

Christopher Schretzmann  
Octave Implementation

For this portion of the project, I had to learn how to use Octave or Matlab. The assignment was to recreate the plotter, salter, and smoother through Octave. While I was learning the language, it was not too challenging, just very annoying. The compiler for this language was very buggy as well for me. I tried it on both MacOS and Windows. The nice thing about the language is that the graphing functions are built into the compiler or language. While I could have done everything in the global section, I created functions for each of the methods.

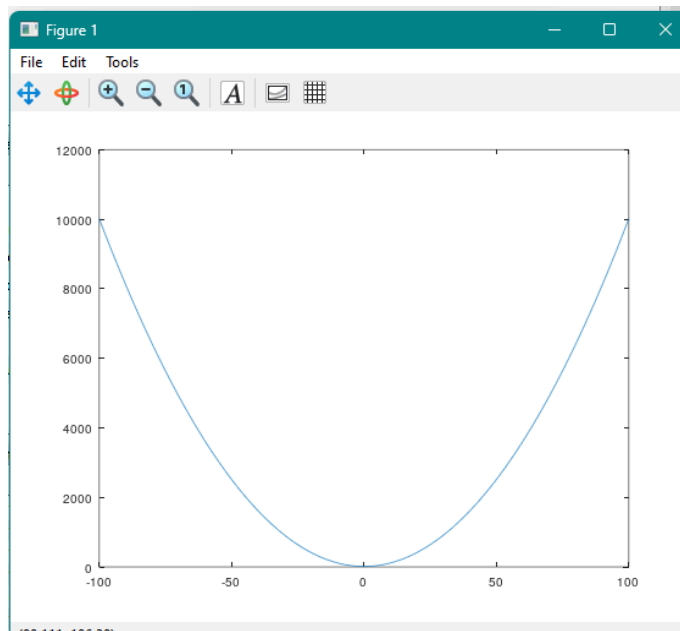
```
lowerBound = -100
upperBound = 100
x = linspace(lowerBound,upperBound);

function y = plotter(x)
    y = x.^2 + 13;
endfunction

yValues = plotter(x);
%plot(x, yValues);
```

For my plotting method, I made the bounds -100 to 100. I used the same exact function as my other programs.

The graph for  $Y = x^2 + 13$

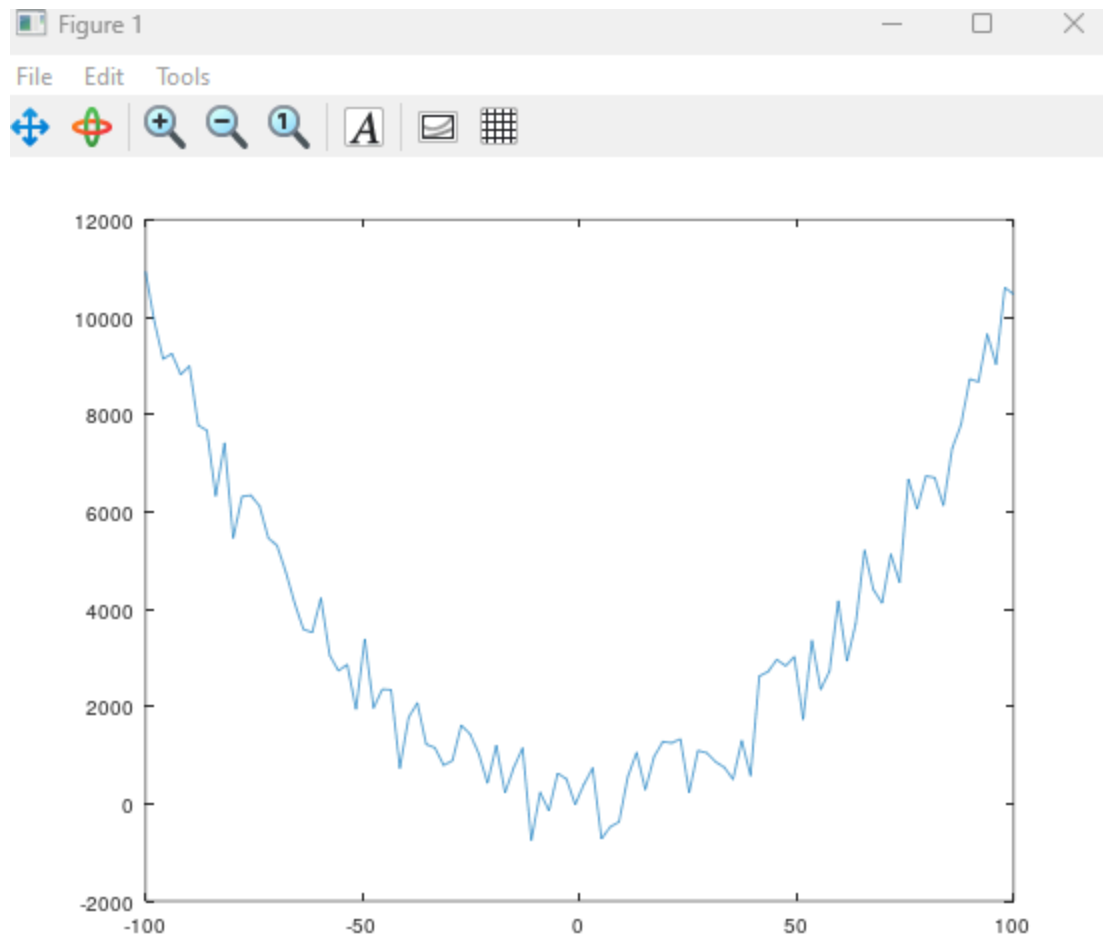


# Salting

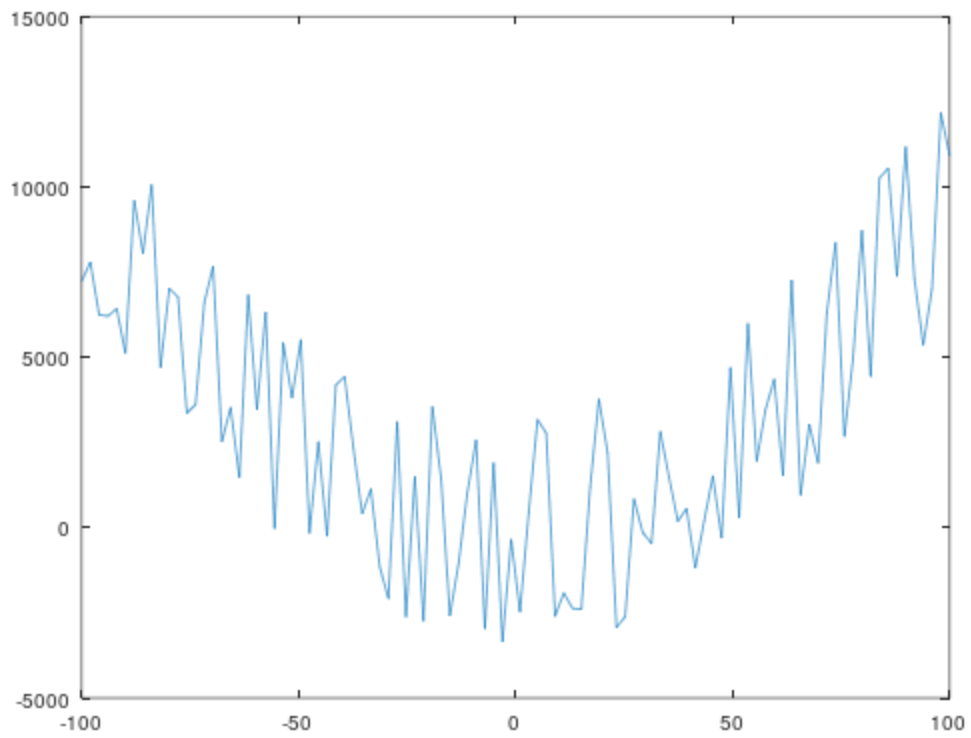
```
randomY = yValues
function y= salty(y)
    for i = 1:length(y)
        y(i) = y(i) + randi([-1000, 1000]);
    end
end

randomY = salty(randomY);
%plot(x, randomY);
```

For my salting function, the language has very convenient random methods in it. Using the documentation, I was able to put random number ranges. The salter works very similar to my Java program. Salting the the values with the value -1000 - 1000 looks like this for  $y = x^2 + 13$



Increasing the values to -100000-100000 makes the graph behave like this



(98.249, -1711.2)

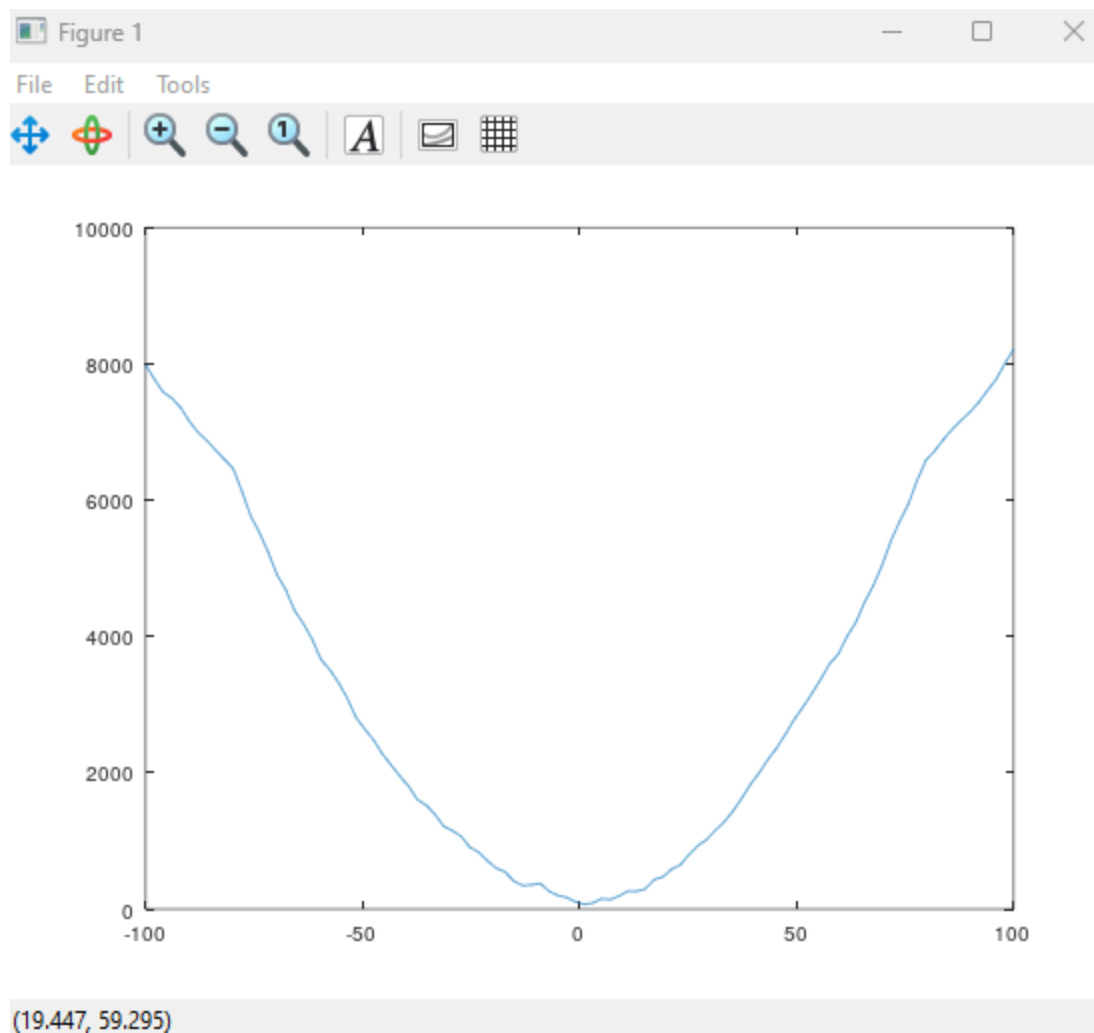
# Smoothing

```
smoothY = randomY;
function smoothY = smoothie(y)
    window = 10;
    for i = 1:length(y)
        sum = 0;
        count = 0;
        for j = max(1, i - window):min(length(y), i + window)
            sum = sum + y(j);
            count = count + 1;
        endfor
        smoothY(i) = sum/count;
    endfor
endfunction

smoothY = smoothie(randomY);
plot(x,smoothY);
```

For the smoothing section, I implemented the same window logic as my other program. The window size above and below the point is how it gets the average. After the average is taken, it assigns it to that point. Learning for loops and the language for Octave was very annoying.

With the window of 10, This is the graphical representation



With a window of 13, we get this graph

## Sources

<https://octave.sourceforge.io/octave/function/min.html>

<https://octave.sourceforge.io/octave/function/max.html>

<https://docs.octave.org/latest/Cell-Arrays.html>

<https://docs.octave.org/latest/The-for-Statement.html>

<https://docs.octave.org/v6.4.0/Random-Number-Generation.html>

<https://docs.octave.org/latest/Defining-Functions.html>

<https://www.youtube.com/playlist?list=PL1A2CSdiySGJ6oZe6XB-TTCFuHc5Fs1PO>

<https://stackoverflow.com/questions/25904643/plotting-function-in-octave>