Jaylen Knight  
CSCI-3327-001  
Plotter, Salter, and Smoother report

The java program I have created pseudo mimics how programs like Microsoft Word and Excel Takes the data that is inputted in these programs and encrypts them through salting. A salter’s purpose is to feed the data already present with additional “dummy” data to prevent the file from being properly utilized by other programs. This salted data by itself can’t really be read and needs to be converted back to its original form through a method called smoothing. This generally undoes any salting done to the data to have it brought back to a readable format for an end user.

For my program I have simulated this process by graphing a given function with an amount of data points determined by the user. Then randomizing each data point on the graph by adding a number between -10,000 and 10,000 generated randomly for each data point then plotting said data. Lastly the smoothing part of this process which involves employing a moving average in order to average out all the data points by using the points around each point currently being smoothed. The accuracy of the smoother is heavily dependent on not only the range of the moving average but also the volatility the salter. The following is 3 graphs, 1st being the data from the function y=2\*(x-3)^2+4 with 200 points of x ranging from -100 to 100, the second being the data salted between -10,000 to 10,000 and the third being the data smoothed using a range of 10.

A graph of data and data

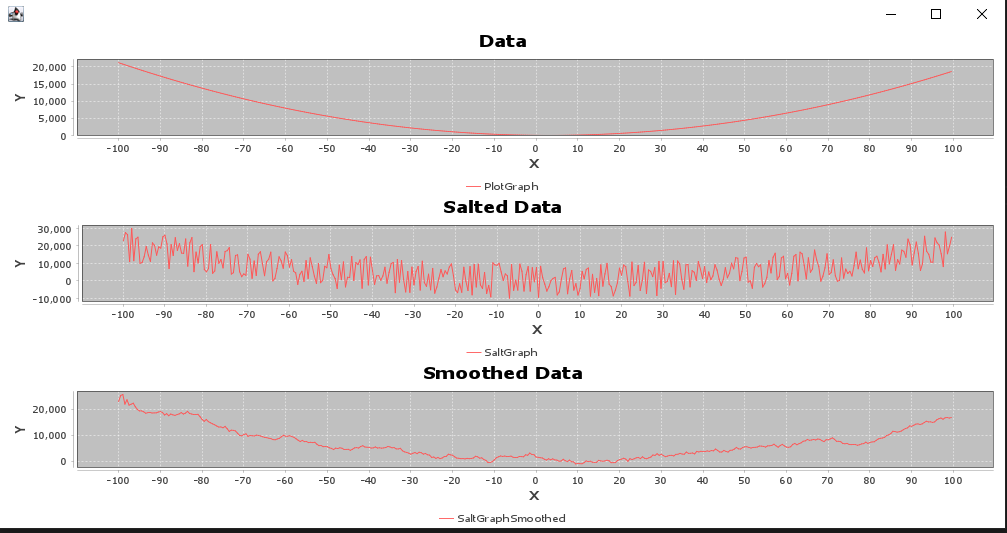
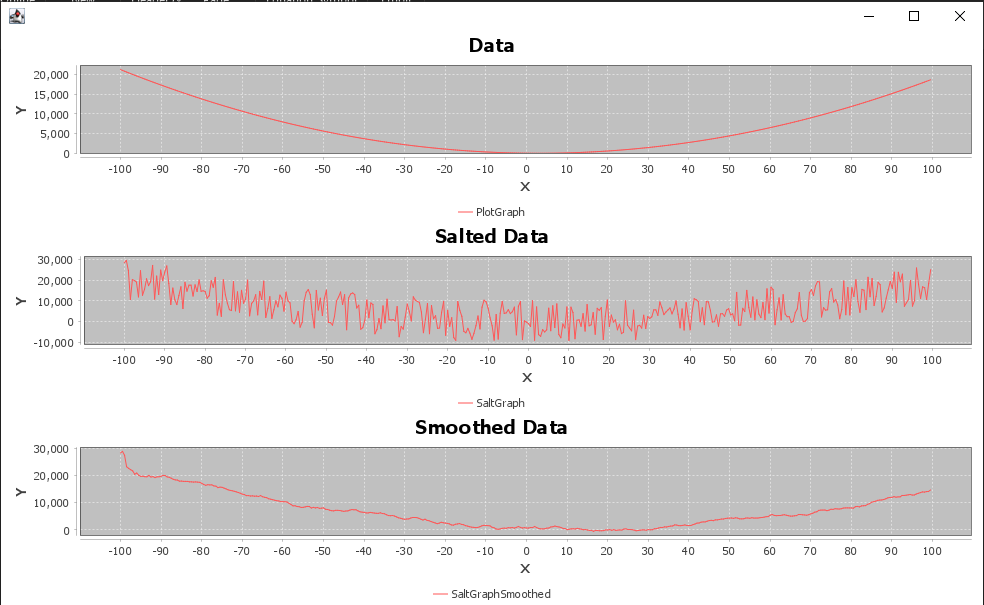
Description automatically generated

The smoothing of the graph isn’t perfect but is significantly smoother than the salted graph and even though utilizing different data points due to the randomness of the salter the following graph was smoothed using a range 20.

A graph of data and data

Description automatically generated

While a more smoothed curve that resembles the original data closer it is still a ways off from being exactly what the original function was. Now I want to compare these graphs which incorporate 200 total points against using the same algorithm however instead of iterating between –100 and 100 by 1 we will now iterate at a rate of 0.5, doubling our total amount of data points to be salted and smoothed. First with a smoothing range of 10 then 20.

As we add more data points of data for the smoother to work with the overall curve of the data gets smoother and smoother however that doesnt necessarily mean more accurate to the original function, especially with the start of each graph since all the smoother has to work with is half the points for your given average range(since it's supposed to take the range amount of values from the left and the right). This causes a dip for the first 5~10% of the given data until the smoother is given more points to work with and those first points being already roughly smoothed helps the algorithm more accurately try and replicate the original data. Some improvements I can think of for the program would be leaving the first 5% or the first half of whatever your moving average range is alone in the slating process to give the smoothing process a better reference of the original data to more closely replicate it.   
 Another method of improvement would be adding an upper and lower limit for the graph at each point. For example, if your smoothed data is outside of a certain percentage range of the original data then it would round to the nearest range. This would be most effective on the start of the smoothed graph and would massively benefit the rest of the smoothing process as mentioned above with giving the smoother a better reference.