NYC STREET FLOODING ANALYSIS

GROUP MEMBERS

Amit Sinha, Sina Rassaei Kashuk, Vijaya Kumar Venkatramanan, Imen Harrouch, Amruta Pawar

ABSTRACT

Our project involves studying the emergency calls and complaints during any event of flooding of the streets in New York City from 2010 to 2014. We gathered data for all the census data for the emergency calls made during these events and decided to break it down by age group of the callers and also by the income of households in that given region. We intend to present this data on a map and break it down by specific dates, for a given time period (start date and end date), months, annually or for the entire 4 years. We have also computed and presented some graphs to evaluate our data in a more efficient manner. We will study the data from the map and the graphs to analyze and estimate which age groups and which income groups tend to face more inconveniences during flooding events and also which regions usually tend to have more street flooding compared to other regions. This should help us establish a trend (if there is a trend) to help us better understand the people and the region which tend to suffer the most during flooding events. We can then use this to help the city officials prepare better for any such events in the future by having a better understanding of the regions which will be hit the hardest or by knowing the regions and the households which tend to be most ill prepared for such events. This information can and perhaps will also be used the better educate these people on how to be more aware of such upcoming disasters due to bad weather or excessive rainstorms and how to stay well prepared for such events should they end up facing such hazardous conditions in the future. This will not only help in minimizing the emergency calls which will save the city lot of money but also ensure that by being better prepared, the lives of individuals and the people on the emergency response teams are safe by not having to go out during hazardous and/or severely inclement conditions. Also this can help the city be better prepared and plan on how to improve and upgrade the drainage designs in such areas so that the flooding is minimized even during hazardous events such as hurricanes. This project can be of immense use to NYC-OEM who plan for emergencies and coordiantes emergency response, NYC-DEP which is concerned with public health and with the city's

drinking water supply, NYC-DOT in their efforts to provide safe and reliable means of transportation and NYS-RISE to help reduce the risks involved with storms.

INITIAL IDEA

Initially we were planning to use all the data for the flooding complaints and also to include the information on all the streets that were more affected by floodings compared to other streets. This would have added information on how the streets of NYC featured during flooding and how and why some streets fared better than others and/or why some streets tend to suffer more than streets in other areas. Given our time constraints, we decided to no pursue the data analysis of street flooding and limit of project to the evaluation of regions as a whole and evaluation of people based on age group and household income.

OUR PRIMARY TARGET (USEFULNESS)

The data analysis from this project can be extremely helpful to many organizations in NYC who are responsible for public health and safety. Our project can offer some insight on how to be better prepared and how to better predict which areas would need more help and attention during flooding events. Here are some organizations in our city who can benefit from such project work.

- NYC – Office of Energy Management:

The agency plans and prepares for emergencies and also educate the public about how to be better prepared during an emergency event. They also coordinate emergency response and recovery, and collect and study emergency information. Our project can help them better analyze and focus on regions which need more attention than others in terms of educating the people how to be better prepared for flooding and for planning better emergency response for such regions during flooding due to rainstorms or hurricanes.

- NYC – Department of Environmental Protection:

They are responsible for the public health and the environment by supplying clean drinking water and also for collecting and treating wastewater, reducing air, noise, and hazardous material pollution. They are also in charge of NYC storm water network. Our data analysis can help them come up with better plans on managing and preventing water pooling and flooding of streets and also on ensuring that the regions which suffer most have clean water supply during such events.

- NYC – Department of Transportation:

DOT's mission is to provide for the safe and efficient movement of people and goods in the City of New York and to maintain and enhance the transportation infrastructure crucial to the economic vitality and quality. They can benefit by maintain and preparing better roads which do not suffer from pooling of water.

- NYS – RISE:

One of their responsibilities include to study storm risks in lifelines and communities along the New York State coasts, including shelter, power, portable water, sanitation, communication, transportation, medical care, and emergency response. Project such as ours can help them with better plannings and designs for reconstruction projects and to help improve the emergency response teams by being better prepared and coordinated.

This way not only can the city better aware of people and regions which will need more attention than others, the city can better prepare these regions so that in future occurrence of rainstorms and hurricanes, the impact on these areas is less and the safety of the general public and the people involved with emergency response and support is vastly increased.

GOALS (INTERESTINGNESS)

- Gathering 311 data for the emergency calls from 2010 through 2014. Also gathering the census-tract data for NYC which would include the household incomes and age groups of the people in those regions.
- Study and clean the data to ensure that it can provide us with useful information on past flood events and resiliency. This will help us ensure that we are accurate in establishing the regions which tend to be at high risk be it inlands or coastal.
- Investigate who are making the calls and then analyze them based on their socio-economic status (their incomes) and their age groups.
- If possible during the course of this project (as there is a significant time constraint), try and suggest ways in which the emergency complaint reporting

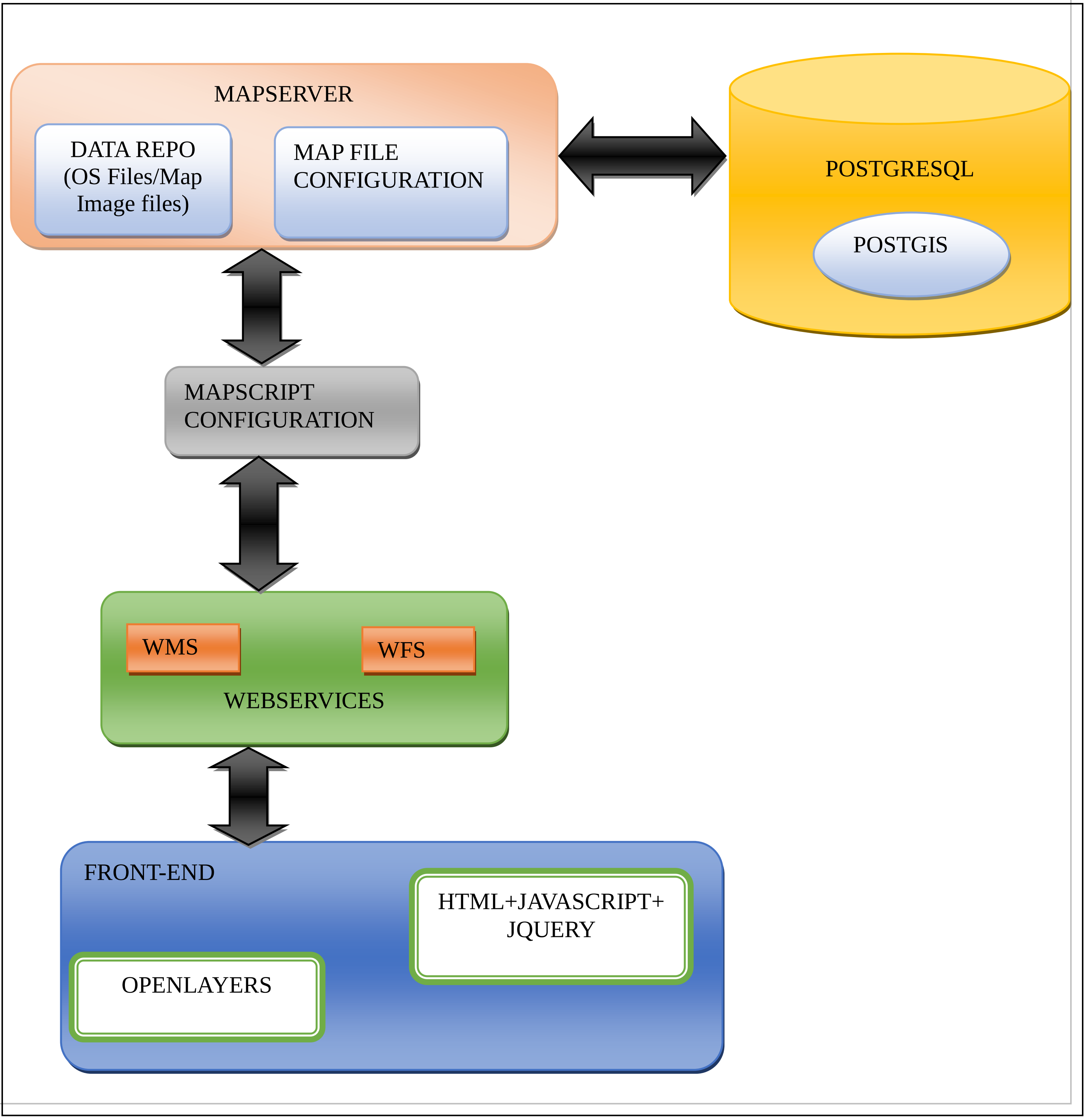
may be improved to get better information.

- Provide insights on how flooding impacts risk, when and/or where they occur most and what could be done to mitigate them.

DATA ANALYSIS STRATEGY (INTERESTINGNESS)

For analyzing the data, our group decided to make use of highcharts along with representation of data on map using open layers. We have broken down our data based on the complaints made vs the day and time on which the complaint was made. This way we can analyze data per year, per month, per week, or per day. We can also break it down on an hourly basis. By using this way to query for our data, we can study the complain density for each borough across the city. Studying the complaint density will help us understand which regions are under the highest risk and need most attention. It will also help with trying to study and understand why these regions tend to suffer the most and comparisons can be made between these regions with the regions where the complaints tend to be much less. All this information can be very helpful in understanding every region, the needs of these regions, and what can be done to improve them so that during flooding they are not so adversely affected.

SYSTEM ARCHITECTURE



IMPLEMENTATION DETAILS

311 Complaints on the map:

This feature is used to represent all 311 complaints received during 2010-2014 on a Map. The geometric representation of a tuple of 311 is a point. To represent the point a vector polygon style with no size was used, which is represented by a small v-shape. There are two ways of representing a point, 1) Eclipse(Circle) and 2) Vector. For the representation of points a combination of Vector and eclipse were used.

Implementation of Complaints by date:

For a better analysis of 311 complaints a feature to visualize the complaints based on the date of occurrence is introduced. A user can view the complaints occurred in a date range by selecting the start date and end date. The date feature was part of a Jquery package called date picker, which was used to show a calendar for user to pick a date. Once the user enter the start date and end date, the respective attributes for start and end date are set in the WMS call for the map layer which fetches the complaints by date. Through the WMS call we pass the start date and end date to the map script file which extracts the date range from the request and query postgresql database. The result of the query from the data base contains all the geometrical data of the complaints. The Map script also fetch a new map layer from the Map server. Along with the geometry of the complaints, a new map layer is displayed on the screen. This map layer can be viewed by selecting the option in Data Analysis Section. Below Openlayer WMS call was made to fetch the complaints in a date range:

```
var startdate= new OpenLayers.Layer.WMS( "Complaints:"+strt_date+" TO
"+end_date, url+start_date+strt_date+aprnd+e_date+end_date, {layers:
'nycd,ny12',format: "image/png"} );
```

Implementation of NYC Complaints by census tract:

The NYC complaints by census tract is used to show information about each census tract. Each census tract has related information such as the Area , The Borough , total population of that census tract, Median Household Income, Different Age group. All of these data was available in a data set called census tract, but only information which is missing was Number of complaints which was available in another data set called Data311.

The challenge was to find out how many complaints were present in each census tract. A spatial relationship was used to identify the number of complaints in each census tract. The spatial relationship is known as ST_Contains. Below was the query used to join identify number of complaints and it was added to the census tract data set:

QUERY:

```
UPDATE censustractshpdata
SET numofcomplaints=B.total
FROM
(
 SELECT count(*) AS total, censustractshpdata.objectid
 FROM Data311,censustractshpdata
 WHERE ST_contains(censustractshpdata.geom,Data311.geom)
 GROUP BY censustractshpdata.objectid
) B
WHERE censustractshpdata.objectid = B.objectid;
```

To fetch the information related to census tract a new WMS layer was created and WFS was created. GetFeatureInfo provides feature information by identifying a point on a map based on its pixel location. Following were attributes were sent as a part of request when a user click at a particular census tract:

```
REQUEST: "GetFeatureInfo",
EXCEPTIONS: "application/vnd.ogc.se_xml",
BBOX: l_wms.map.getExtent().toBBOX(),
X: e.xy.x,
Y: e.xy.y,
INFO_FORMAT: 'text/html',
QUERY_LAYERS: l_wms.params.LAYERS,
WIDTH: l_wms.map.size.w,
HEIGHT: l_wms.map.size.h
```

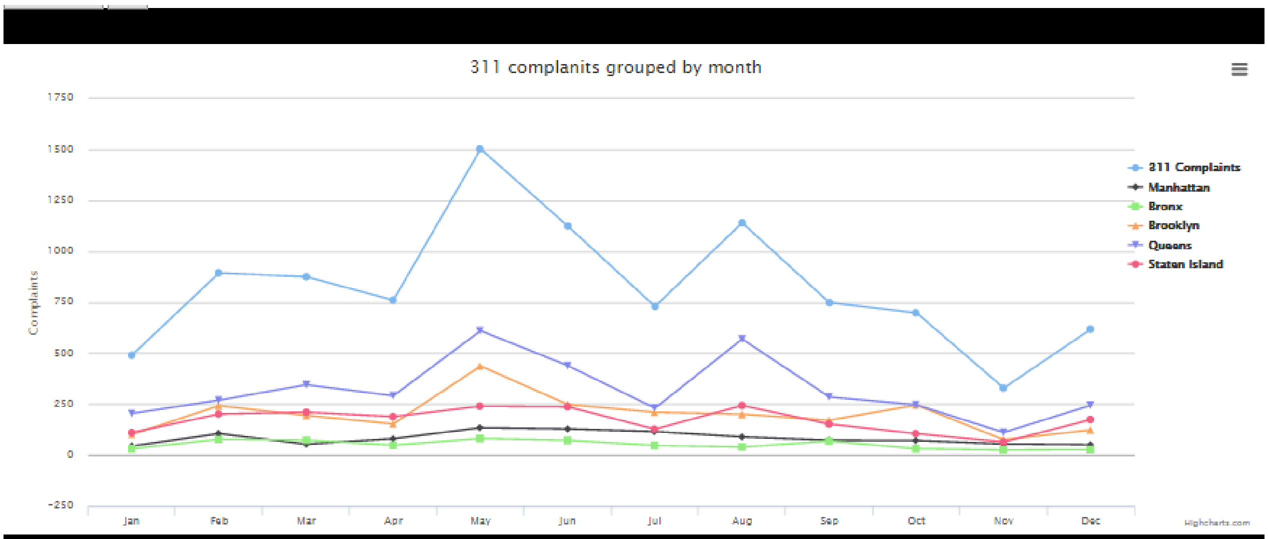
The response received by this call consist of the information related to the area selected in the census tract. The response is set in the table present in the html file.

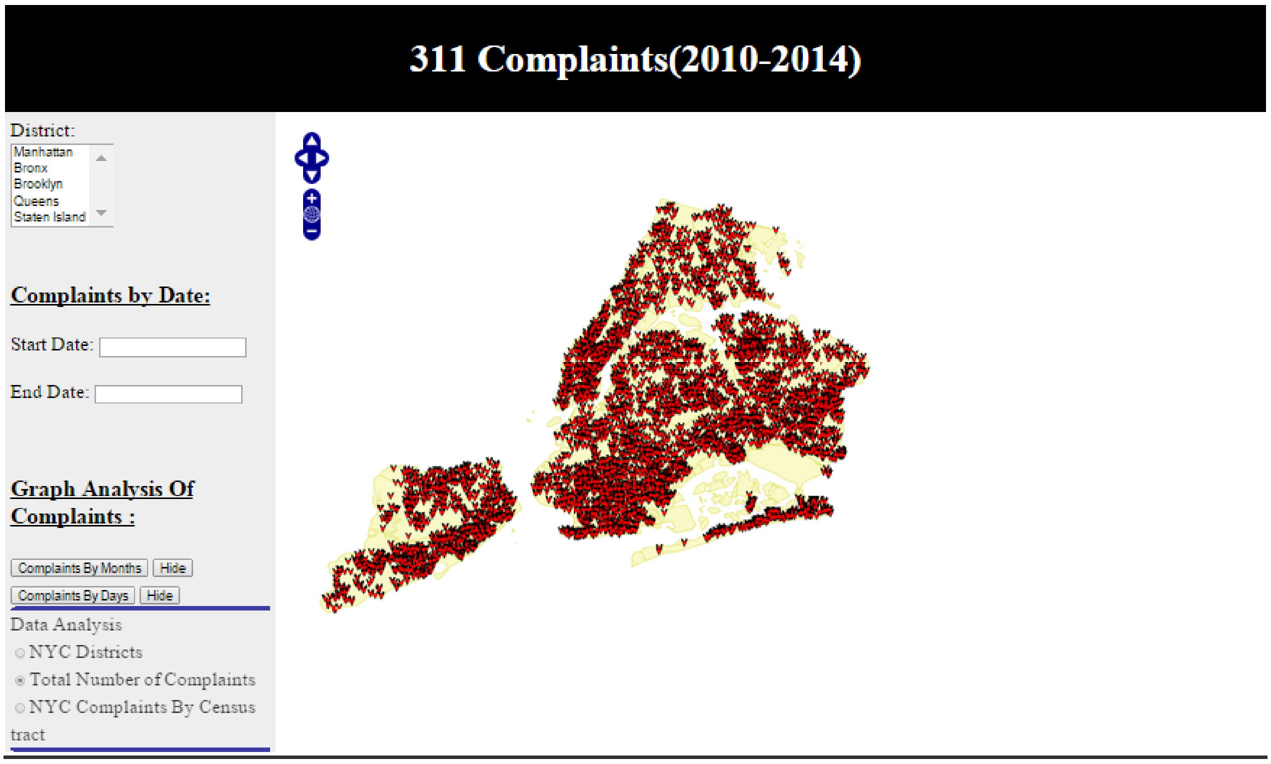
Highcharts is a javascript library which is used to create interactive charts on our webpage. The main intention to use this in our project was to display our 311 Data graphically. We have 311 Data on Y axis and year(2010-2014) on X axis. On hovering the tooltips it displays the number of complaints and year for the selected borough. Analysing the data using these graphs are easier as it displays all the required data on the graph and the effect of flood can be analyzed for each borough, for a given date range or year from 2010-2014.

We have also used Highstock which is based on Highcharts and has all the core functionality of Highcharts. Highstock has a range selector where we can select a range to display our 311 Data in terms of Date(From: To:), for a duration of 3months, 6months, 1 year, ALL by clicking on one of the option and also using a navigator and a scrollbar at the bottom we can choose display our 311 Data for selected range .

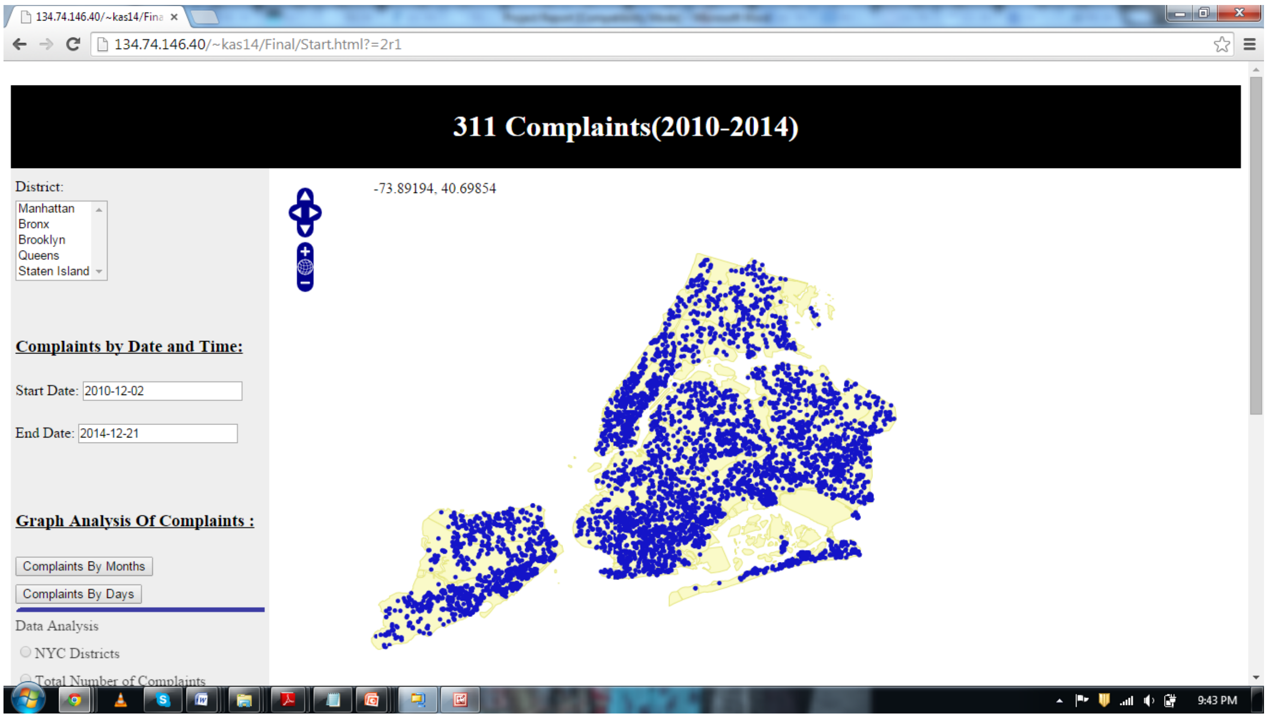
The creation of highcharts using javascript requires a <div> in our page, a chart is initialized by adding the JavaScript tag <script> </script> in a webpage, containing the following code for jQuery as shown below in the screeshots. The div is referenced in the jQuery object. A separate constructor method called Highcharts.StockChart is used to insert the stock chart. In these charts the data is supplied in a separate javascript array, taken from the server i.e, the data is retrieved in the JSON format from the server(data2.php in our project) to Chart.html ,Chart2.html file where the Highcharts and Highstock objects are created to display the graph in our webpage.

Screenshots:

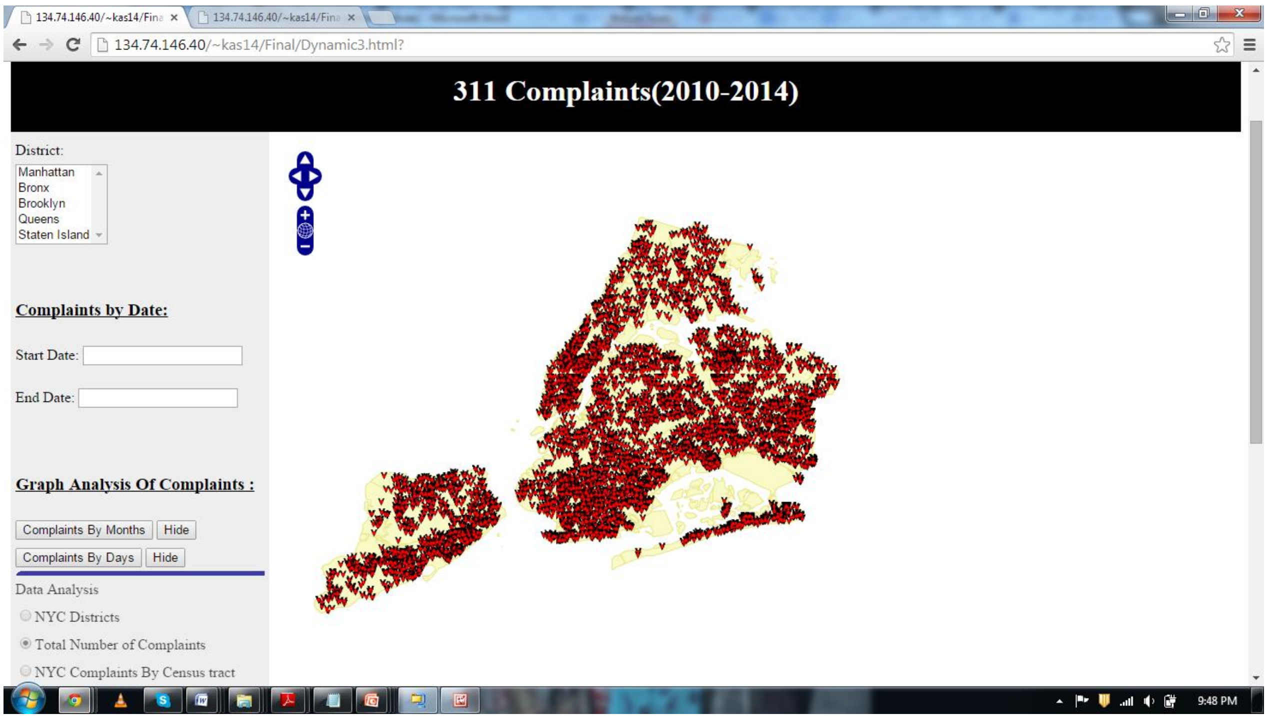




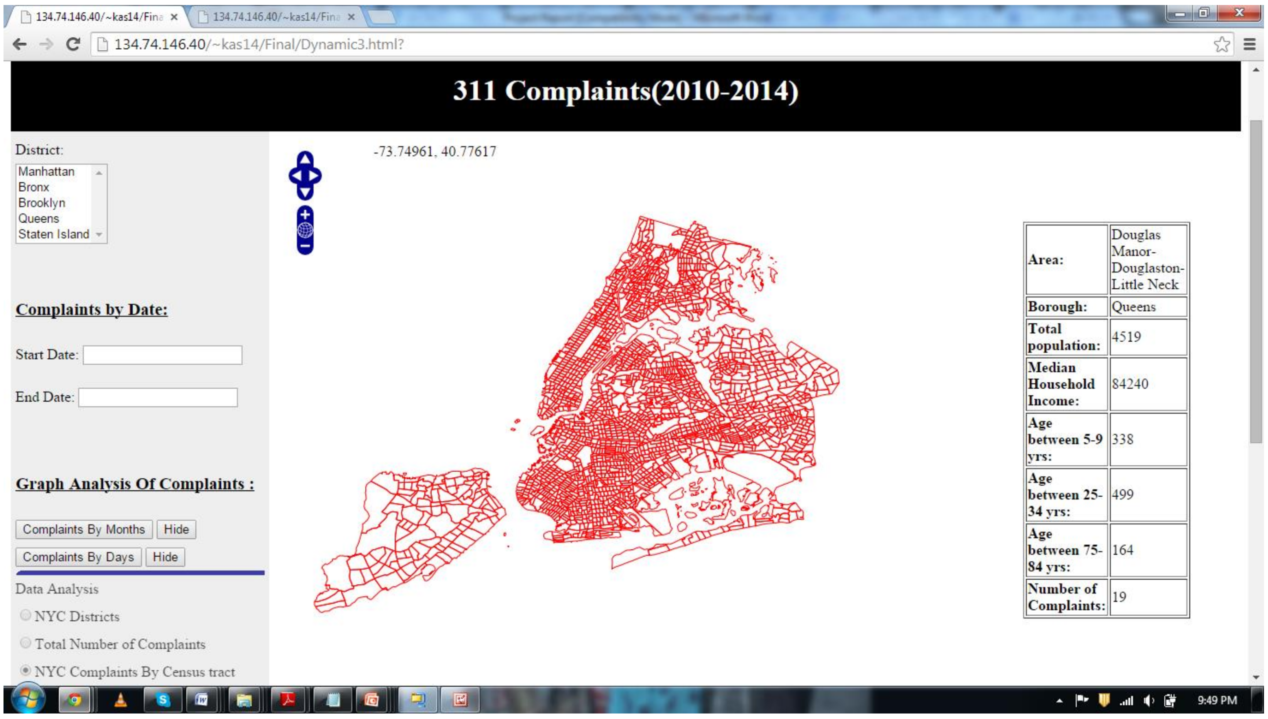
Number of complaints from Start Date: 2010-12-02 End date:2014-12-21:



Total Number of complaints in NYC from 2010-2014:



NYC Complaints by Censustract:


311 Complaints(2010-2014)

District:

- Manhattan
- Bronx
- Brooklyn
- Queens
- Staten Island

Complaints by Date:

Start Date:

End Date:

Graph Analysis Of Complaints :

Complaints By Months | Hide

Complaints By Days | Hide

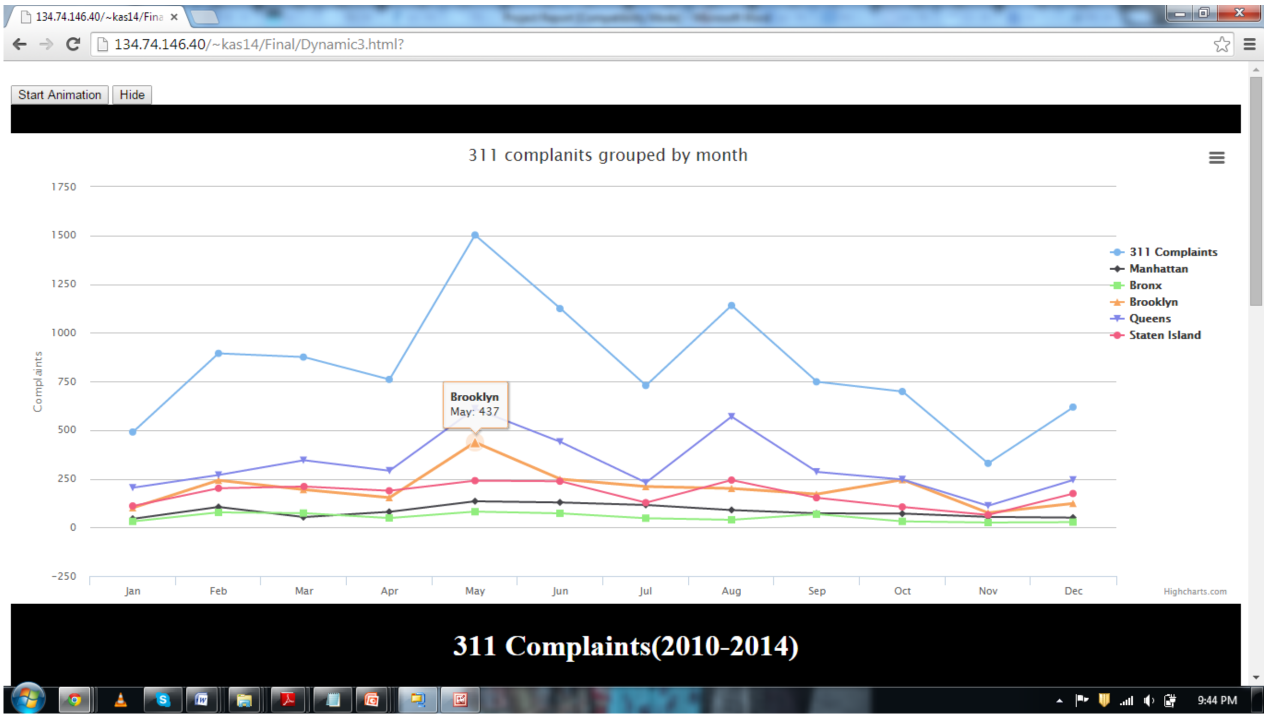
Data Analysis

- NYC Districts
- Total Number of Complaints
- [✓ NYC Complaints By Census tract]

-73.74961, 40.77617

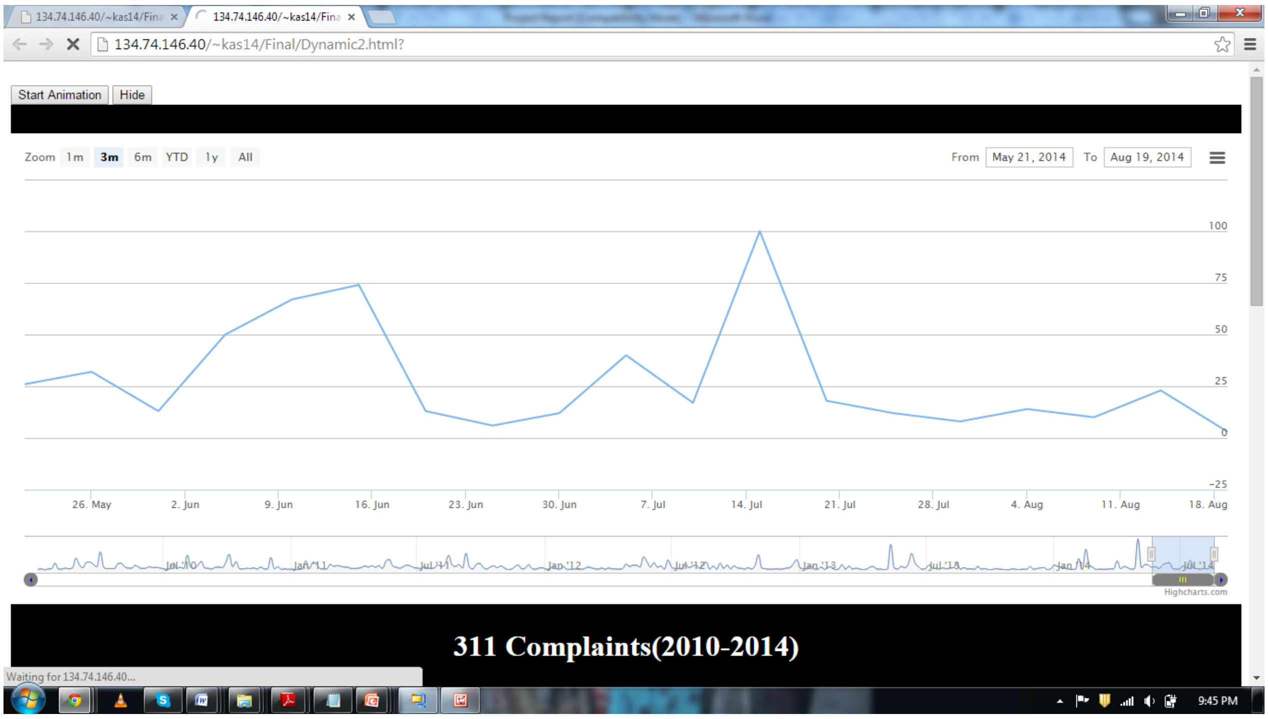
Area:	Douglas Manor-Douglaston-Little Neck
Borough:	Queens
Total population:	4519
Median Household Income:	84240
Age between 5-9 yrs:	338
Age between 25-34 yrs:	499
Age between 75-84 yrs:	164
Number of Complaints:	19

Highchart displaying 311 Complaints for all the 5 boroughs:



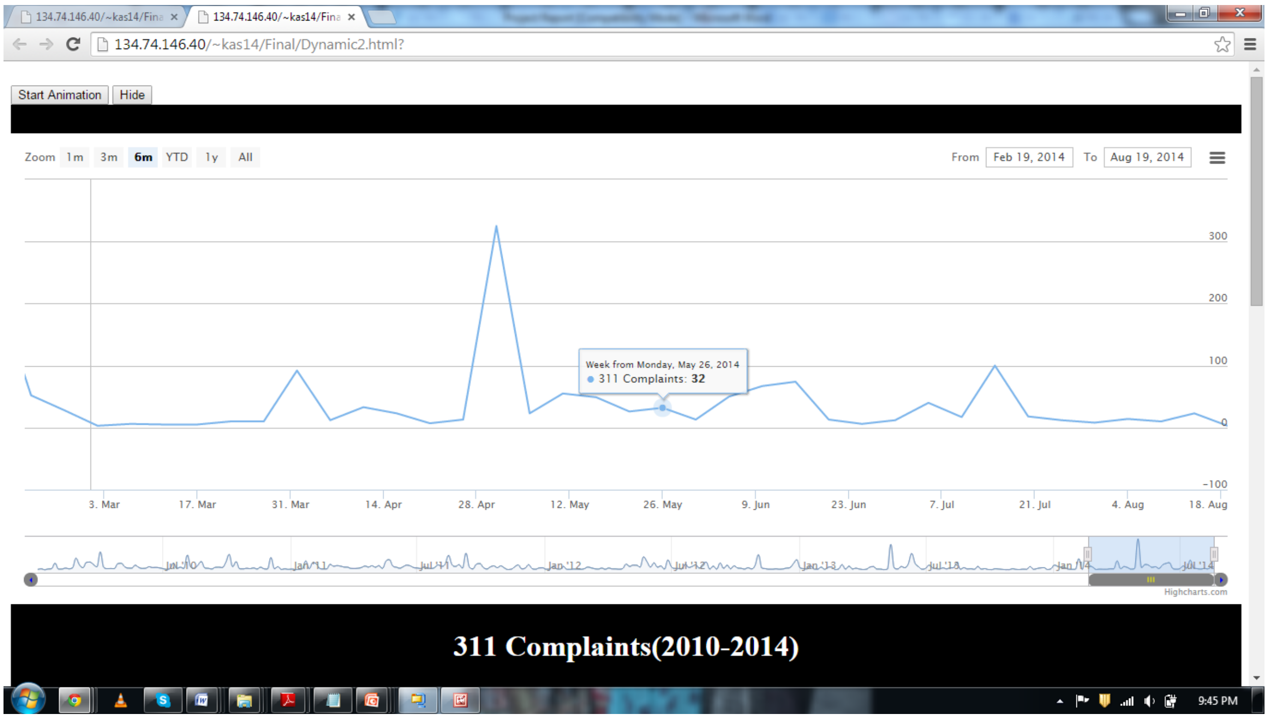
StockCharts :

Complaints showing for 3 months(3m):



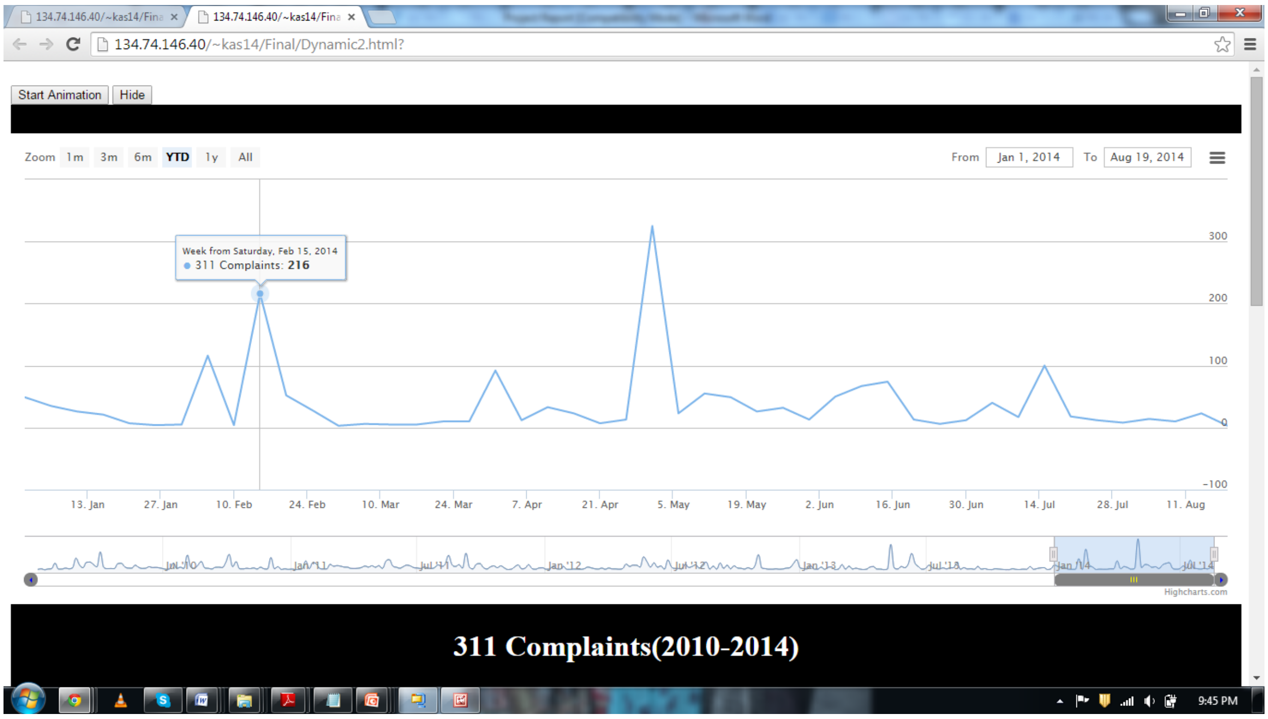
311 Complaints(2010-2014)

Complaints showing for 6 months(6m):

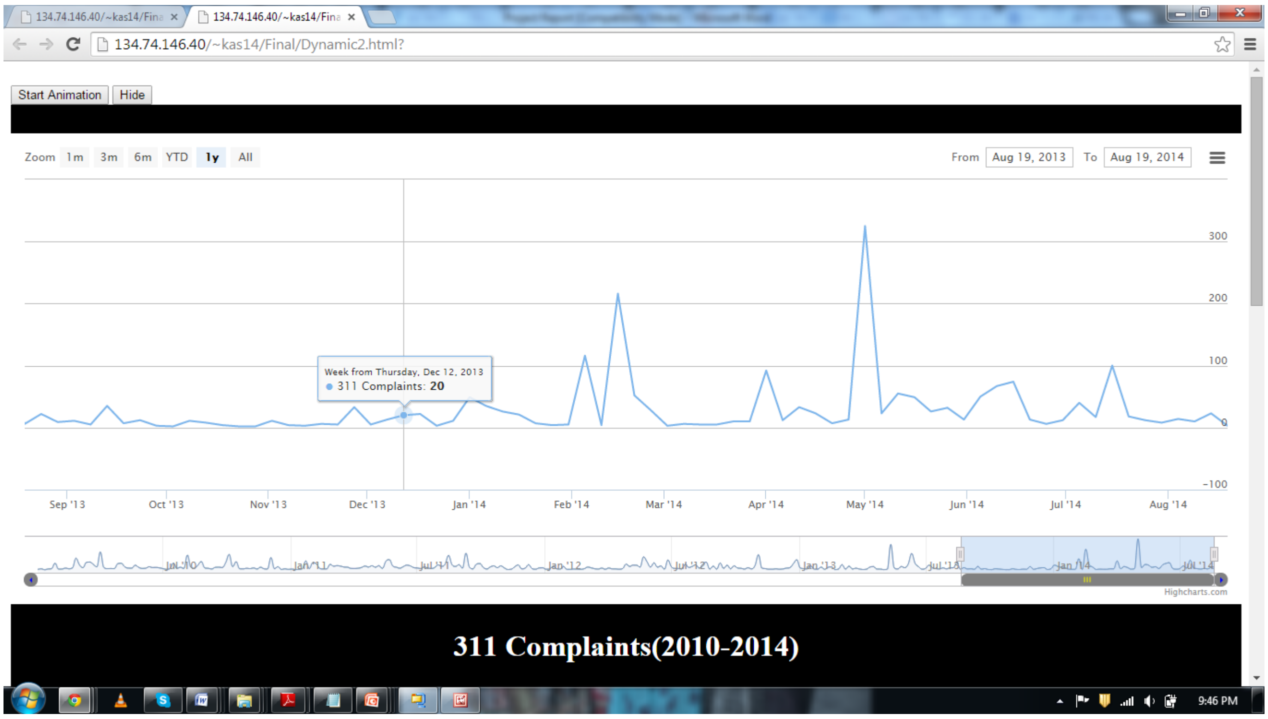


311 Complaints(2010-2014)

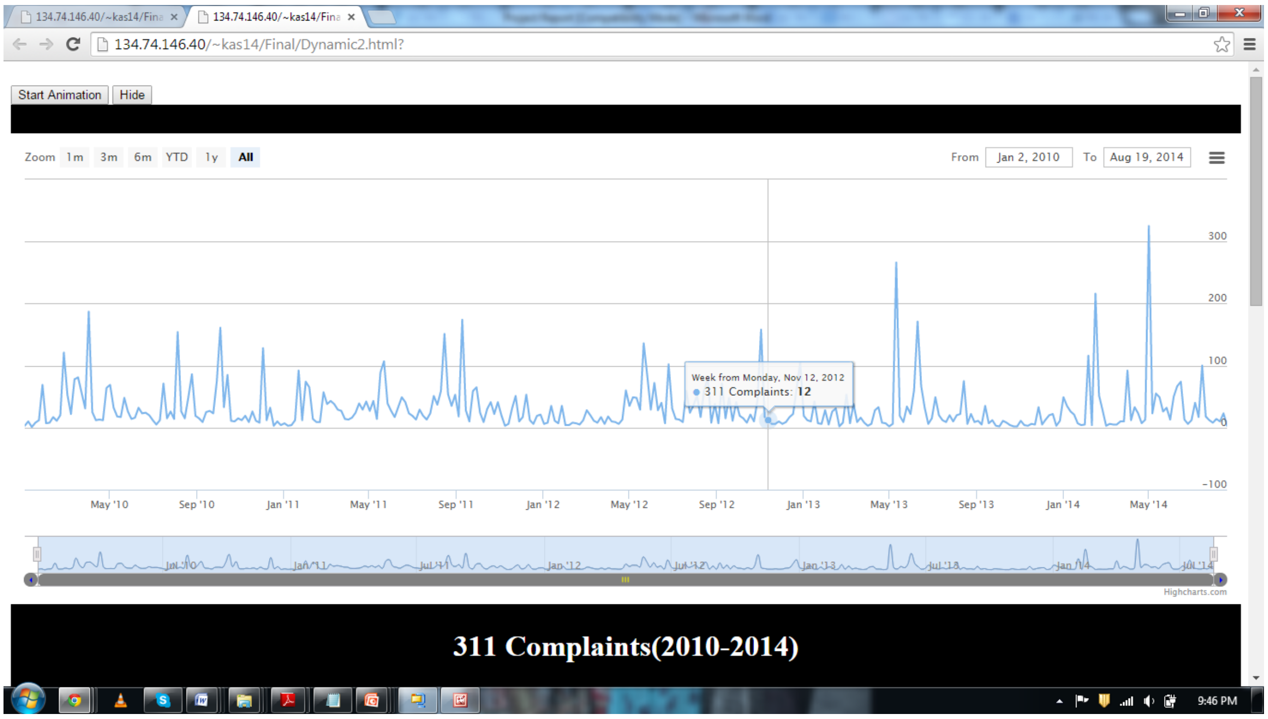
Complaints showing for YTD-year,date:



Complaints showing for 1 year(1y):



Complaints showing All (from Jan 2,2010 To Aug 19,2014):



Below are the steps involved in gathering and cleaning up of the data:

STEP1: Import CensusTractData.csv to PostgreSQL using CompileCT.java

- **Create a java file (Appendix A) to Convert CSV to SQL**

  In the Server:

  Copy CompileCT.Java to Final directory

  Copy CensusTractData.csv to Final directory

  javac CompileCT.java

  java CompileCT > CompileCT.sql

- **Create a Census tract Table in the database**

  In the database (/usr/local/pgsql/bin/psql -U ras14 -d d817)

```
CREATE TABLE censustract ( OBJECTID INT ,Borocode INT
,Shape_Leng NUMERIC ,Shape_Area NUMERIC,K_id INT
,TotalPopulation INT ,Areainsqmiles NUMERIC,
MedianHouseholdIncome INT ,AgeUnder5yrs INT ,Age5to9yrs INT
,Age10to14yrs INT ,Age15to17yrs INT ,Age18to24yrs INT
,Age25to34yrs INT ,Age35to44yrs INT ,Age45to54yrs INT
,Age55to64yrs INT ,Age65to74yrs INT ,Age75to84yrs INT
,Ageabove85yrs INT);
```

In the Server

```
psql -U ras14 -d d817 -f Compile311.sql
```

STEP2: Import Final311Data.csv to PostgreSQL using Compile311.java

- **Create a java file (Appendix B) to Convert CSV to SQL**

  In the Server:

  Copy Compile311.Java to Final directory

  Copy Final311Data.csv to Final directory

  ```
  javac Compile311.java
  java Compile311 > Compile311.sql
  ```

- **Create a Census tract Table in the database**

  In the database (/usr/local/pgsql/bin/psql -U ras14 -d d817)

  ```
  CREATE TABLE Data311 (UniqueKey INT , DATE Date ,Month INT, Day INT ,Year INT, TIME Time ,Key NUMERIC,daten NUMERIC ,Incident_Zip INT ,City Varchar(50) ,Status Varchar(50),Borough Varchar(50) ,Xcoordinate INT , Ycoordinate INT ,Latitude NUMERIC ,Longitude NUMERIC);
  ```

  In the Server

  ```
  psql -U ras14 -d d817 -f Compile311.sql
  ```

STEP3: Convert Shape File to a Geometry Table

- **Convert Shape file to an SQL file**

  ```
  /usr/local/pgsql/bin/shp2pgsql -s 2263 -c -I -D CT_Shape_and_Data.shp public.nycCT>nycCT.sql
  ```

- **Create a NYCCT(Geometry) Table in the database**

  ```
  psql -U ras14 -d d817 -f nycCT.sql
  ```

STEP4: Create a Geometry file based on 311 Complaints Coordinates (lat, long)

- **Create a Geom Column based on the coordinates of 311 Compliments and update the Data311 table with the geometry**

```
ALTER table Data311 ADD column geom Geometry(POINT,2263);
UPDATE Data311 SET geom = ST_SETSRID(ST_POINT(xcoordinate,ycoordinate),2263);
```

STEP5: Use *ST_Contains* to find interact between Data311 and CencusTract tables

- **The number of complaints in each borough (other St_Contains queries are provided in Appendix C)**

```
Select p.boroname, count(*) As numofcomplaints from nycct p, Data311 q where _ST_Contains(p.geom,q.geom) GROUP BY p.boroname;
```

STEP6 (Optional): Indexing to speed up the queries

- **Indexing Data311 and NYCCT**

```
select * into Data311_indexed  from Data311;
create index Data311_geom_idx on Data311_indexed using gist(geom);
select * into nycct_indexed  from nycct;
create index nycct_geom_idx on nycct_indexed using gist(geom);
```

- **Spatial Queries**

```
Select p.boroname, count(*) As numofcomplaints from nycct_indexed p, Data311_indexed q where _ST_Contains(p.geom,q.geom) GROUP BY p.boroname;
```

STEP7: Finding Number of complaints per Month in each Borough (Used in HighCharts)

- **Number of complaints in Brooklyn order by the Month**

```
SELECT count(*) FROM data311 WHERE borough=  'BROOKLYN' group by month order by month
```

- **Number of complaints in StatnIslnd using key (Key= Borough_Code+Cencus_Tract)**

```
SELECT count(*) FROM data311 WHERE key>5000000 group by month order by month;
```

STEP8: Combine five Days of Complaint (Five days Moving Average)

- **Add Bin Column to Data311 table in order to give each five days a Bin number so they can be summed up to get five day moving average**

```
ALTER table Data311 ADD column bin int;
UPDATE Data311 SET bin=daten/5;
```

STEP9: Connect the Data311 table to CencusTract

- **Creating a Key column in Data311 Table that connect 311 to CensusTract**

```
ALTER table Data311 ADD column K char(7);
Select q.uniquekey, p.BoroCt2010 into temp from nycct p, Data311 q where _ST_Contains(p.geom,q.geom);
Update data311 set k=boroct2010 from temp where data311.uniquekey=temp.uniquekey;
Update data311 set key=cast(k as int);
```

Summary and Conclusion:

In this day and age, nearly half of the world's population is now living in urban areas. Over the next 15 years, this figure is expected to rise up to 60% of the world's population. As more and more people continue to migrate towards cities, it is incredibly difficult to adapt the cities for this change overnight. This would only mean that the number of people in big cities affected by adverse weather conditions, such as excessive rainfall and the resulting flooding issues, will only continue to grow if nothing is done about if in the current era. Especially for coastal cities, and cities like New York, this problem is not only going to be more prevalent but also more difficult to deal with. Then there is also global warming and melting snowcaps and increasing sea leaves to take into account. This is going already having a very negative impact NYC and other coastal cities as can be seen in the rise of heat waves and storms and hurricanes.

With our project we aim to not only raise awareness for our cities agencies such as NYC – OEM and NYC – DOT but also to help them be more conscious of how dire the situation really is. NYC is really one of the worst hit cities every time such an event occurs and the efforts the city officials are putting in amending the current situation is clearly lacking. Surely enough if there are big hurricanes on the horizon, the city has evacuations protocols and people are safely moved before the storm hits, but the damage the city's infrastructure and economy takes is just way too high and unjustifiable. And the fact is that it does not need to be as bad it really turns out to be. The damage on the economy and infrastructure takes days to recover and the monetary toll is way too high, the best the city officials can do is speed up the planning on improving the ways how to minimize the impact from storms on the cities aging infrastructure. There is sever need for new planning and development within the city itself and around the cities coast line. One of the best things that can be done is to learn from countries like Netherlands which is one of the more pioneering nation when it comes with dealing with land below sea level, dealing with rising sea levels, dealing with high tides, and claiming land back from the sea. What we need is developing and deploying new and effective measures which will help keep safe not only the residents of our city but the city itself. Our work on this project is a far cry from helping solve the problem at hand, but it surely is a step towards the right direction, one which leads

Contribution by each team member:

1) Vijaya Kumar Venkatramanan: Worked on adding the date feature to the map. Added the data analysis section to the report. Used Jquery to design and add features. Added styling to the page. Performed sql queried to join tables. Made WMS and WFS calls to fetch census tract data.Worked on creating report.
2) Sina Rassaei Kasuak: Performed data gathering. Created highchart for Data Analysis. Worked with Vijaya Kumar to create Mapscripts files and styling of the web page. Integrated Highcharts to the Webpage. Worked on creating report.

3) Amruta Pawar: cleaning Data, importing data to postgresql, worked on coding part of highcharts with Sina, and report wrirting java program to import data. Worked on creating report.

4) Amit Sinha: Created the presentation and assisted in Project report.
5) Imen Harrouch: Contributed to the foundational idea and brain storming process and oversaw team members work as well as created the presentation and assisted in Project report.

Appendix A: CompileCT.java

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.*;


public class CompileCT {

        public static void main (String args[]) throws Exception
          {

                String filename = "/home/kas14/public_html/Final/CensusTractData.csv";
              String sqlstr="CREATE TABLE censustract " +
                                          "( OBJECTID INT ," +
                                          "K_id INT ," +
                                          "TotalPopulation INT ,"+

                                          "Areainsqmiles NUMERIC,"+

                                          "MedianHouseholdIncome INT ,"+
                                          "Age(Under 5 yrs) INT ,"+
                                          "Age(5-9 yrs) INT ,"+
                                          "Age(10-14 yrs) INT ,"+
                                          "Age(15-17 yrs) INT ,"+
                                          "Age(18-24 yrs) INT ,"+
                                          "Age(25-34 yrs) INT ,"+
                                          "Age(35- 44 yrs) INT ,"+
                                          "Age(45-54 yrs) INT ,"+
                                          "Age(55-64 yrs) INT ,"+
                                          "Age(65-74 yrs) INT ,"+
                                          "Age(75-84 yrs) INT ,"+
                                          "Age(above 85 yrs) INT )";



             // SELECT AddGeometryColumn('public','nycgrid','geom','2263','POINT',2);
              System.out.println(sqlstr);
              BufferedReader fr = new BufferedReader(new FileReader(filename));
              String s=null;
              int count =0 ;


              while ((s = fr.readLine()) != null)
              {

                  String[] dat=s.split(",");

                String OBJECTID = dat[0];

                String K_id = dat[1];
                String TotalPopulation = dat[2];
                String Areainsqmiles = dat[3];
                String MedianHouseholdIncome = dat[4];
                String AgeUnder5yrs = dat[5];
                String Age5to9yrs = dat[6];
                String Age10to14yrs = dat[7];
                String Age15to17yrs = dat[8];
                String Age18to24yrs = dat[9];
```

```
                        String Age25to34yrs = dat[10];
                        String Age35to44yrs = dat[11];
                        String Age45to54yrs = dat[12];

                        String Age55to64yrs = dat[13];
                        String Age65to75yrs = dat[14];
                        String Age75to84yrs = dat[15];
                        String Ageabove85yrs = dat[16];



                //          System.out.println("attr is" +OBJECTID+ " "+CB2010+ " " + BoroCode + " " + BoroName
+ " " + CT2010 + " "+ BCTCB2010 + " " + Shape_Leng + " " + Shape_Area + " " +BCT2011+ " ");


                        count++;
                        sqlstr= "insert into censustract values ("+OBJECTID+","


            +K_id+","+TotalPopulation+","+Areainsqmiles+","+MedianHouseholdIncome+","+AgeUnder5yrs+","
+Age5to9yrs+","+Age10to14yrs+","+Age15to17yrs+","+Age18to24yrs+","+Age25to34yrs+","+Age35to44yrs
+","+Age45to54yrs+","+Age55to64yrs+","


            +Age65to75yrs+","+Age75to84yrs+","+Ageabove85yrs+",2263));";
                         System.out.println(sqlstr);
                  }
                   System.out.println("count is " + count);
                  fr.close();
             }
}
```

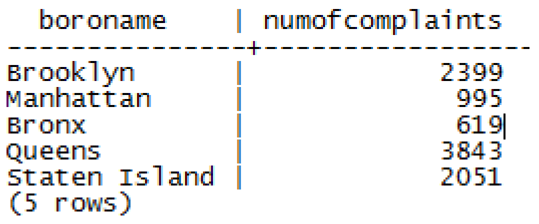
Appendix B: Compile311.java

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.*;


public class Compile311{

        public static void main (String args[]) throws Exception
         {
                        String filename = "/home/kas14/public_html/Final/Final311Data.csv";
             String sqlstr="CREATE TABLE Data311 " +
                                        "(Unique Key INT," +
                                        "DATE INT ," +
                                        "Month INT ," +
                                        "Day INT ," +
                                        "Year INT ," +
                                        "TIME INT ,"+
                                        "DAY1 NUMERIC,"+
                                        "Agency Varchar(50) ,"+
                                        "Incident Zip INT ,"+
                                        "City Varchar(50) ,"+
                                        "Status Varchar(10) ,"+
                                        "Community Board Varchar(50) ,"+
                                        "Borough Varchar(10) ,"+
                                        "Xcoordinate INT ,"+
                                        "Ycoordinate INT ,"+
                                        "Latitude NUMERIC ," +
                                        "Longitude NUMERIC)" ;


              // SELECT AddGeometryColumn('public','nycgrid','geom','2263','POINT',2);
               System.out.println(sqlstr);
               BufferedReader fr = new BufferedReader(new FileReader(filename));
               String s=null;
               int count = 0;


               while ((s = fr.readLine()) != null)
               {
                 // System.out.println("s is " + s);

                   String[] dat=s.split(",");
                   String UniqueKey = dat[0];
                  String DATE = dat[1];
                  String Month = dat[2];
                  String Day = dat[3];
                  String Year = dat[4];
                  String TIME = dat[5];

                  String Agency = dat[6];
                  String IncidentZip = dat[7];
                  String City = dat[8];
                  String Status = dat[9];
                  String CommunityBoard = dat[10];
                  String Borough = dat[11];
                  String Xcoordinate = dat[12];
```

```
                    String Ycoordinate = dat[13];
                    String Latitude = dat[14];
                    String Longitude = dat[15];

                    count++;
                     sqlstr=              "insert              into              Data311              values
("+UniqueKey+","+DATE+","+Month+","+Day+","+Year+","+TIME+","+Agency+","+IncidentZip+","

+City+","+Status+","+CommunityBoard+","+Borough+","+Xcoordinate+","+    Ycoordinate    +","+    Latitude
+","+Longitude+");";
                     System.out.println(sqlstr);

                }

              //   System.out.println("count is" + count);
                fr.close();

           }

}
```

Appendix C: ST_Contains Queries

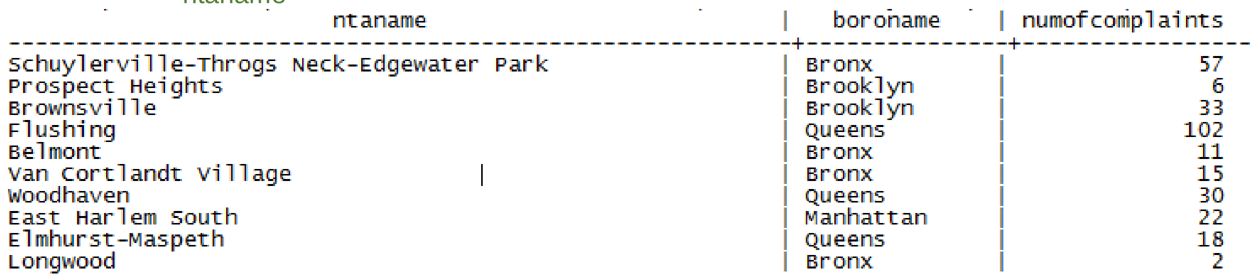
Select p.boroname, count(*) As numofcomplaints from nycct p, Data311 q where _ST_Contains(p.geom,q.geom) GROUP BY p.boroname;



```
   boroname     | numofcomplaints
----------------+-----------------
Brooklyn        |            2399
Manhattan       |             995
Bronx           |             619
Queens          |            3843
Staten Island   |            2051
(5 rows)
```

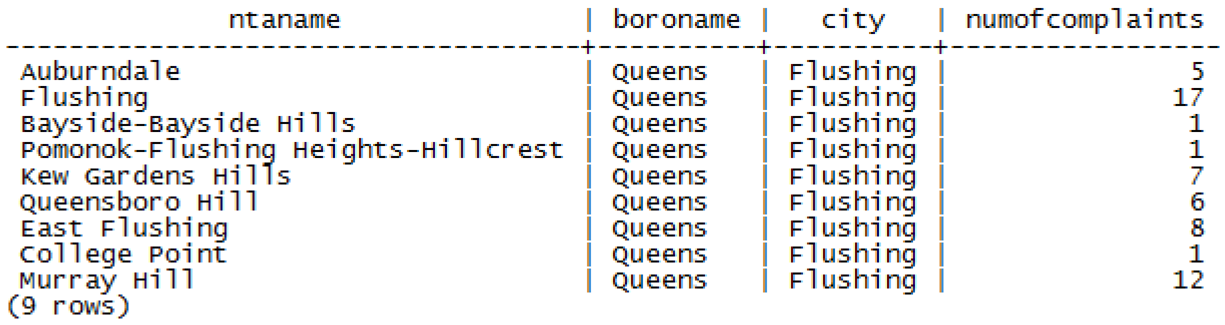
Select p.ntaname , p.boroname, count(*) As numofcomplaints from nycct p, Data311 q where _ST_Contains(p.geom,q.geom) GROUP BY p.ntaname,p.boroname;

ntaname



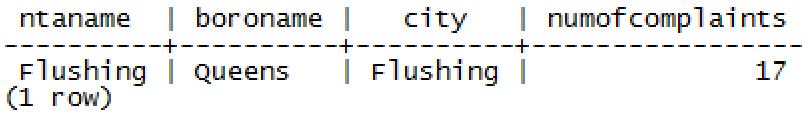
```
                      ntaname                                     |   boroname    | numofcomplaints
------------------------------------------------------------------+---------------+-----------------
Schuylerville-Throgs Neck-Edgewater Park                          | Bronx         |              57
Prospect Heights                                                  | Brooklyn      |               6
Brownsville                                                       | Brooklyn      |              33
Flushing                                                          | Queens        |             102
Belmont                                                           | Bronx         |              11
Van Cortlandt Village                                             | Bronx         |              15
Woodhaven                                                         | Queens        |              30
East Harlem South                                                 | Manhattan     |              22
Elmhurst-Maspeth                                                  | Queens        |              18
Longwood                                                          | Bronx         |               2
```

Select p.ntaname , p.boroname,q.city, count(*) As numofcomplaints from censustractshpdata p, Data311 q where _ST_Contains(p.geom,q.geom) AND q.city='Flushing' GROUP BY p.ntaname,p.boroname,q.city;



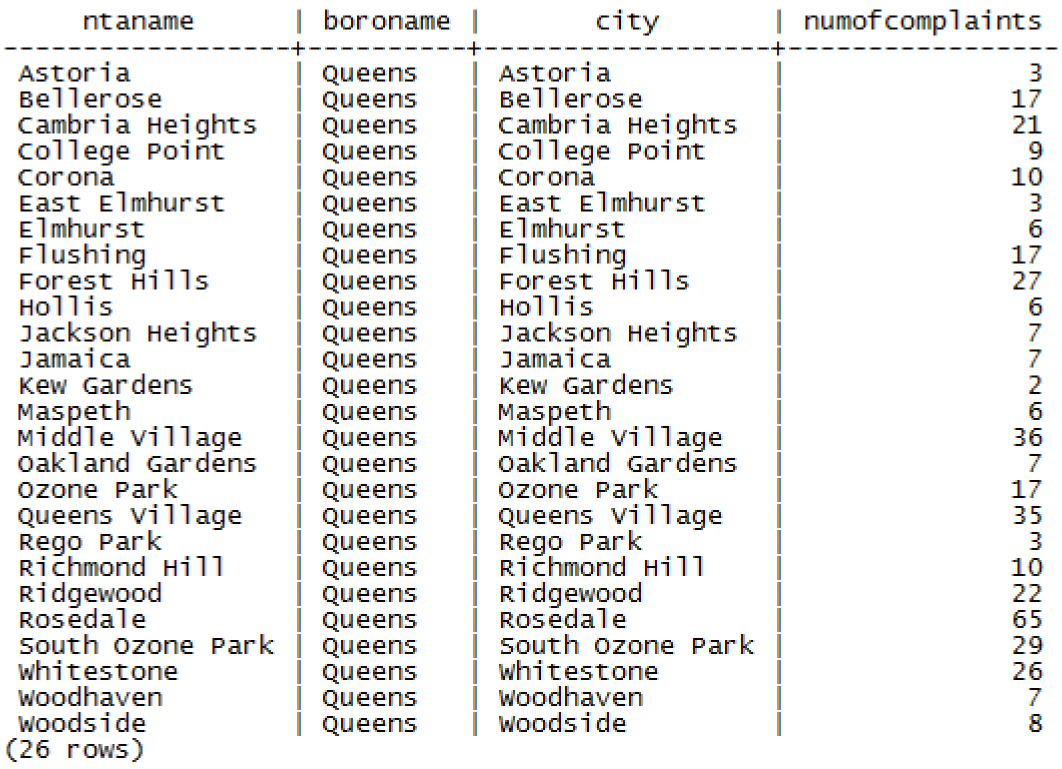
```
               ntaname                 | boroname |   city   | numofcomplaints
---------------------------------------+----------+----------+-----------------
 Auburndale                            | Queens   | Flushing |               5
 Flushing                              | Queens   | Flushing |              17
 Bayside-Bayside Hills                 | Queens   | Flushing |               1
 Pomonok-Flushing Heights-Hillcrest    | Queens   | Flushing |               1
 Kew Gardens Hills                     | Queens   | Flushing |               7
 Queensboro Hill                       | Queens   | Flushing |               6
 East Flushing                         | Queens   | Flushing |               8
 College Point                         | Queens   | Flushing |               1
 Murray Hill                           | Queens   | Flushing |              12
(9 rows)
```

Select p.ntaname , p.boroname,q.city, count(*) As numofcomplaints from censustractshpdata p, Data311 q where _ST_Contains(p.geom,q.geom) AND q.city='Flushing' AND q.city=p.ntaname GROUP BY p.ntaname,p.boroname,q.city;



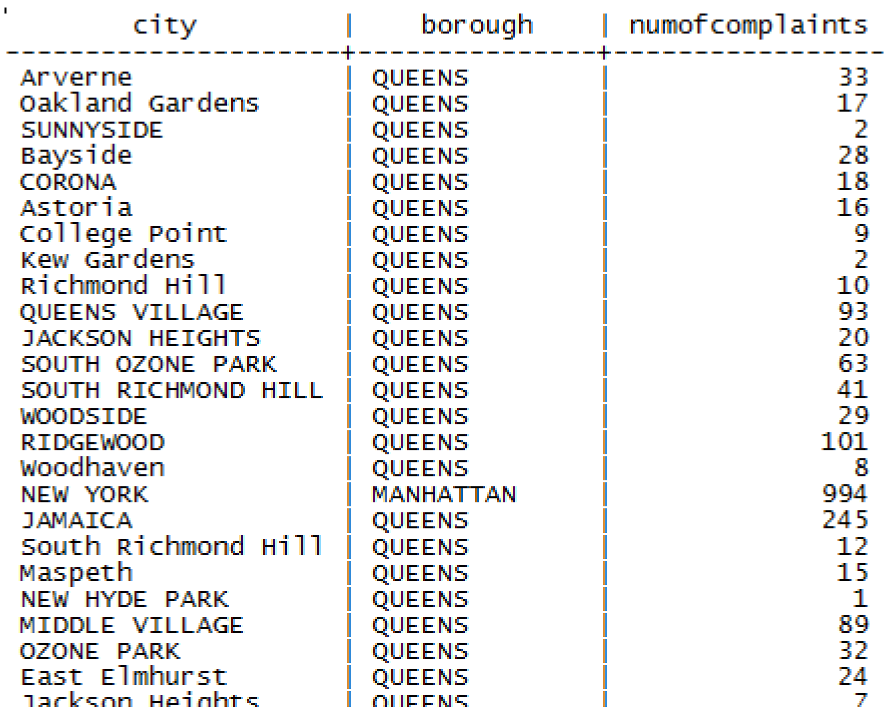
```
 ntaname  | boroname |   city   | numofcomplaints
----------+----------+----------+-----------------
 Flushing | Queens   | Flushing |              17
(1 row)
```

Select p.ntaname , p.boroname,q.city, count(*) As numofcomplaints from censustractshpdata p, Data311 q where _ST_Contains(p.geom,q.geom) AND q.city=p.ntaname GROUP BY p.ntaname,p.boro name,q.city;



```
      ntaname       | boroname |       city        | numofcomplaints
--------------------+----------+-------------------+-----------------
 Astoria            | Queens   | Astoria           |               3
 Bellerose          | Queens   | Bellerose         |              17
 Cambria Heights    | Queens   | Cambria Heights   |              21
 College Point      | Queens   | College Point     |               9
 Corona             | Queens   | Corona            |              10
 East Elmhurst      | Queens   | East Elmhurst     |               3
 Elmhurst           | Queens   | Elmhurst          |               6
 Flushing           | Queens   | Flushing          |              17
 Forest Hills       | Queens   | Forest Hills      |              27
 Hollis             | Queens   | Hollis            |               6
 Jackson Heights    | Queens   | Jackson Heights   |               7
 Jamaica            | Queens   | Jamaica           |               7
 Kew Gardens        | Queens   | Kew Gardens       |               2
 Maspeth            | Queens   | Maspeth           |               6
 Middle Village     | Queens   | Middle Village    |              36
 Oakland Gardens    | Queens   | Oakland Gardens   |               7
 Ozone Park         | Queens   | Ozone Park        |              17
 Queens Village     | Queens   | Queens Village    |              35
 Rego Park          | Queens   | Rego Park         |               3
 Richmond Hill      | Queens   | Richmond Hill     |              10
 Ridgewood          | Queens   | Ridgewood         |              22
 Rosedale           | Queens   | Rosedale          |              65
 South Ozone Park   | Queens   | South Ozone Park  |              29
 Whitestone         | Queens   | Whitestone        |              26
 Woodhaven          | Queens   | Woodhaven         |               7
 Woodside           | Queens   | Woodside          |               8
(26 rows)
```

Select q.city,q.borough, count(*) As numofcomplaints from censustractshpdata p, Data311 q where _ST_Contains(p.geom,q.geom) GROUP BY q.city,q.borough;



```
        city          |    borough    | numofcomplaints
----------------------+---------------+-----------------
 Arverne              | QUEENS        |              33
 Oakland Gardens      | QUEENS        |              17
 SUNNYSIDE            | QUEENS        |               2
 Bayside              | QUEENS        |              28
 CORONA               | QUEENS        |              18
 Astoria              | QUEENS        |              16
 College Point        | QUEENS        |               9
 Kew Gardens          | QUEENS        |               2
 Richmond Hill        | QUEENS        |              10
 QUEENS VILLAGE       | QUEENS        |              93
 JACKSON HEIGHTS      | QUEENS        |              20
 SOUTH OZONE PARK     | QUEENS        |              63
 SOUTH RICHMOND HILL  | QUEENS        |              41
 WOODSIDE             | QUEENS        |              29
 RIDGEWOOD            | QUEENS        |             101
 Woodhaven            | QUEENS        |               8
 NEW YORK             | MANHATTAN     |             994
 JAMAICA              | QUEENS        |             245
 South Richmond Hill  | QUEENS        |              12
 Maspeth              | QUEENS        |              15
 NEW HYDE PARK        | QUEENS        |               1
 MIDDLE VILLAGE       | QUEENS        |              89
 OZONE PARK           | QUEENS        |              32
 East Elmhurst        | QUEENS        |              24
 Jackson Heights      | QUEENS        |               7
```