Fahid Shah

Plot, Salt, Smooth (Java) Essay

I wrote three java files called Plotter, Salter, and Smoother to generate, salt, and smooth data for a line graph. The program Plotter.java generates a CSV file containing data points for a line graph based on the equation y = mx + b. Since I could choose any equation, I thought this would be fine and make it easy to analyze the effects of the salting and smoothing. The user of the program can define the range and number of data points, which are then calculated and written to the CSV file. Once the CSV file is generated, it is opened with the default application, in this case, Excel. The Salter program takes the initial data set created by the Plotter.java program and adds a random number to each y value, this value could be positive or negative. The range of the random number to be added is determined by the user. It then reads the CSV file containing the x and y values and writes the modified data, including the new salted y values, to a new CSV file. The names of these files have been hard coded so that it doesn’t complicate the process of running for the program in case they are less experienced when it comes to java. The Smoother.java program reads the salted data set created by the Salter.java program and calculates the smoothed values for the salted y values using a window of user-defined size. It writes the original data and the smoothed values to a new CSV file.The techniques I used to create the programs are as follows. The programs utilize Java's standard libraries for file Input/Output, such as BufferedReader, BufferedWriter, FileReader, and FileWriter, to read and write data from and to CSV files. Error handling, specifically for IOExceptions, is employed to manage any potential issues related to reading or writing files. The Plotter program uses a BufferedWriter to create a CSV file and writes data rows to it based on the user-defined parameters. The data is calculated using y = mx + b, and then the file is opened with the default application using Desktop.getDesktop().open(). The Salter program uses a BufferedReader to read the input CSV file and a BufferedWriter to write the modified data to a new CSV file. It iterates through each line in the input file, parsing the x and y values and adding a random number to the y value. The random number is generated using Math.random(). The Smoother program consists of several methods, including readCsv(), calculateSmoothedYValues(), and writeCsv(). I tried to split this file into four files, one of which was the driver, but it would just break when I did that and after all the time I spent trying to debug it, I sort of just gave up. But, it reads the salted data set and calculates the smoothed values using a sliding window approach. The smoothed values are then written to a new CSV file along with the original data.

I only included one CSV file from each program, but I ran tests where I lowered and raised both the number of data points and different salt values and here is what I found:

I found when the salt range is increased, the noise in the data increases, making the graph appear more scattered. With a higher salt range, the random values added to the original y values have a broader range, resulting in more significant deviations from the original line. Conversely, when the salt range is decreased, the noise level reduces, making the graph appear closer to the original line. This means, varying the salt range alters the amount of noise introduced to the data, and changes appearance of the graph that we can generate in excel. Increasing the number of data points results in a denser graph, as more points are plotted along the x-axis. This can lead to a clearer representation of the trends in the data, especially when the salt values are high. On the other hand, decreasing the number of data points leads to a sparser and less detailed graph, which may not accurately represent the trends in the data. In essence, the number of data points affects the accuracy of the graph.

Finally, in relation to the smoother, when the salt range is high, and the number of data points is low, the Smoother program may struggle to produce a smooth curve that accurately represents the original line. This is because the high noise level and low data density make it difficult to discern the underlying trend. Conversely, when the salt range is low and the number of data points is high, the Smoother program will have an easier time smoothing the data, as the noise level is lower and the data density is higher.