Graphing CSV files

 I wrote a program that will allow you to graph anything for the function . To run my program All you need to do is input the range of where it will be between two x values on the graph, and the name of the file you want to save it as.

This file is then saved as a csv to be inputted into an excel file to later be graphed. I have also created a second method that allows you to specify how many points you would like to graph; this is within the parameter as well.   
In which this method has the same functionality as the first, except it gives you more customizable features. (can be seen in figure 1-3)  
  
**Figure 1-1**- this is an example of a large graph with an undefined number of points.

A graph with a line

Description automatically generated

**Figure 1-2**- this is another example with different inputs

A graph with a line

Description automatically generated

**Figure 1-3**- this is an example of a plot with a specific number of points.

A graph with a line

Description automatically generated

As for improving the methods for graphing the points, could implement a way for the user to change the formula as needed within the parameters. Making it even more customizable than before.

I also created a Salter, which allows you to try and hide your data if it is important or sensitive. It does so by asking you for the data you want to salt and asking for a value you want to salt by. Once the program gets the value it will add or subtract a random value from 0 to the salt value to every y in your data thus changing the base. (can be seen between figure 1-1 and 2-1)

**Figure 2-1** – This is the salted version of the graph in Figure 1-1 with a salt value of 76.

**A graph of a graph

Description automatically generated with medium confidence**

**Figure 2-2** - This is the salted version of the graph in Figure 1-1 with a salt value of 27

A graph with blue lines

Description automatically generated

As for trying to improve the salter the best thing I could think about would be to add alternating ranges where within a certain range of numbers the salt value will change. The only issue this will cause is it will be near impossible to decipher the output without knowing the ranges in which the values are being salted. Although this would make it even better for security purposes.

Next I created a smoother, a smoother is meant to accept already salted csv files and turn them into a more readable value. This will be used with the salted to help try and protect your data, the only issue is that there is room for error when it comes to smoothing out salted values. (look at salted graphs in Figures 3-1 and 3-2)

**Figure 3-1 –** this is the smoothed version of Figure 2-1

A graph with blue lines

Description automatically generated

**Figure 3-2 –** this is the smoothed version of Figure 2-2

A graph with a line

Description automatically generated

As you can see the smooth version does not reflect the perfect method of returning the data to its origin. Now this is good, because it gives us room to improve and still helps with security. Even though this smoother isn’t the best at what its meant to do, it can be improved by giving the smoother the input salt value allowing us to get closer to the original values. However, since it isn’t perfect it shows how it can help protect us from those who don’t know what sort of salting, we did.

Now you may be thinking well what happens when I put already smoothed code into the smoother. (check Figure 4-1, 4-2, 4-3)

**Figure 4-1 –** base smoothed file

A graph with a line

Description automatically generated

**Figure 4-2 –** Smoothed 2nd time

A graph with a line

Description automatically generated

**Figure 4-3 –** Smoothed 3rd time

A graph with a line drawn on it

Description automatically generated

As you can see the graph will tend to smooth towards the middle most value of the graph almost creating a straight line. Throughout this cvs coder we find that there may be better methods to incorporate what we are aiming to do as multiple parts of the PSS’s need improvement.