Principles of Urban Informatics
Assignment 7

Posted on: 10/27/2014
Due Date: 11/03/2014

Introduction

In this assignment we are going to create 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 *Matplotlib* to do visual data analysis.

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*. It is part of the assignment to figure out how to download the CSV file so that it contains ONLY data inside the specified period, so if you have doubts it is ok to google or ask other students how to download the data. 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. We must be able to run your results by using a dataset that only includes complaints in the date range, with ALL the columns.

Problem 1

In a previous assignment we processed *311* data to get the *top-k* complaint types. For a given period of time, we computed the most common complaint types and how many complaints there were for each type. The result was a list of complaint types and associated counts.
A more compelling way of communicating the same information is through visualization. An example is the bar chart below, that depicts the *top-5* complaint types for a period of time. Each bar corresponds to one complaint type, and the height conveys volume: how many complaints happened with that type.

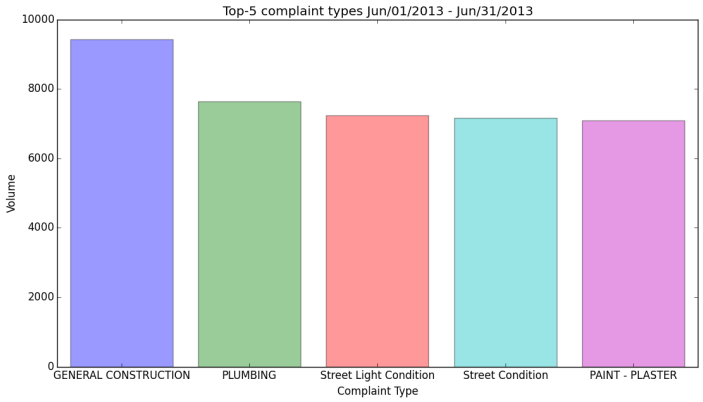

Figure 1: Bar chart example for top-5 complaint types in terms of volume. The date range is NOT the same as the dataset used in this assignment.

1) Write a python script (called *problem1_1.py*) to plot a bar chart that shows the volume of agencies: $\{NYPD, DOT, DOB, TLC, DPR\}$.
The plot's axes must have labels, and it must be possible to identify what agency corresponds to each bar.

   Your program must execute as: *python problem1_1.py [input.csv]*, where *[input.csv]* should be replaced by the name of the CSV file with 311 data. Example: *python problem1_1.py 311data.csv* should show that bar chart for the mentioned agencies.

2) Write a python script (called *problem1_2.py*) to plot a bar chart for the *top-k* agencies in terms of number of complaints. The plot's axes must have labels, and it must be possible to identify what agency corresponds to each bar.

   Your program must execute as: *python problem1_2.py [input.csv] [k]*, where *[input.csv]* should be replaced by the name of the CSV file with 311 data, and *[k]* specifies the number of bars (*top-k* agencies) to show. Example: *python problem1_2.py 311data.csv 5* should show that bar chart for the *top-5* agencies.

In a separate file *problem1.txt*, write:

1) A small paragraph describing any choices you made while processing the data to produce the plots.

2) A small paragraph with interesting observations that you can make using the bar charts you produced.

Extra credit) Write a python script (called problem1_extra.py) to plot one or multiple histograms to reveal interesting patterns in the *311* data for different times of day.
Interesting patterns could be agencies that have the same number of complaints

independent of the time of day, contrasted with agencies that have considerable variance in the number of complaints for different times of day.

In a separate file *problem1_extra.txt*, write:

1) Instructions of how to run your script. At least the CSV file name must be a command-line parameter of your script.

2) A small paragraph with interesting findings made possible by your visualization.

Problem 2

In this problem, you will have to plot timeseries. Such visualization allows us to observe how a given variable behaves over time. Figure 2 shows an example of a timeseries: the volume of complaints of three different agencies is plotted over a period of one month.

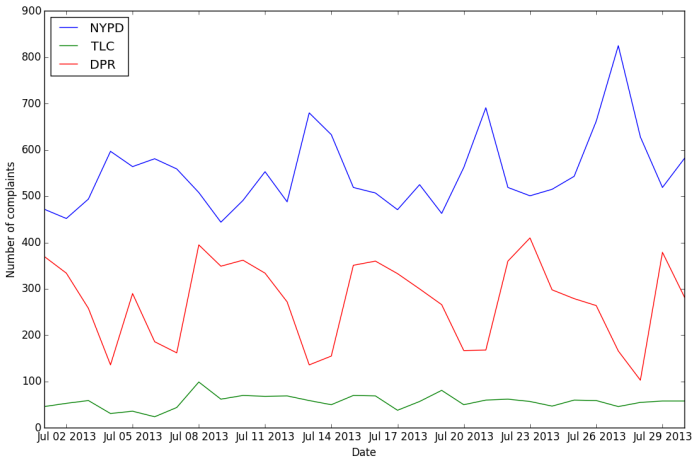

Figure 2: Plot with number of complaints over time for three city agencies.

The following two items ask you to create timeseries based on the 311 dataset.

1) Write a python program (called problem2_1.py) that creates a plot with 3 timeseries. The idea is to observe how the number of 311 complaints varies over time, considering 3 different city agencies: $\{NYPD, TLC, DPR\}$. The time resolution of the timeseries should be days; in other words, all complaints in one day should be aggregated into one value. Figure 2 shows an example.

   Your program must execute as following: *python problem2_1.py [input.csv]*, where *[input.csv]* should be replaced by the name of the file which contains the 311 data.

2) In the first item, you had to consider 3 pre-defined agencies. This time, you will have to consider the *top-k* agencies with most complaints during the period of the timeseries. The time resolution should be days.

   Your program must execute as following: *python problem2_2.py [input.csv] [k]*, where *[input.csv]* should be replaced by the name of the file which contains the 311 data, and *[k]* by the number of agencies.

Extra Credit) Create a timeseries plot that reveals interesting patterns in the 311 dataset. You can choose any date range, number of agencies and time resolution. Is it possible to see any pattern, depending of hour or day? What about number of complaints in Holidays or in disaster events (e.g. Sandy)?

In a separate file *problem2_extra.txt*, write:

1) Instructions on how to run your script. At least the CSV file name must be a command-line parameter of your script.

2) A small paragraph with interesting findings made possible by your visualization.

Problem 3

In this portion of the assignment we are going to use one of the most common visualizations to understand the relationship between two quantitative variables, namely, a scatterplot. An example of scatterplot can be found at `http://matplotlib.org/examples/lines_bars_and_markers/scatter_with_legend.html`.

We are going to study the relationship between the number of complaints per zip code and the zip code population. For this you are going to use the 311 data previously mentioned and zip code data population data that you can download at `http://vgc.poly.edu/projects/gx5003-fall2014/week7/data/zipCodePopulationData.csv`. Your job is:

1) Write a python script called *problem3_1.py* that receives both the names of the complaints dataset (as downloaded from NYC open data) and the zip code population files as command-line parameters and plots for each zip code the zip code population (x-axis) and the number of complaints (y-axis) in the given datasets. You should follow the principles saw in class to produce a good plot.

2) Now, imagine that we want to modify the previous plot to convey another piece of information: the agency that generated the most complaints in the zip code (in addition to displaying the relationship between population and number of complaints per zip code). Assume that we as analysts want to focus on the following agencies: $\{NYPD, DOT, DOB, TLC, DPR\}$. How can you modify your plot to do it? Write a script called *problem3_2.py* that does the job. This is script should also receive the complaint data and zip code population files via command-line.

In a separate file called *problem3.txt*, write:

1) A small paragraph describing any choices you made while processing the data.

2) A small paragraph describing any choices you made when producing the plot.

Extra Credit) Can you find anything interesting from these plots? Describe your findings supported by them in a separate file called *problem3_extra.txt*. Your observations can contain additional questions that you would like to investigate about the data and ideas on how would you proceed to answer them.

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 justifications for the choices you made. Try to test your code before submitting: your script solves the problem when the plot is created as requested. The grading for the extra credits will be subjective: plots that do do not reveal any interesting patterns will be ignored, whereas plots that provide non-trivial insights explicitly explained in your answer will receive maximum grade.

References

- *311 data*: `http://goo.gl/i3ltE8`
- *Matplotlib reference (installation and documentation)*: `http://matplotlib.org/`

Consult the Matplotlib documentation to find out how to create each plot. *Googling* for example code for the requested plots is encouraged.