CIS 210, Fall 2016
Introduction to Computer Science

Earthquake Analysis

This assignment is due at 5pm on Friday, November 11. Use Canvas to turn in eqanalysis.py

Purpose

This project builds on the statistical analysis material we have covered in class (and in chapter 5 in the textbook) and the material in chapter 7 of the textbook on data mining. It is a relatively larger project than we have tackled in the past, although many parts are somewhat repetitive.

Pair Assignment

You are encouraged to use *pair programming* to complete this assignment. You can share code with one other classmate, who should be listed as an author in your program docstring. You may discuss general approach and design with other students, without sharing code; cite them in your docstring clearly distinguishing design discussion from code authorship.

You have to create 6 functions in this assignment, as well as understand how a number of other provided functions and modules work. It will take concerted, sustained effort to complete this assignment successfully.

How can a computer science novice tackle data science?

As we discussed in lecture, there is a rapid increase in the amount of published data in the world; some say the growth is exponential. Data analytics is the science of examining raw data with the purpose of drawing conclusions about (inferring information from) the raw data. Data analysis is the process of inspecting, cleaning, transforming, and modeling data with the goal of discovering useful information.

We discussed several central tendency measures of data, a couple of dispersion measures, and frequency tables for showing some aspects of data. The purpose of this project is for you to apply these measures, and to use turtle graphics for plotting the raw data, to assist an analyst in understanding earthquake data here in the Pacific Northwest.

Requirements

I have provided some code to get started. You don't have to use it, but if you do, the FIXME comments indicate places where you will need to add or modify code.

Your program will be executed as follows from the command or terminal window:

```
$ python3 eqanalysis.py file.csv plot clusters
$ python3 eqanalysis.py file.csv plot magnitudes
$ python3 eqanalysis.py file.csv plot depths
$ python3 eqanalysis.py file.csv analyze clusters
$ python3 eqanalysis.py file.csv analyze magnitudes
$ python3 eqanalysis.py file.csv analyze depths
```

The first three will generate *dot plots* on a turtle graphics window, while the last three will display mean, median, standard deviation, and frequency table information on the standard output; for cluster analysis, the mean, median, and standard deviation information is shown for each cluster.

How to proceed

I described several statistical functions in lecture, and have provided them here; you should place this file in the directory in which you are running *eqanalysis.py*.

You should *thoroughly* understand the material in chapters 5 and 7 of the textbook. The authors show you how to use the turtle.dot() function to draw dots on a graph; you will need to do this for the plotting required in this assignment.

In order to use the background map of the Pacific NorthWest upon which to draw your dots, you have convert an event's longitude and latitude into x and y pixel coordinates on the turtle window. I have provided a function in the starter code, xy_calculate(lon, lat), that returns a tuple with the correct x and y coordinates to feed into turtle.goto().

Besides the statistical functions, I have a number of data files that you will want to download into the same directory:

- A map of the Pacific Northwest which is used as the background in a turtle window
- A CSV file containing 10,000 events
- A CSV file containing 100 events
- A CSV file containing 500 events
- A CSV file containing 1,000 events
- A CSV file containing 2,000 events

- [A CSV file containing 3,000 events]
- [A CSV file containing 4,000 events]
- [A CSV file containing 5,000 events]

Examples using EqData01.csv

$ python3 eqanalysis.py EqData01.csv plot magnitudes

$ python3 eqanalysis.py EqData01.csv plot depths

$ python3 eqanalysis.py EqData01.csv plot clusters

$ python3 eqanalysis.py EqData01.csv analyze magnitudes

```
Analysis of magnitude data
  Mean magnitude = 1.3
  Median magnitude = 1.1
  Standard deviation = 0.57
ITEM    FREQUENCY
  0.6     3
  0.7     8
  0.8     11
  0.9     8
  1.0     10
  1.1     13
  1.2     4
  1.3     6
  1.4     4
  1.5     6
  1.6     4
  1.7     4
  1.8     3
  1.9     4
  2.0     1
  2.1     5
  2.3     3
  2.4     2
  2.5     3
  2.6     1
  2.9     2
```

$ python3 eqanalysis.py EqData01.csv analyze depths

```
Analysis of depth data
  Mean depth = 6.1 miles
  Median depth = 4.3 miles
  Standard deviation = 6.87 miles
ITEM    FREQUENCY
  0.0     31
  0.2     1
  0.3     1
  1.1     3
  1.3     1
  1.6     2
  2.3     1
  2.6     1
  2.7     1
  2.8     2
  3.1     1
  3.3     2
  3.5     2
  3.8     1
  4.0     2
  4.3     2
  4.4     1
  4.9     1
  5.0     1
  5.2     1
  5.7     1
  5.9     1
  6.3     2
  6.5     2
  6.8     1
  7.0     1
  7.2     1
  8.5     1
  8.7     1
  8.9     2
  9.2     1
  9.4     1
  9.5     1
  9.7     2
 10.0     1
 10.1     1
 10.2     1
 10.4     1
 10.5     1
 10.8     1
 11.2     1
 11.3     1
 11.4     1
 11.7     1
 11.8     1
 12.0     1
 12.1     1
 12.2     1
```

```
12.4    1
12.5    1
12.6    1
12.7    1
12.9    1
13.1    1
14.1    1
14.7    1
15.1    1
15.2    1
15.3    1
19.8    1
30.7    1
33.8    2
```

$ python3 eqanalysis.py EqData01.csv analyze clusters

```
Analysis of cluster 0
  Analysis of magnitude data
    Mean magnitude = 1.8
    Median magnitude = 1.9
    Standard deviation = 1.48
  Analysis of depth data
    Mean depth = 21.6 miles
    Median depth = 21.6 miles
    Standard deviation = 17.25 miles
Analysis of cluster 1
  Analysis of magnitude data
    Mean magnitude = 1.3
    Median magnitude = 1.1
    Standard deviation = 0.51
  Analysis of depth data
    Mean depth = 7.2 miles
    Median depth = 4.4 miles
    Standard deviation = 10.48 miles
Analysis of cluster 2
  Analysis of magnitude data
    Mean magnitude = 1.2
    Median magnitude = 1.1
    Standard deviation = 0.53
  Analysis of depth data
    Mean depth = 5.8 miles
    Median depth = 5.1 miles
    Standard deviation = 5.07 miles
Analysis of cluster 3
  Analysis of magnitude data
    Mean magnitude = 1.6
    Median magnitude = 1.6
    Standard deviation = 0.71
  Analysis of depth data
    Mean depth = 1.8 miles
    Median depth = 0.0 miles
    Standard deviation = 2.71 miles
Analysis of cluster 4
  Analysis of magnitude data
    Mean magnitude = 1.2
    Median magnitude = 1.1
    Standard deviation = 0.43
  Analysis of depth data
    Mean depth = 8.7 miles
    Median depth = 8.9 miles
    Standard deviation = 6.65 miles
Analysis of cluster 5
  Analysis of magnitude data
    Mean magnitude = 1.6
    Median magnitude = 1.7
    Standard deviation = 0.40
  Analysis of depth data
    Mean depth = 2.9 miles
    Median depth = 1.1 miles
    Standard deviation = 3.98 miles
```

Grading rubric

Functional correctness					**60**
	plot_magnitudes()	10	10 if meets spec, 5 = minor discrepancy, 0 = ignored spec		
	plot_depths()	10	10 if meets spec, 5 = minor discrepancy, 0 = ignored spec		
	plot_clusters()	10	10 if meets spec, 5 = minor discrepancy, 0 = ignored spec		
	analyze_magnitudes()	10	10 if meets spec, 5 = minor discrepancy, 0 = ignored spec		
	analyze_depths()	10	10 if meets spec, 5 = minor discrepancy, 0 = ignored spec		
	analyze_clusters()	10	10 if meets spec, 5 = minor discrepancy, 0 = ignored spec		
Other requirements					15
	Header docstring	5	5 = as specified, 3 = minor issue, e.g., as #comment, 0 = missing or incomplete		

	Program style and readability	10	10 Good variable names, indentation, etc --- very readable code, 8 = minor issues, such as inconsistent indentation, 5 = major issues that interfere with readability of code, 0 = unreadable mess		
Total					75

I can't anticipate all issues that may be encountered in grading, so points may be deducted for other issues not listed in the rubric. A program that does not compile and run (e.g., because of a syntax error) starts with 0 points for functional correctness, but the grader at his or her discretion may award some partial credit.