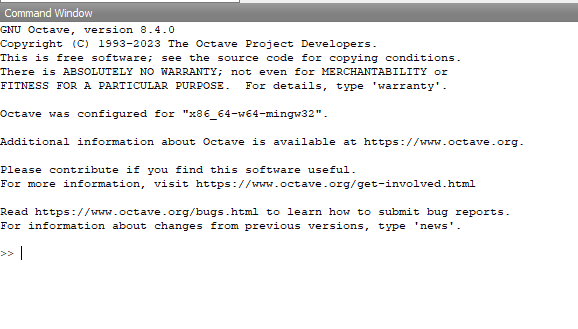
Kenny Vo

PSS2

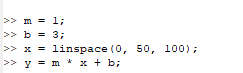
**Octave Set Up:**

1. Navigate to the Octave site: <https://octave.org/>
2. On the menu bar located at the top of the site, select "Download". Then Locate and Download the appropriate installation.
3. Once Installation is complete, open Octave. If done correctly, the command window should display a command window ready to take inputs ( >>):



**Create a Base Graph:**

1. Enter the base graph's equation and variables used into the command window.
   * The code structure facilitates the generation of random points for each variable within specific ranges. When using Octave, you can input values for these variables to align with those predefined ranges. The example below illustrates a possible representation of the code's output. Please note that due to the randomization in the code, exact replication of data is not possible.
     + m can be any integer value from 1-3.
     + b can be any integer value from 1-3.
     + x has a range of 100, lower bound of 0, and upper bound of 50.
     + base plot equation: y = m \* x + b
   * Visual representation on Octave should appear as or similar to:



**Create a Salted Graph**

1. With the Base Graph now established, it's time to introduce the Salt variable and a new variable for the Y-values after the application of salt
   * For the salt component of the code, a strength range of -2 to 2 was designated to ensure the graph remained within specified bounds. Unlike a constant value assigned uniformly to each point, salt strength is distributed individually to every point within the range of -2 to 2. This distribution, implemented in Octave as salt = randi([-2, 2], size(x)), results in varied strengths applied to each point, contributing to the creation of jagged edges in the graph
     + salt can be distributed to any point with an integer value from -2-2.
     + Create a new y variable with salt applied: salt\_y = salt + y;
   * Visual representation on Octave Should appear as or similar to:



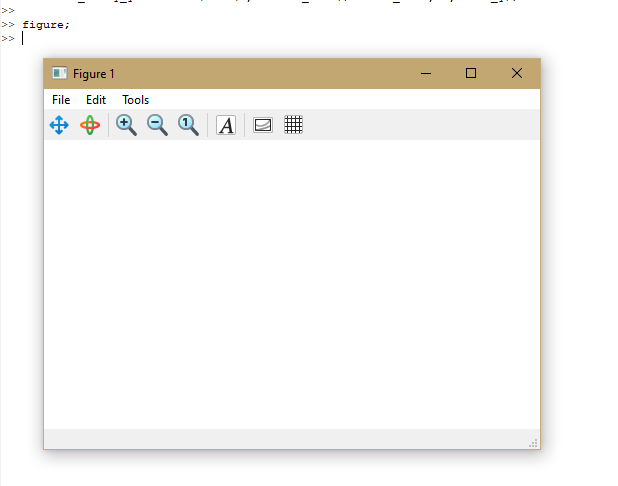
**Create a Smooth Graph**

1. After the Salt Variables have been established, it is not time to introduce the Smooth Variable which requires a window\_size variable and the salted variable from the previous step.
   * The window size determines the number of points grouped together to achieve a more rounded appearance in the salted graph
     + Window size in the code is defined with an integer 9.
     + Create a new y variable with salted y variable and window size.
       - Please note that there are Smooth functions available but it does not come with the base download of Octave, as an alternative you would have to make your own Smoother using built in functions such as filter() and ones().
         1. ones() creates a row vector of ones with a length equal to window\_size.
         2. filter() used for signal processing and applies a filter to a sequence of data
   * Visual representation on Octave should appear as or similar to:

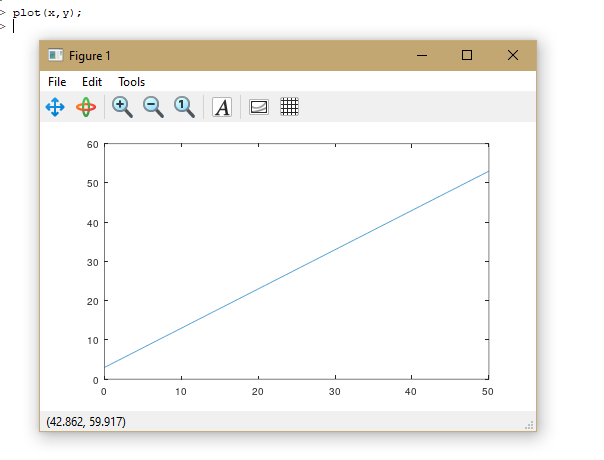


**Plot**

1. Set up Octave to begin plotting the figures by entering **figure;** in the Octave Command Window. It should open a new window that will display the graphs once plotted.
   * Visual representation on Octave Should appear as or similar to:



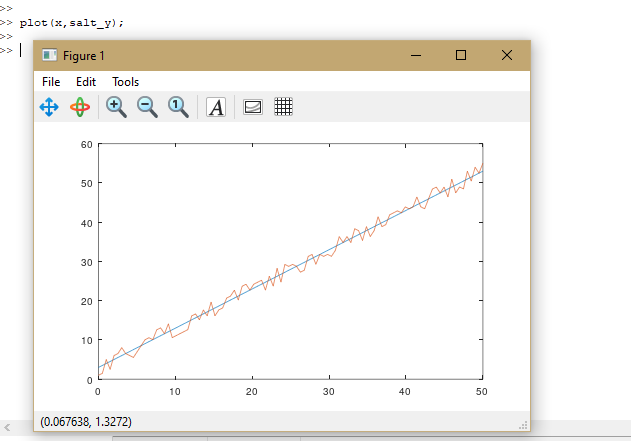
1. Plot the original base graph: y = m \* x + b by entering **plot(x,y);** in the Octave Command Window. In that separate window, the graph should be displayed.
   * Visual representation on Octave Should appear as or similar to:
     + Original graph is represented as blue in the graph.



1. To not discard previous graph and to simply add a new one on top of the previous graph enter **hold on;** in the Octave Command Window
   * Visual representation on Octave Should appear as or similar to:



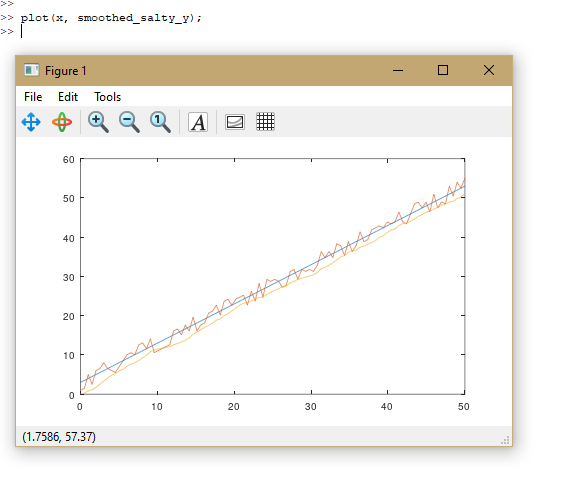
1. Plot the salted graph by entering: **plot(x,salt\_y);** in the Octave Command Window. In that separate window, the salted graph should be displayed along with the base graph.
   * Visual representation on Octave Should appear as or similar to:
     + The original graph is represented as blue in the graph.
     + Salted graph is represented as orange in the graph.



1. To not discard previous graph and to simply add a new one on top of the previous graph enter **hold on;** in the Octave Command Window
   * Visual representation on Octave Should appear as or similar to:



1. Plot the smooth graph by entering**:** **plot(x, smoothed\_salty\_y);** in the Octave Command Window. In that separate window, the salted graph should be displayed along with the base and salted graph.
   * Visual representation on Octave Should appear as or similar to:
     + Original graph is represented as blue in the graph.
     + Salted graph is represented as orange in the graph.
     + Smooth graph is represented as yellow in the graph.

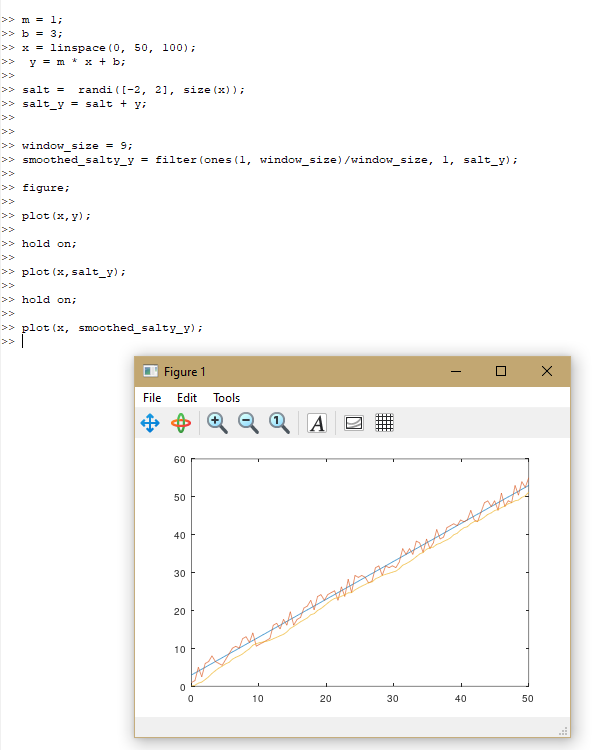


**Completed**

What I learned from this task was first and foremost how to use the basics of Octave and create Graphs, it’s always nice having more options available that provide this kind of services and functionalities. Also, the work arounds for using a base Octave that does not include a built in smoother and having to create one myself. I had to learn how to mimic my code structure and replicate how the data was input there and put it into Octave so it would show the same kind of graphs. I had to familiarize myself with replicating my code structure and adapting the data input methodology into Octave to achieve consistent graph outputs.

Extra

Overall data shown



Texts used:

m = 1;

b = 3;

x = linspace(0, 50, 100);

y = m \* x + b;

salt = randi([-2, 2], size(x));

salt\_y = salt + y;

window\_size = 9;

smoothed\_salty\_y = filter(ones(1, window\_size)/window\_size, 1, salt\_y);

figure;

plot(x,y);

hold on;

plot(x,salt\_y);

hold on;

plot(x, smoothed\_salty\_y);