueKey) as NumofCase from Service where Substr(CreatedDate,1,10)='07/04/2019' and Latitude !='' and Longitude !='' group by hour(from\_unixtime(unix\_timestamp(CreatedDate,'MM/dd/yyyy hh:mm:ss a'),'yyyy-MM-dd HH:m:ss')) order by Hr ASC;

**OUTPUT**

****

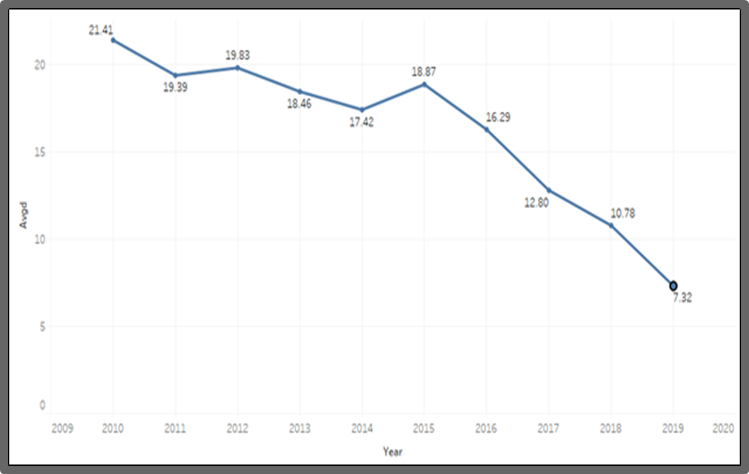
**\***0 represents the time period from 0 to 1,..., while 23 represents the time period from 23 to 24.

**Step 3: Visualization**

1) This step is to analyze the total number of cases over the past 10 years.

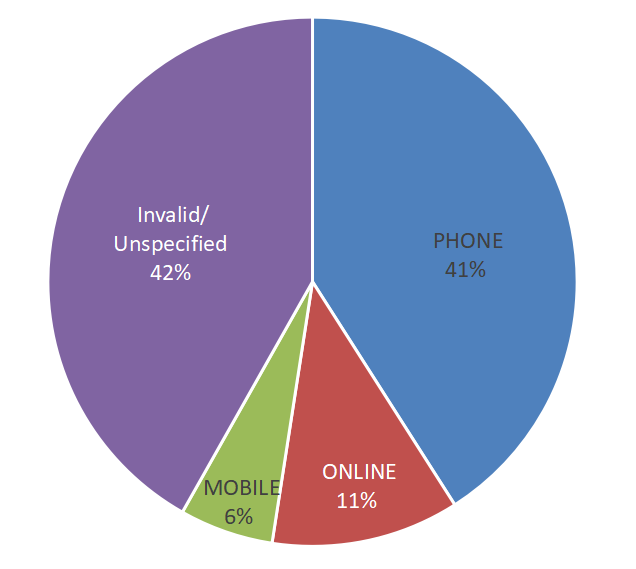
* We need to display the corresponding number of cases happened in each year and trend of cases through 2010 to 2019.
* In Tableau, the row is number of cases happened; the column is individual year.We choose the bar chart, and label the number of cases to show this result.

2) This step is to analyze the average days taken to close one case for each year.



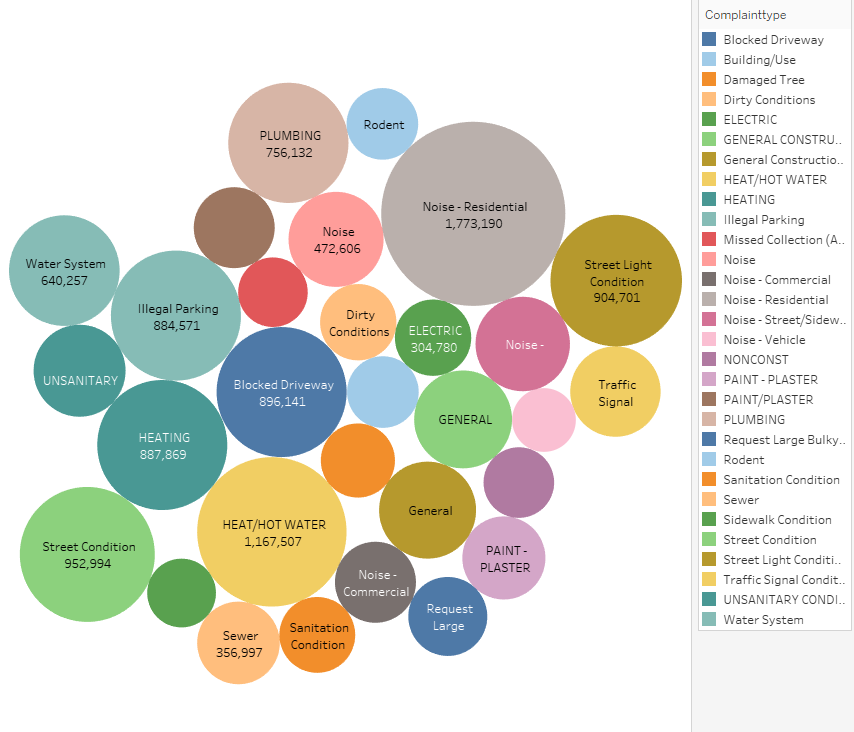
* We need to display the trend of the past 10 years about the average days taken to close one case for each year.
* In Tableau, the row is the average days; the column is year. We choose the line chart to show this result. We put the average days for label.

3) This step is to show the channels that users used to file complaints.



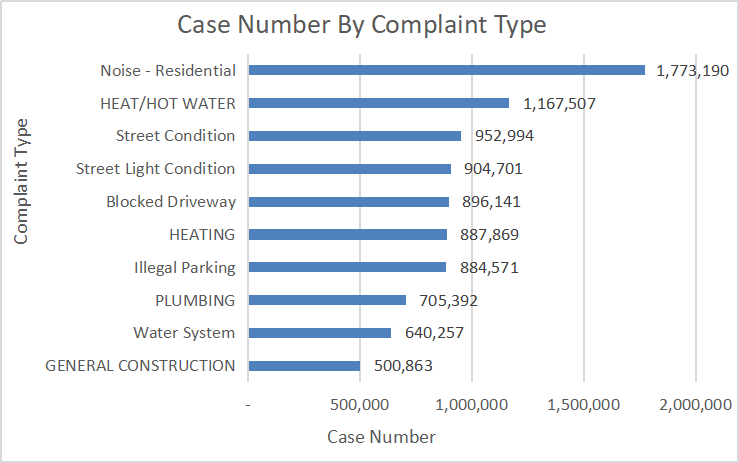
* We need to show the percentage of each channel under the total cases.
* From the query result, only the channels of Phone, Mobile, and Online were identified, others are either empty or unidentified, therefore, we categorized as Invalid/Unspecified in the analysis, we calculated the number for the category of Invalid/Unspecified by: total case number - the sum of numbers for Phone, Mobile, and online.
* In Tableau, the row and the column is open data channel type and number of cases, Then we choose the pie chart to show the result. Select “Entire View” in fit. We drag open data channel type into label; we also drag the number of cases inside label and chose the quick table calculation as “Percentage of total.”

4-1) This step is to analyze the top 30 of the most common complaint type.



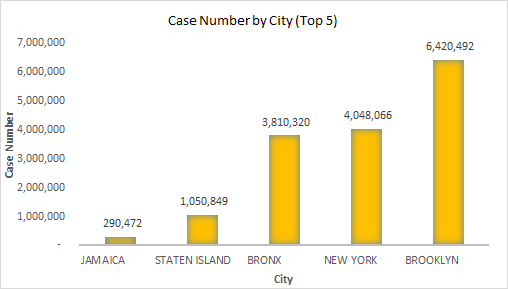
* We first see top 30 complaint types for the 311service in New York City for the past 30 years.
* In Tableau, the row and the column are complaint type and number of cases. We choose the bubble packed to show this result. We drag the complaint type to color, and drag the number of cases to label.

4-2) This step is to analyze the top 10 of the most common complaint type by bar chart.



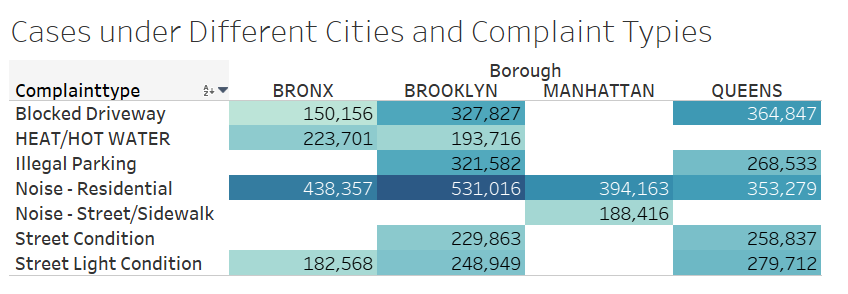
* This is to visualize the top 10 complaint type, and the bar chart shows Noise is the most common complaint type .
* In Excel, we use case number as horizontal axis, while the complaint type as the vertical axis to generate this bar chart.

5) This step is to analyze the top 5 cities with most common complaints.



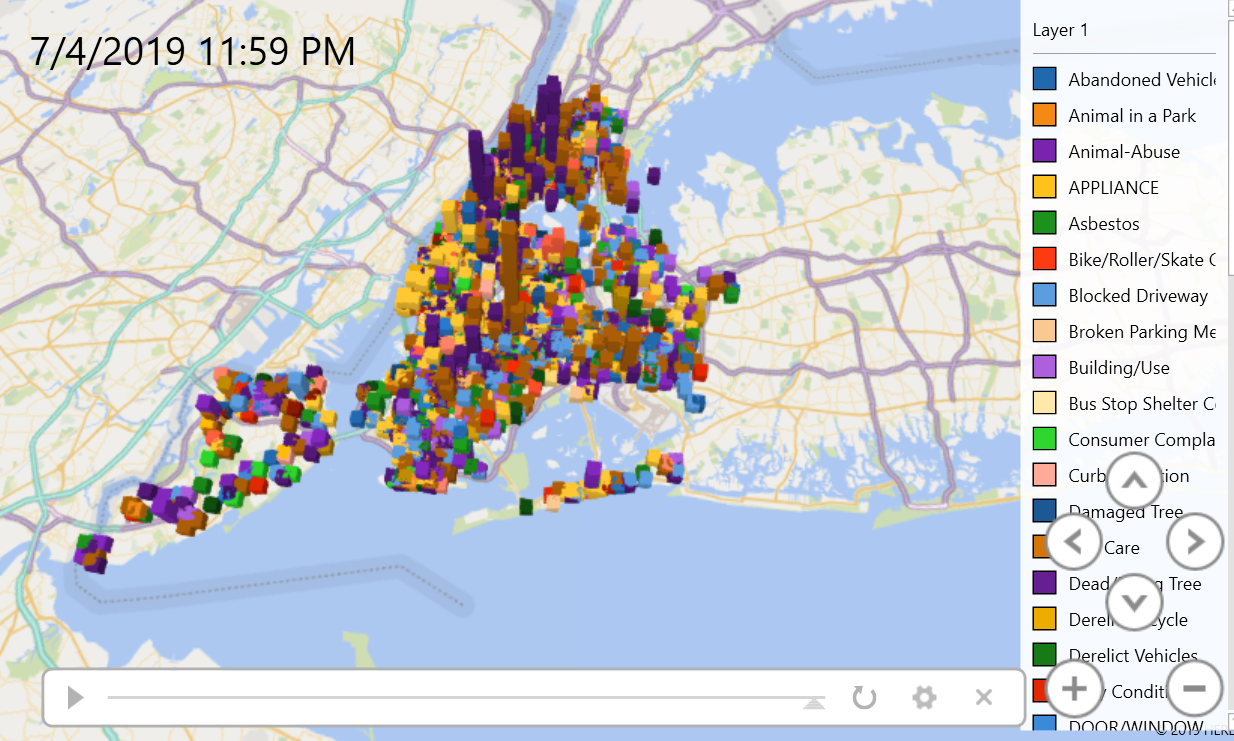
* This is to visualize the top 5 cities with most complaints filed, and Brooklyn, New York, and Bronx take significant parts of the cases.
* In Excel, we use city as horizontal axis, while the case number as the vertical axis to generate this bar chart.

6) This step is to analyze the different complaint type under different boroughs.

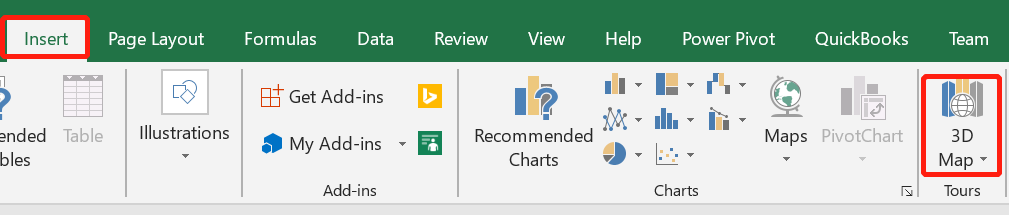


* The condition to get the result is that the total number of cases are greater than 150,000 for major borough and complaint type. We will need to visualize that the significant number of cases are located at which borough and under which complaint type.
* In Tableau, the row is complaint type; the column is borough. We choose the highlight tables to show this result. We drag the number of cases into text and color icon.

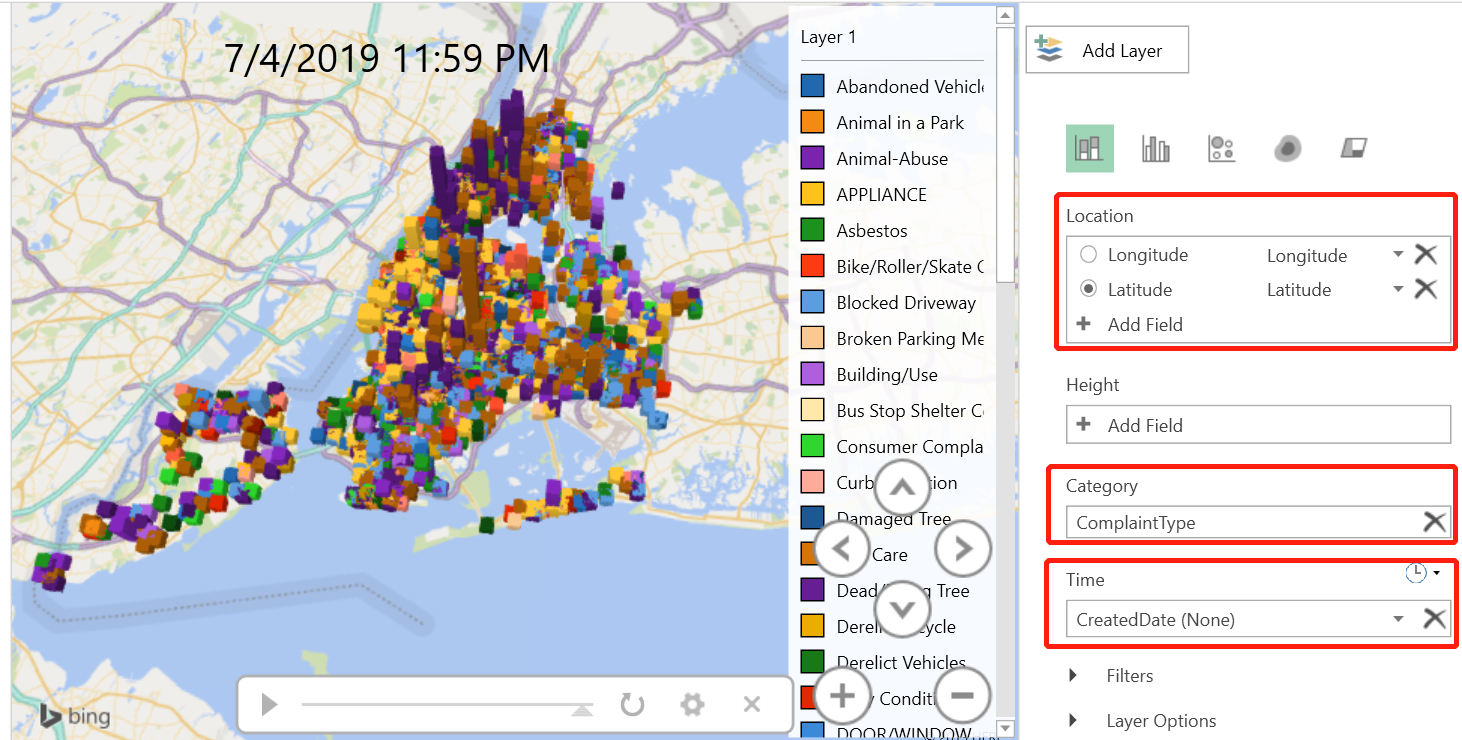
7) This step is to analyze the complaint case distribution geographically on a specified date (July 4,2019).



In order to do this, we need to export the cases filed on July 4,2019 to CSV, then use Excel to open and save it as Excel format. After that we click “3D Map” from “Insert” menu bar as the following:

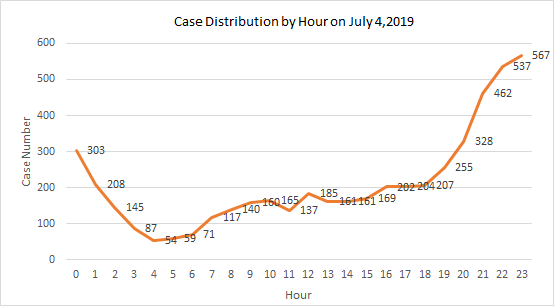


After that, the following screen appears:



Then we need to put the fields Longitude and Latitude to the location section, put ComplaintType to Category section, and put CreatedDate to Time section. After that, the 3D map with time (Geo-Spatial Visualization) appears and is ready to play. We recorded a video to demonstrate, please refer to the following link for this video: <https://youtu.be/6WULXFWI598> .

8) This step is to further analyze the case number by hour on July 4,2019.



\* 0 represents the period from 0-1, while 23 represents the period from 23-24.

* This is to visualize the case distribution by each hour throughout the day on July 4,2019. It shows that more cases were reported in night from 8:00PM to 1:00AM, and less cases were reported in earlier morning from 3:00AM to 7:00AM.
* In Excel, we use hour as horizontal axis and the case number as the vertical axis to generate this line chart.

**Dataset and GitHub Link:**

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

GitHub Link: <https://github.com/aweihe/CIS5200-Group2-Project/tree/master>

**References:**

* 1. <https://council.nyc.gov/data/311-services/>
  2. https://portal.311.nyc.gov/