**[](http://www.calstatela.edu/centers/hipic)** **CIS5560 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@caltstatela.edu](mailto:mcampo61@caltstatela.edu))

Garrett Tice ([gtice@calstatela.edu](mailto:gtice@calstatela.edu))

Joey Tran ([jtran66@calstatela.edu](mailto:jtran66@calstatela.edu))

Brandon Ascencio ([bascenc8@calstatela.edu](mailto:bascenc8@calstatela.edu))

Kyle Alfonso ([kalfons@caltstatela.edu](mailto:kalfons@caltstatela.edu))

12/7/2020

## **NYC 311 Service Request Data Analysis using Hive & Tablea**

## 

## **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.

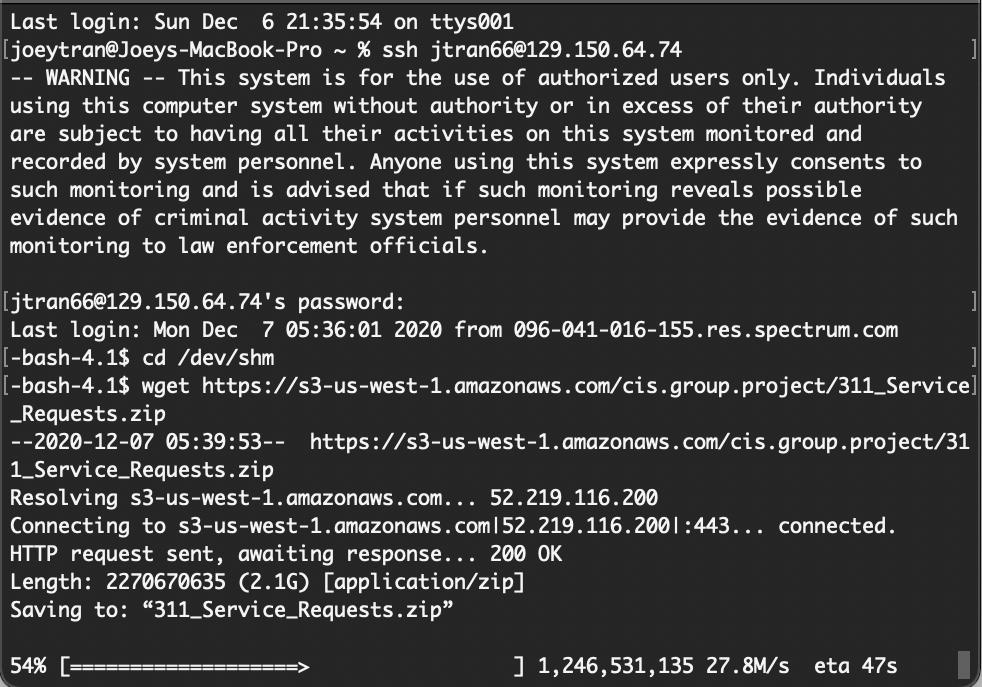
ssh [your\_username]@129.150.64.74

1. 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

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

$ wget <https://s3-us-west-1.amazonaws.com/cis.group.project/311_Service_Requests.zip>



$ unzip 311\_Service\_Requests.zip

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

$ hdfs dfs -mkdir Services

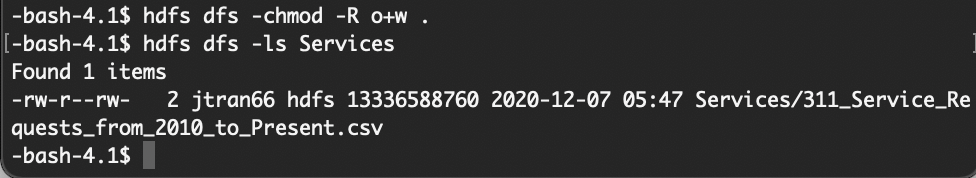
$ hdfs dfs -put 311\_Service\_Requests\_from\_2010\_to\_Present.csv Services

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

$ hdfs dfs -chmod -R o+w .

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

$ hdfs dfs -ls Services



Step 2: Create Hive Tables to Query

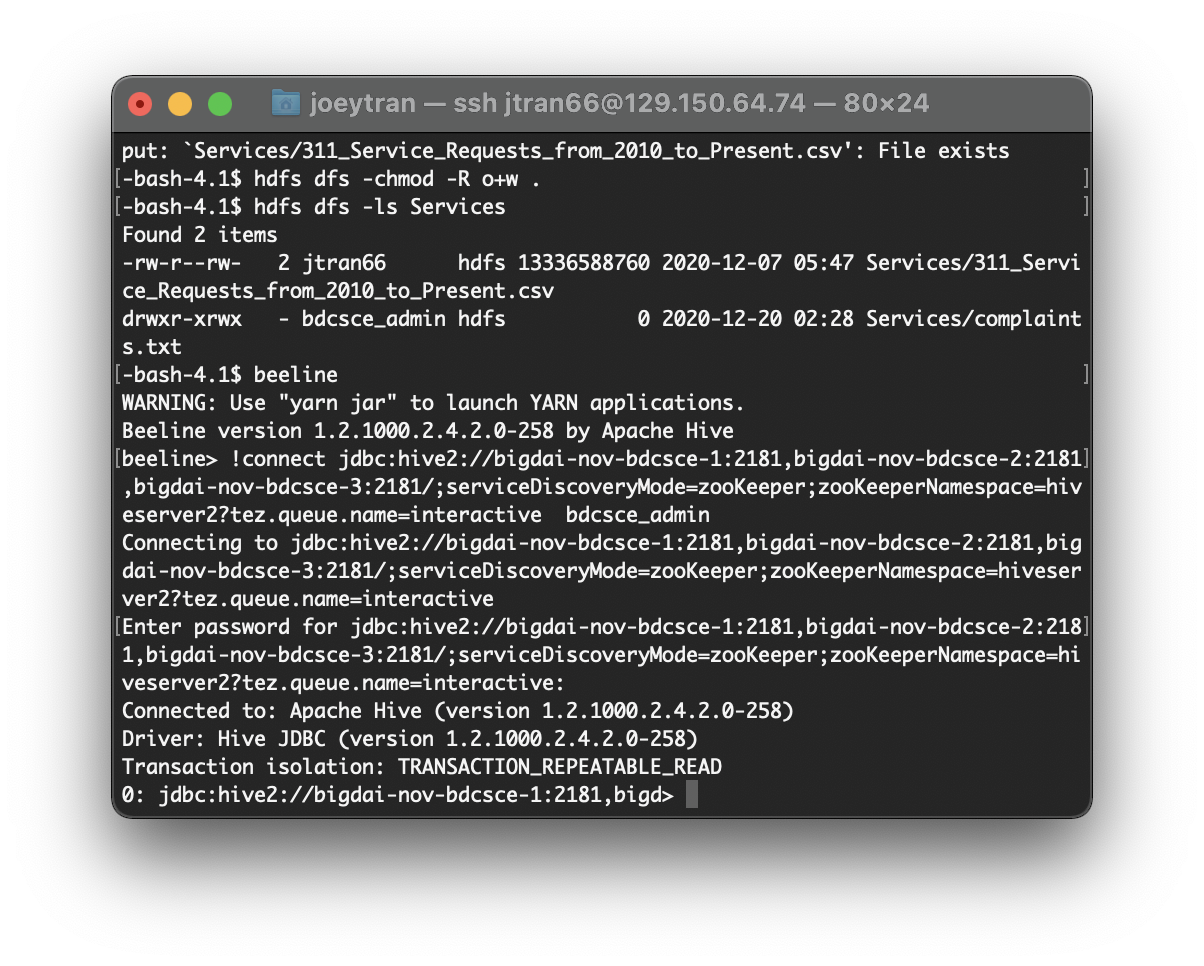
In this step, we will create Hive tables from our uploaded data set.

1. Type beeline to open the Hive interface.

$ beeline

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

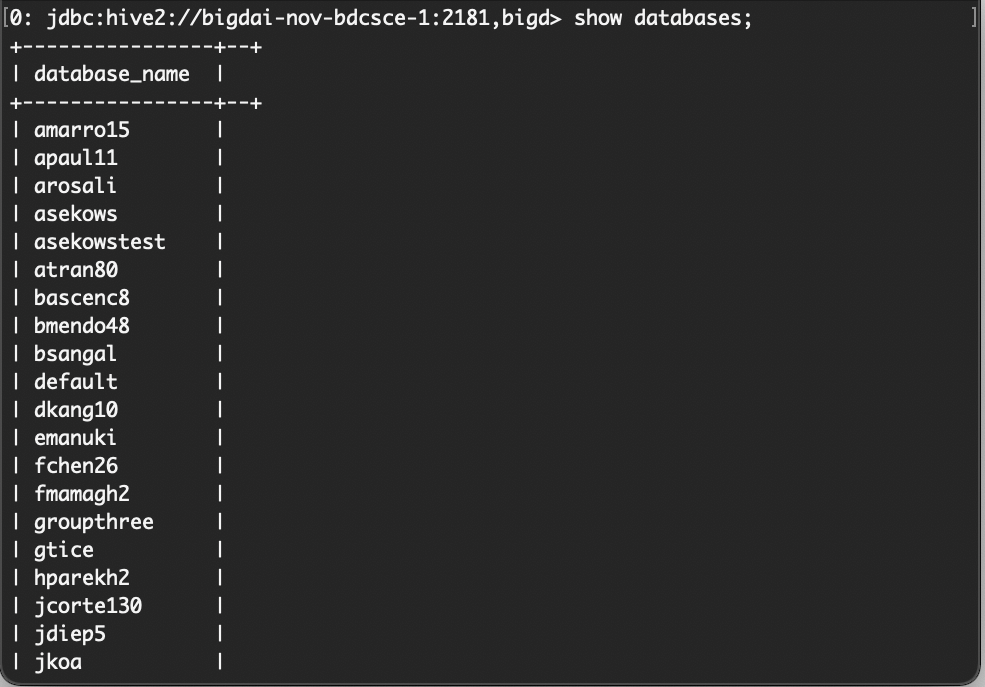
!connect jdbc:hive2://bigdai-nov-bdcsce-1:2181,bigdai-nov-bdcsce-2:2181,bigdai-nov-bdcsce-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce\_admin



1. View and use your database.

beeline> show databases;

beeline> use [your\_username];



1. 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 building table on comma-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,

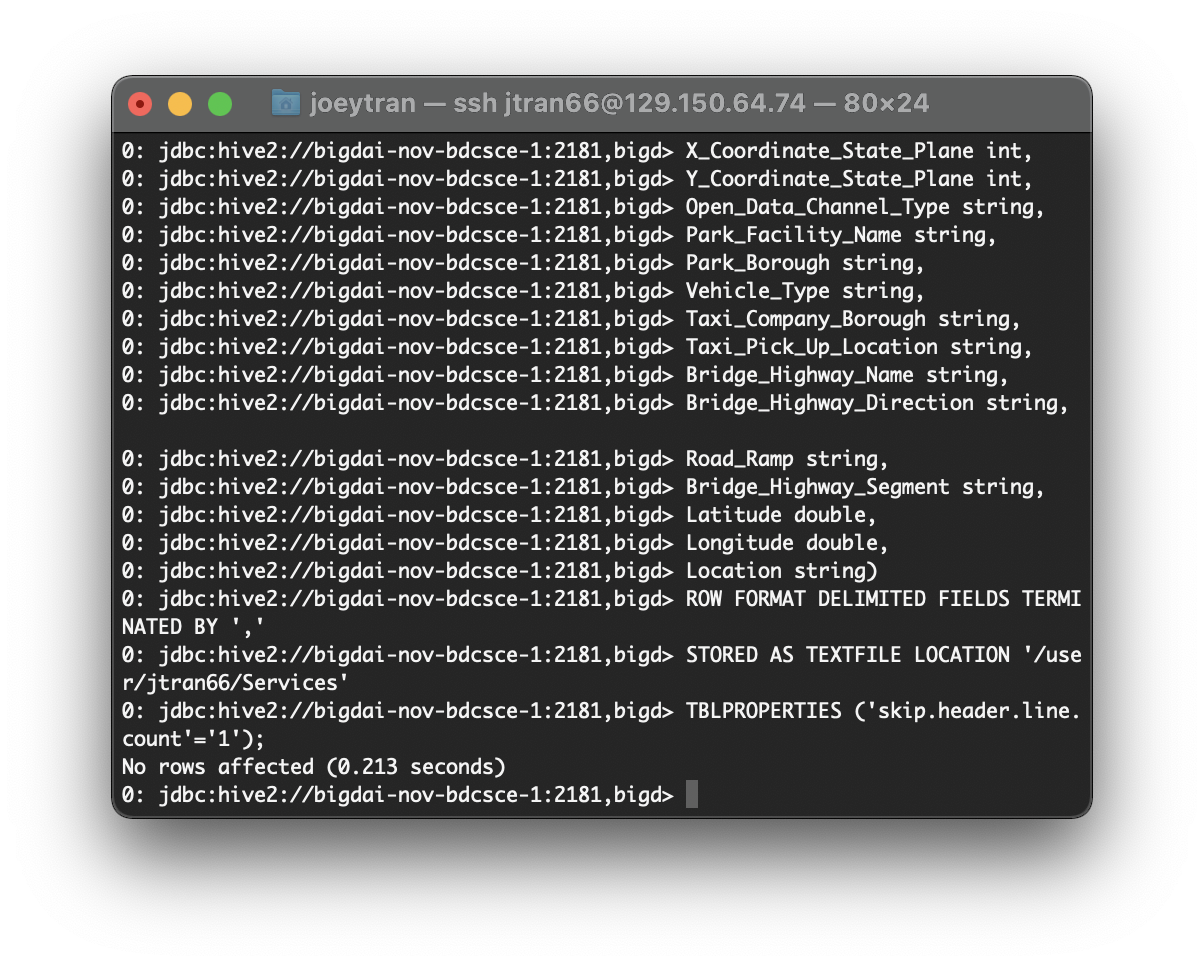
Longitude double,

Location string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/bascenc8/Services'

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



1. 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;

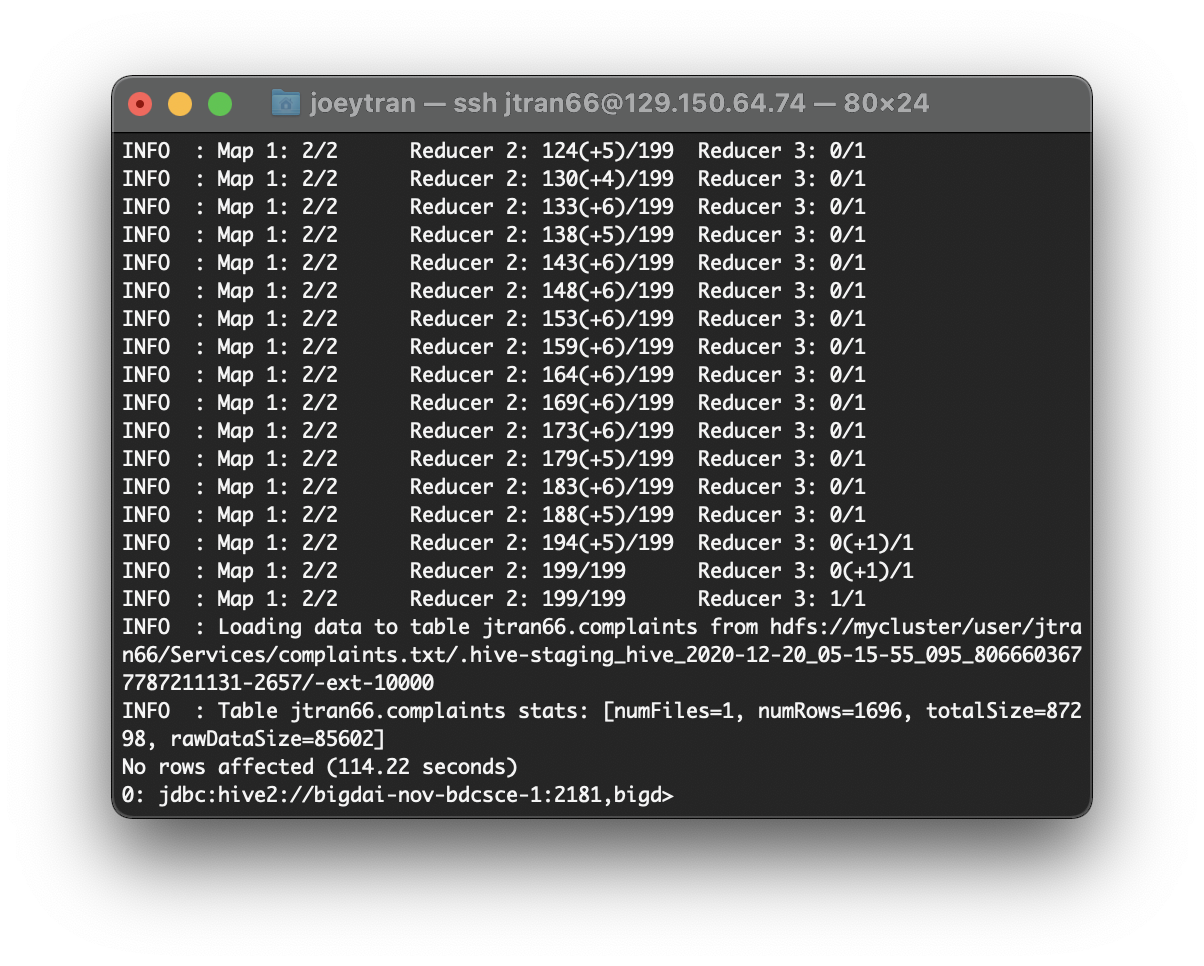
CREATE EXTERNAL TABLE IF NOT EXISTS Complaints(agency\_name string, Complaint\_type string, NumberofComplaints int)

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;



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

beeline> show tables;



Step 3: Hive Query Analysis

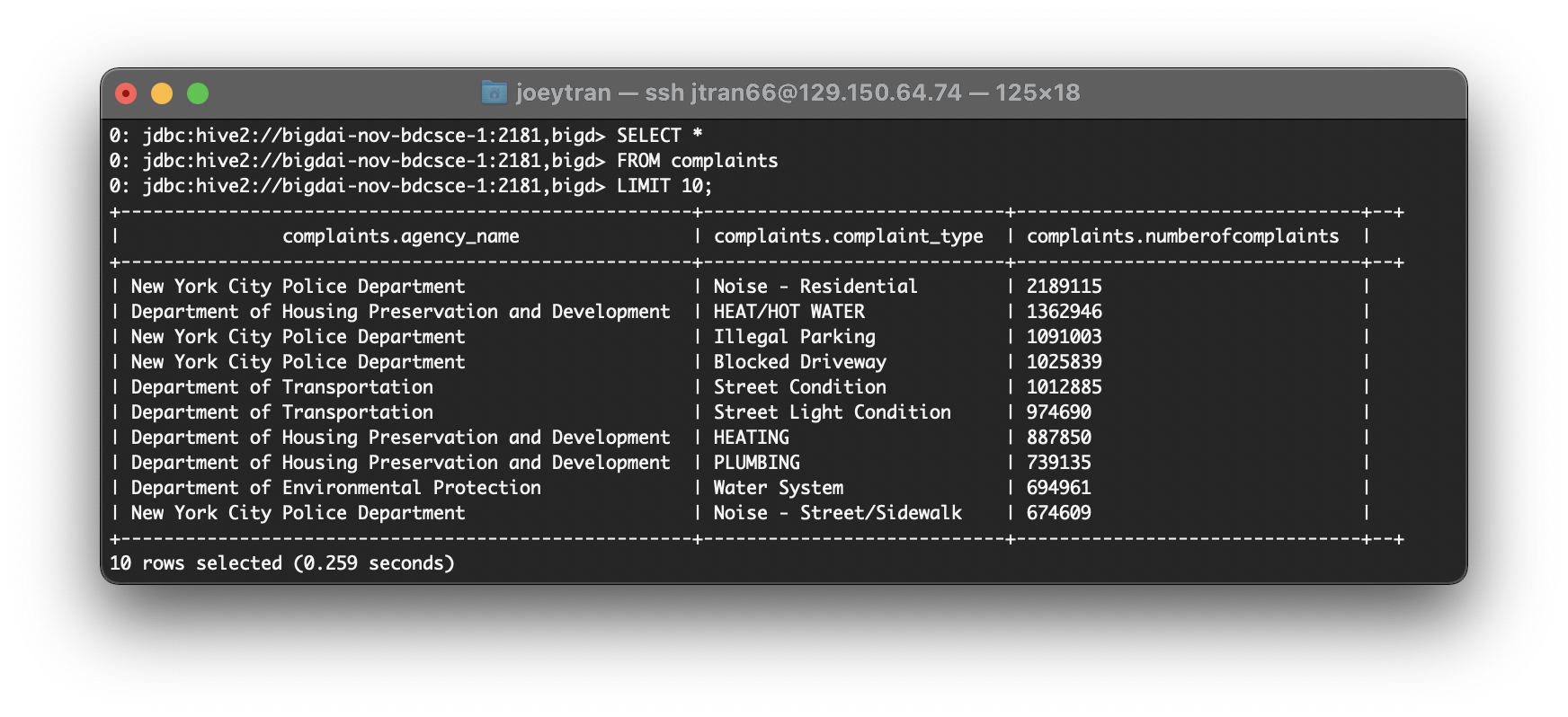
This step will involve running queries from the created tables so that we can extrapolate useful information.

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;



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

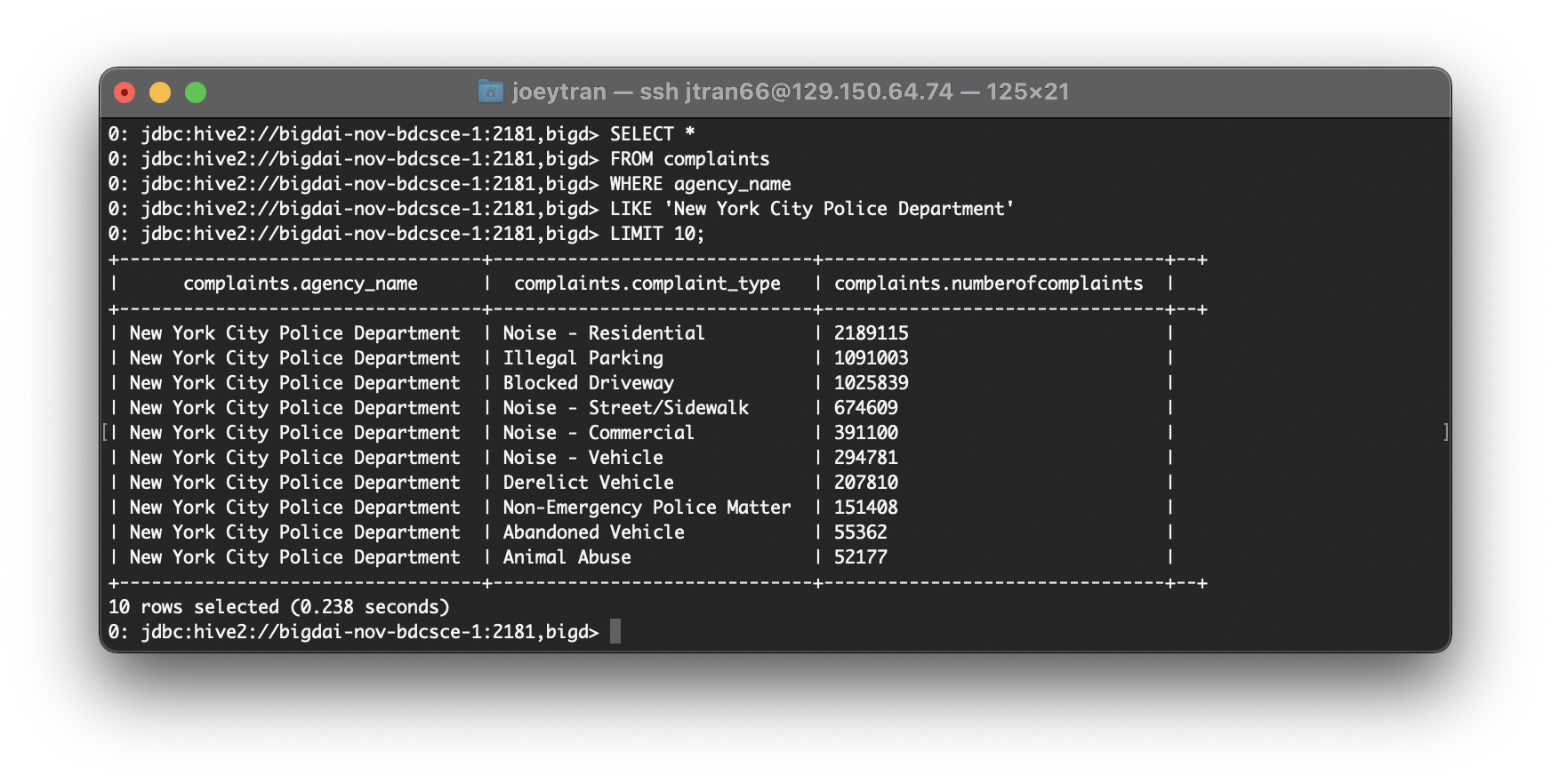
SELECT \*

FROM complaints

WHERE agency\_name

LIKE 'New York City Police Department'

LIMIT 10;



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

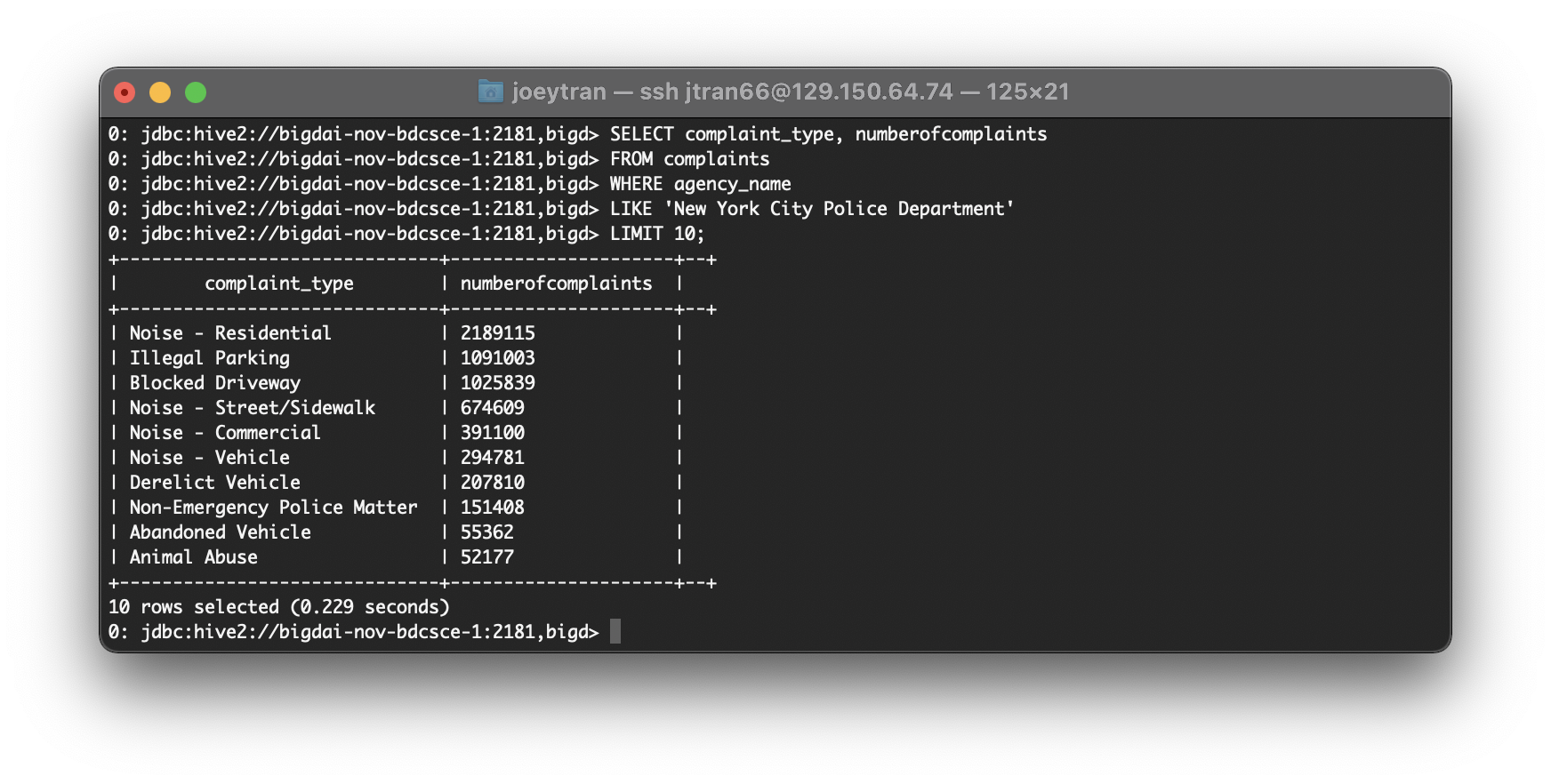
SELECT complaint\_type, numberofcomplaints

FROM complaints

WHERE agency\_name

LIKE 'New York City Police Department'

LIMIT 10;



1. We can run the following query to see which agency handed the most complaints.

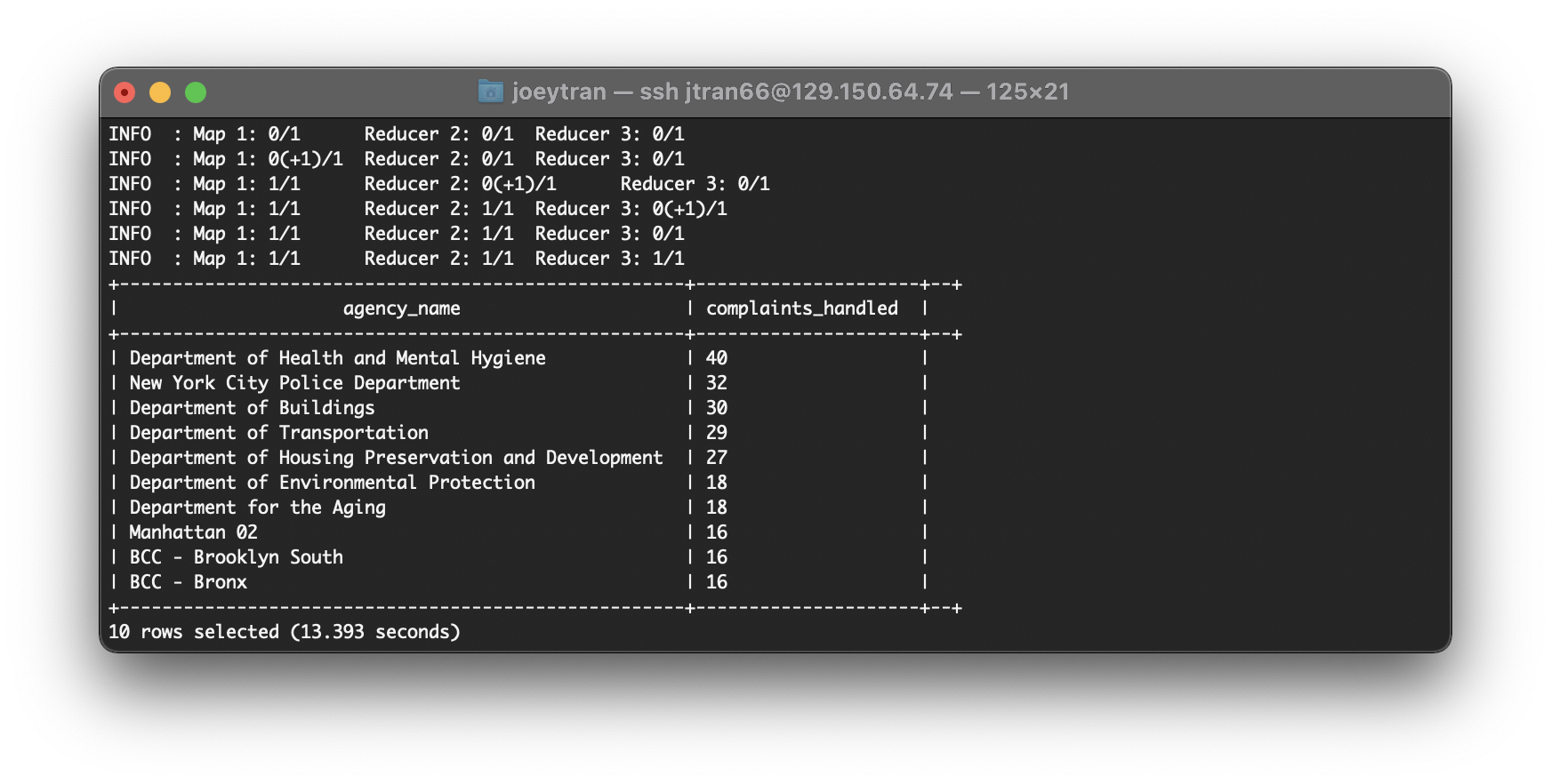
SELECT agency\_name, COUNT(agency\_name) AS complaints\_handled

FROM complaints

GROUP BY agency\_name

ORDER BY complaints\_handled DESC

LIMIT 10;



Step 4: Visualization

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

1. 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.

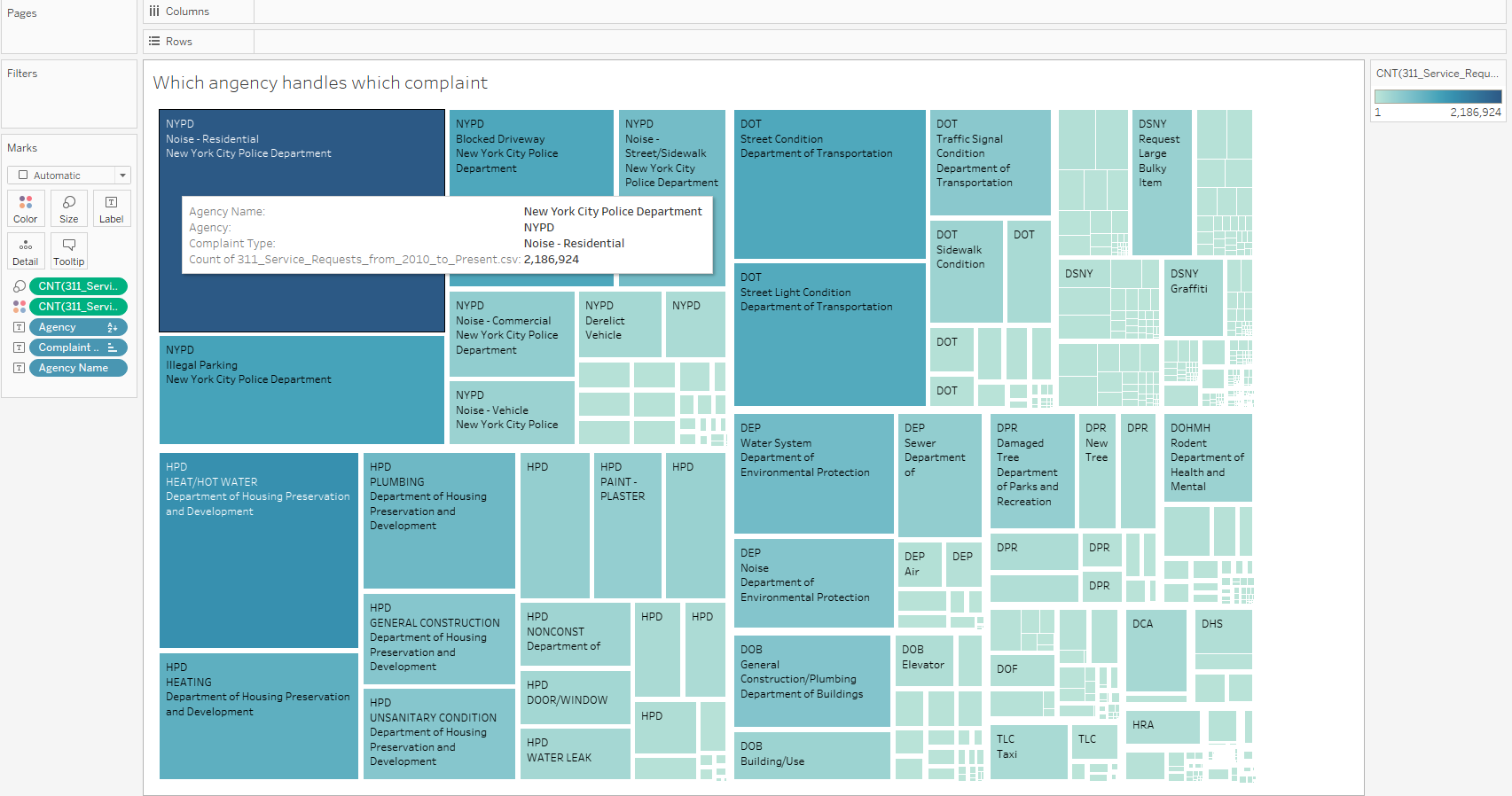
scp [your\_username]@129.150.64.74:/dev/shm/311\_Service\_Requests\_from\_2010\_to\_Present.csv .

1. Open your tableau on your local computer. Select Text File to open your 311\_Service\_Requests\_from\_2010\_to\_Present.csv.

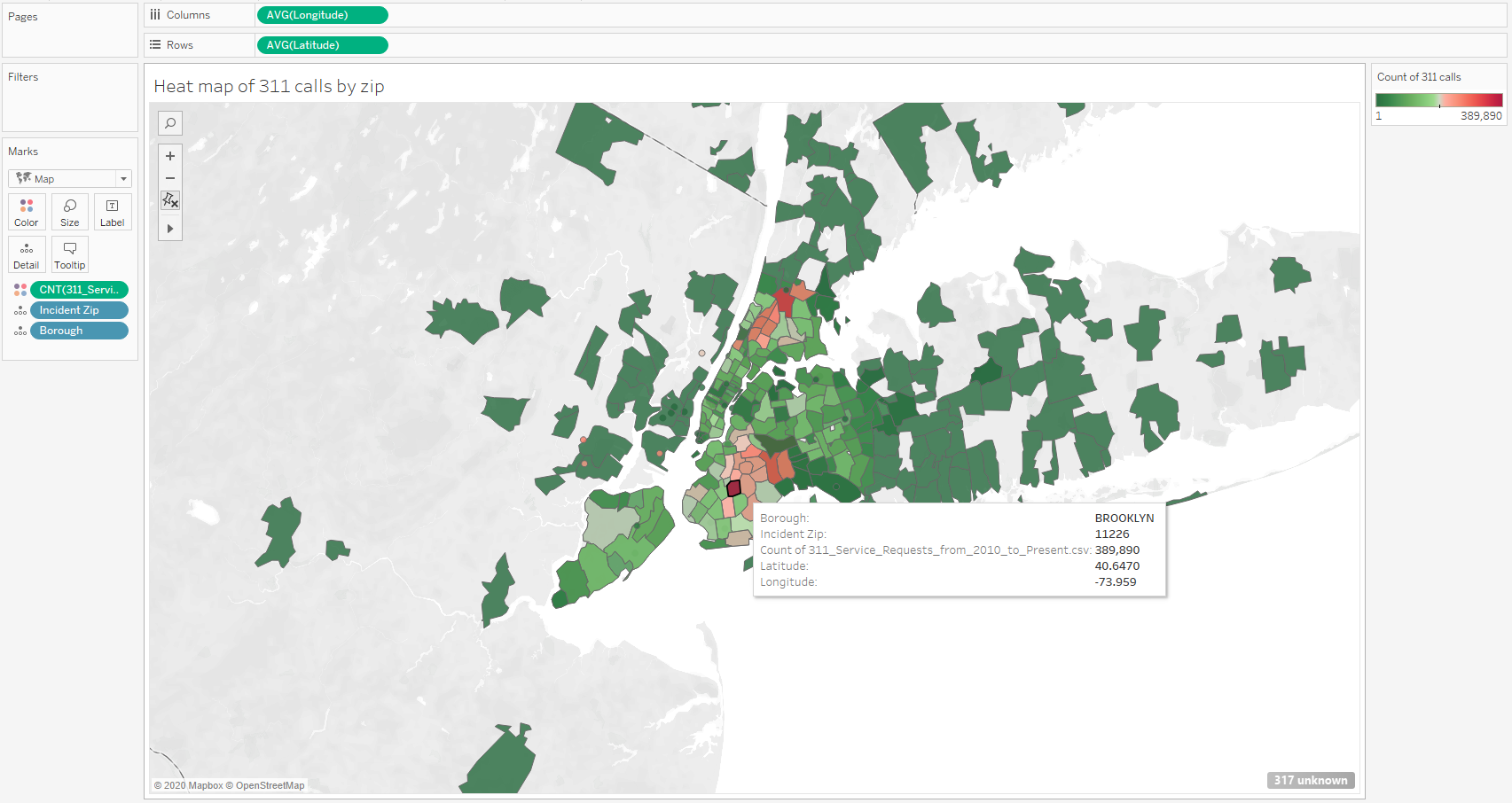


1. Select sheet 1 to start working on our visualization.

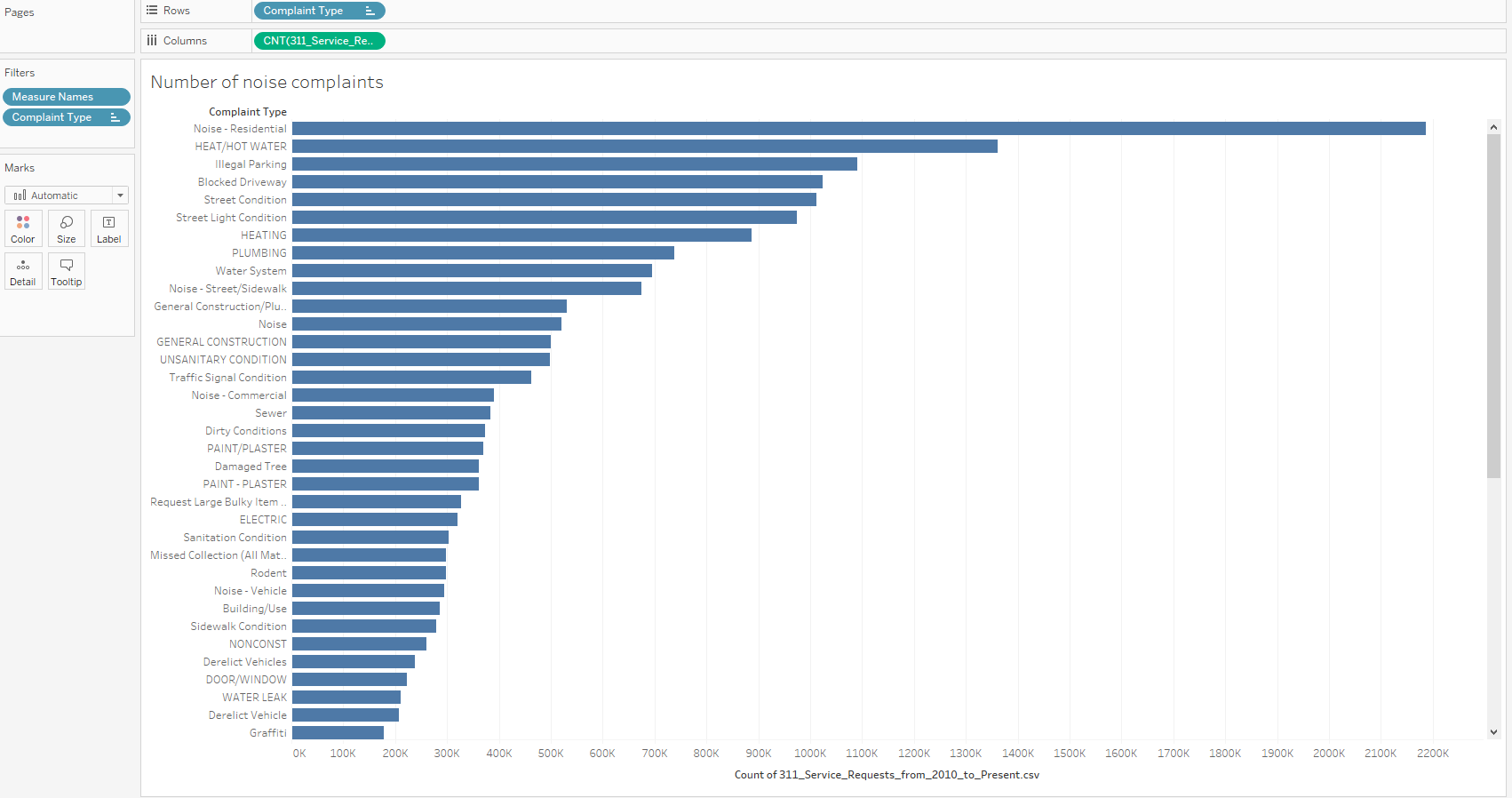
In the tableau viewer drag and drop *agency* to the *columns* section and then *Count* to the *rows* section. Then make the marks set for count size and color by dragging the count to the appropriate mark icon. Drag and drop *agency, complaint type* and *agency name* into the labels icon



1. 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.



1. 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.



References

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