Ervin Sarmiento

Probability and Applied Statistics

Apache Stats Library and JFreeChart on Plotting, Salting, Smoothing and Graphing Report

This report explains the process, design and implementation of the Salting Smoothing and Graphing programs in conjunction with applying different libraries to create a visual representation of the graph and to ease calculation.

First, since I have created a plotter program that plots my function of y = mx + b from Project 1, I used the same codes and modified it to fit the requirements of this version of the plotter. The plotter didn’t involve using any other outside libraries. I was able to export the required x and y values placed into a comma separated value file (csv) named “Plotted Excel.csv” (figure 1) where it can be pulled by the modified salter program that is specifically modified for this project. The amount of x with a corresponding y values amounted to 100 pair to develop points that is graphable.

Text

Description automatically generated

*(figure 1)*

Second, using a csv importer implementation that I developed, I was able to extract the previous 100 points of my function from the csv file “Plotted Excel.csv”. I then proceeded to split the values so I only get the index that has the y values and then inputted it in a String Array then converted into a Double Array afterwards. After that, I “salted” the y values in which I implemented a random Double numbers from the range of -10 to 10 that is added or subtracted to each and every points of y that was extracted previously from the csv reader (figure 2). Afterwards, I developed a csv exporter to export the newly created values of y named “Salted Plotted Excel.csv” to which my Smoother program can pull from to smooth the salted y value data.

Text

Description automatically generated

*(figure 2)*

Third, using the same csv importer that I have created, it imported the csv file named “Salted Plotted Excel.csv” that contained the x values and salted y values from the previous salter program. I then created a function in this java class named ySmoother in which it takes an ArrayList of doubles of the yValue that I previously extracted from the csv importer. I then used an Apache Stats Library class called “DescriptiveStatistics” in order for me to smooth the salted yValues which was a different case from the previous project where as a class, we used our own way to implement the average to get the smoothed data (figure 3). After getting the smoothed yValues, it is then stored in an array of Double to be pulled by my Grapher program.

A screenshot of a computer

Description automatically generated with medium confidence

*(figure 3)*

Fourth, using JFreeChart library to implement graphing utility to express the function of my choosing, y = mx + b, showing the salting process and at the same time, in the same graph, show the smoothing process of the graph (figure 4). The red line signifies the salted points and we can observe the erratic up and down behavior of the salted points. The blue line signifies the smoothed points. It’s not a straight point as observed, but it provides a better clarity in terms of reading the graph versus the salted points.

Graphical user interface, chart, scatter chart

Description automatically generated

*(figure 4)*

Addendum:

Overall, this project has taught me how to import external libraries and utilize JAR files. It shows just how much learning experience is learned and is needed in order to provide a concrete visualization of a graph using Java programming language. As I was learning how to utilize and implement the Apache and JFreeChart libraries, I learned a lot on the different approaches of graphing such as scatter plots, line charts, bar charts, etc. that can be utilize. Apache has very powerful set of tools that can be used to make programming easier in terms of calculation. I am very thankful I get to experience creating this project which has given me further knowledge of my chosen career without any doubt.