

# Principles of Urban Informatics

## Assignment 9

Posted on: 11/10/2014  
Due Date: 11/17/2014

### Introduction

The purpose of this assignment is to get familiarized with map plotting techniques. You will use bokeh/python to produce plots, whose documentation is available at <http://bokeh.pydata.org/>. In this assignment we are going to create map visualizations that help understand 311 data for NYC. Steps to complete this assignment include downloading data from the *NYC Open Data* website into a *CSV* file, processing data with Python to make it ready for visualization, and finally creating visualizations with *bokeh* to do spatial data analysis.

### Data

#### 311 Data

311 data can be found at the url <http://goo.gl/i3ltE8>. Download a CSV file of that data for the following period (complaints created in): *Jun/01/2013 to Aug/31/2013*. Use this local dataset to run all problems in this assignment.

*Note: you should NOT remove any columns of the CSV file when you download the dataset, otherwise it will be impossible to replicate your results..*

#### Zip shape files from US Census Bureau

In order to do this assignment, you will need a shape file with the description of the NYC zip code areas boundaries. Such file is available in <https://www.census.gov/cgi-bin/geo/shapefiles2013/main> (select ZIP Code Tabulation Areas from the dropdown menu). The zip file contains boundaries information for the entire US territory. To restrict it to only NYC, use the zip code file available in [http://vgc.poly.edu/projects/gx5003-fall2014/week9/lab/data/zip\\_borough.csv](http://vgc.poly.edu/projects/gx5003-fall2014/week9/lab/data/zip_borough.csv). As pointed in the previous Lab, you will have to iterate over all shapes, considering only the ones with NYC zip codes.

Use the library *shapefile* to handle the shape files. Feel free to use the python code used in the Lab session.

## Problem 1 - Top agency in each NYC zip code

Create a choropleth map for NYC in which the shape color for each zip code represents its top agency in number of complaints. An example for a small set of agencies is in Figure 1. Choose an appropriate color scale to visualize the agencies.

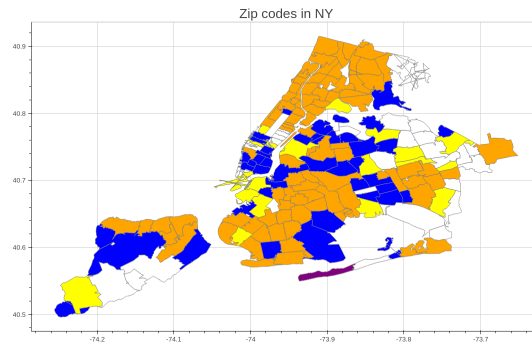


Figure 1: Top agencies in terms of number of complaints per zip code in NYC

The map must include legends (color for each agency) and support hover: when the mouse is over a zip code, the tooltip must show the zip code, the top agency name and the number of complaints for that agency. An example of hover interaction can be seen at <http://bokeh.pydata.org/docs/gallery/texas.html> and more instructions can be found at <http://bokeh.pydata.org/tutorial/advanced.html>.

Your program must run as:

```
python problem1.py [complaints] [zip_borough] [shapefile]
```

where *[complaints]* should be replaced by the name of the CSV file with 311 data, *[zip\_borough]* by the file with zips for NYC, and *[shapefile]* points to the shape file for zip codes in US.

## Problem 2 - Comparing two agencies

Consider Figure 2, where the polarity between Republicans and Democrats is shown for the US 2004 presidential election. In this map, the two measures (number of voters for Republicans and for Democrats) are represented with a gradual color scale from red to blue.



Figure 2: 2004 Presidential Election

You will create an analogous map for NYC to compare two agencies in terms of number of complaints for each zip code. Given two agency names as parameters, your program will compute their number of complaints per zip code, and plot a map to show that ratio, similar to an election map. Choose a color scale that makes the comparison possible.

Your program must run as:

```
python problem2.py [complaints] [zip_borough] [shapefile] [agency1] [agency2]
```

where *agency1* and *agency2* are the name for the agencies to be compared.

### Problem 3

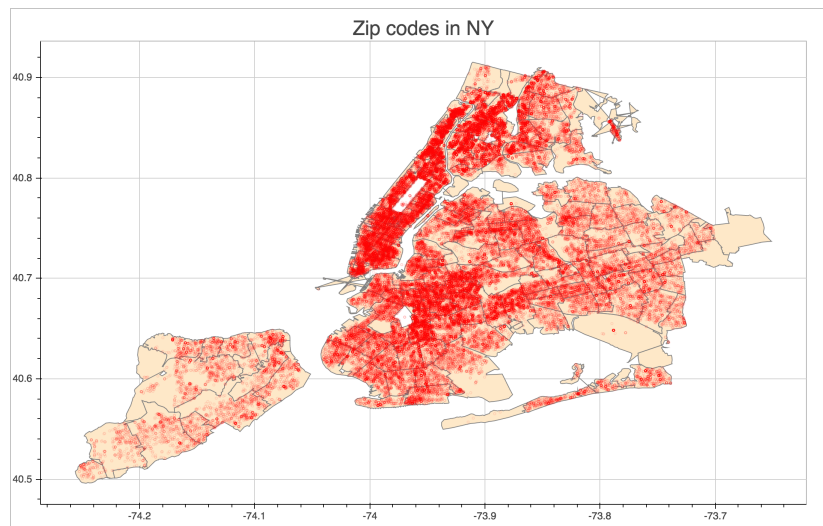


Figure 3: 311 complaints visualization example

In the previous Lab session, you saw how to plot a map of 311 complaints in NYC using the Bokeh library. Figure 3 shows a plot of 311 complaints; each complaint is plotted as a circle on the map. If you have a large number of circles, however, the visualization becomes problematic. A simple solution is to limit the number of circles drawn on the screen. In this problem, instead of drawing each complaint as a circle, you have to do the following:

- Divide the domain in  $n \times n$  cells. For each cell, calculate the number of complaints that happened there.
- For each cell, draw a circle in the center of the cell. The size of the circle must be proportional to the number of complaints in the grid.

The complexity of the visualization is no longer attached to the number of complaints. It is, instead, bounded by the number of cells in the grid. You must be careful when selecting the range of sizes for the circle. Also, consider drawing non-opaque circles.

Your program must run as `python problem3.py n [complaintsfile] [zipboroughfile] [shapefile]`, where  $n$  is the number of subdivisions along an axis, `complaintsfile` is the name of the 311 complaint file, `shapefile` is the name of the shape file (downloaded as instructed above) and `zipboroughfile` is the csv file with NYC zip codes.

## Problem 4

In the last problem, you had to draw a circle for each cell in the grid. Now, draw a circle for each zip code in NYC. The size of the circle must be proportional to the number of complaints in the zip code. Also, each circle must be positioned in the centroid of the zip code area.

Your program must run as `python problem2.py [complaintsfile] [zipboroughfile] [shapefile]`, where `complaintsfile` is the name of the 311 complaint file, `shapefile` is the name of the shape file (downloaded as instructed above) and `zipboroughfile` is the csv file with NYC zip codes.

## How to submit your assignment?

Your assignment should be submitted using the NYU Classes system. You should submit all the requested files in each problem in a zip file named `NetID_assignment_9.zip`, where you should change *NetID* by your NYU Net Id.

## Grading

We should be able to reproduce your plots by running your scripts with the specified parameters. The grading is going to be done based on your plots and your code. It is part of the assignment to understand (and research if necessary) the concepts asked. Try to test your code before submitting: your script solves the problem when the plot is created as requested.

## References

- *bokeh reference (installation and documentation at Google)*: <http://bokeh.pydata.org/>
- *Color scales reference*: <http://colorbrewer2.com>
- *Python Shapefile Library*: <https://code.google.com/p/pyshp/>

*Googling* for example code and similar plots is encouraged.