MATLAB/GNU Octave: Plotting, Salting, and Smoothing

With the MATLAB programming language, you can perform statistics and manipulate data much simpler from a command window. Plotting, Salting, and Smoothing are three things that can be done with almost a simple function call. For these examples, I downloaded and used GNU Octave application and libraries.

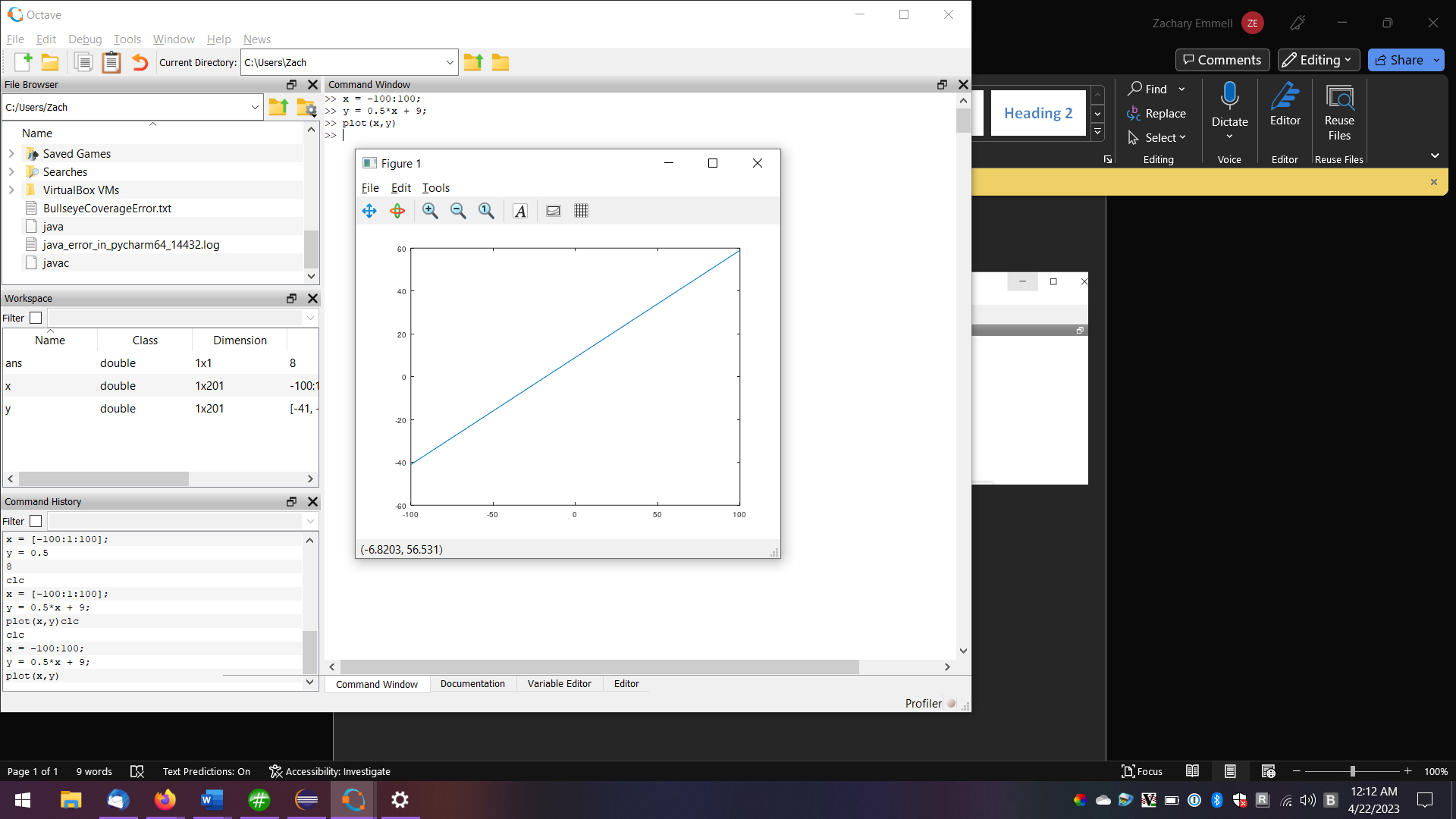
Plotting

To plot a function in GNU Octave, I had to first create your x and y values. These sets or vectors for x and y values and simply be defined in the command window, like so.

x = [0:1:20]; or x = -50:50; Left#-Lower Bound, Middle#-Increment, Right#-Upper Bound

y = sin(x); or y = 2\*x -5; “ ; “ will store the values without running the commands

I then follow up with plot(x,y) and hit enter, you will get….



Salting

In order to salt, I could not just simply add or subtract a random number from y, because MATLAB is not able to do this. I needed to create a vector of random values between 0 and 1, then scalar multiply this vector by the salt range I wanted. This vector could then be added to the y-values and plotted. It looks like this…

z = 100; (salt range)

x = [-100:2:100]; (x-values)

y = 0.5\*x+9; (y-values)

salt = y + z\*rand(size(x)); (salt values are a vector of random numbers of the size of x, multiplied by z, then added to y)

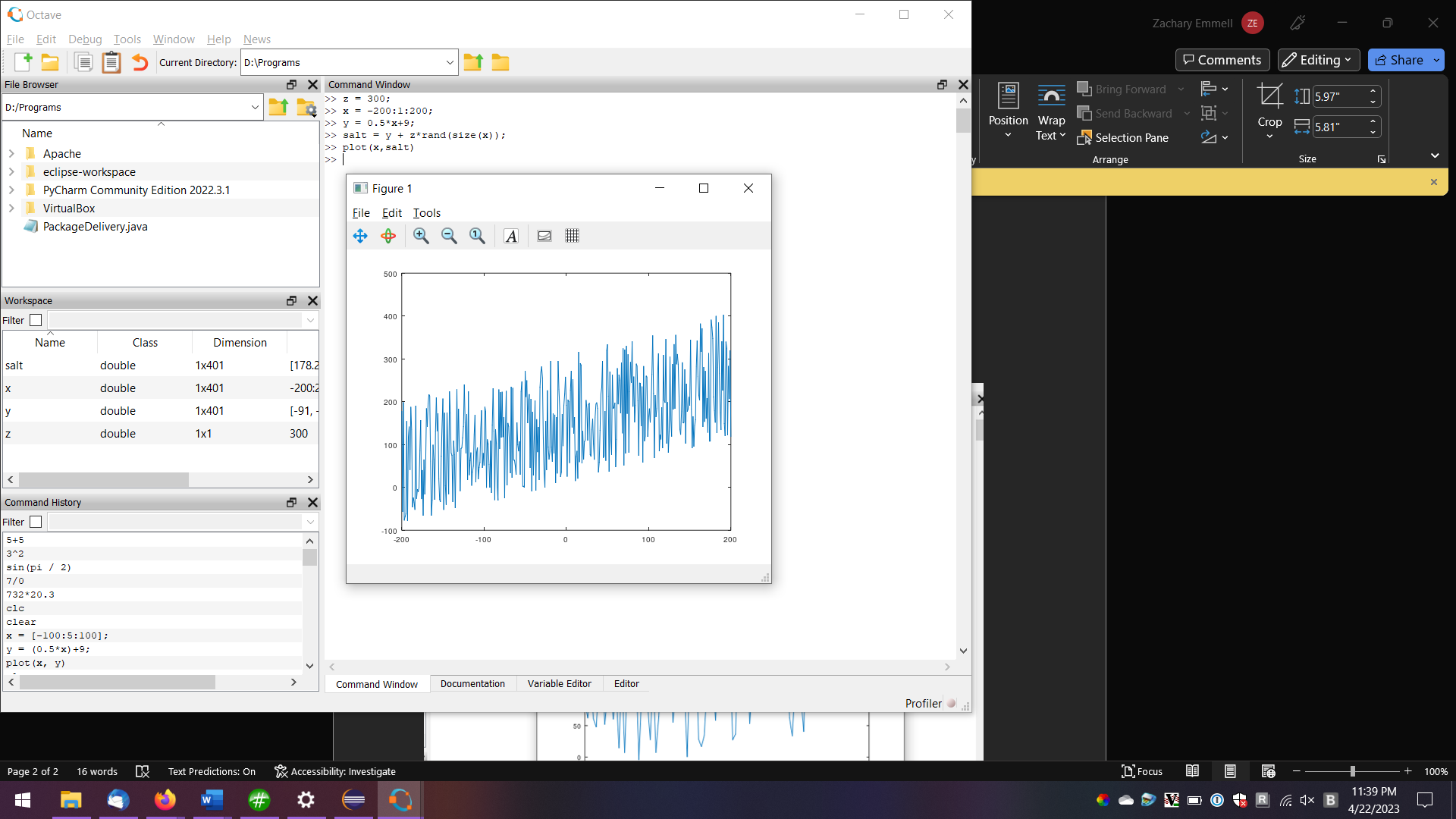
plot(x, salt)

Graphical user interface, application

Description automatically generated

X = -200:200

Z = 300



Smoothing

MATLAB usually has a smoothing function you can call like “B = smoothdata(A)”. This method does not carry over to GNUOctave. Instead you must use the regdatasmooth() function. Using this function, I could perfectly smooth a salted vector like so…

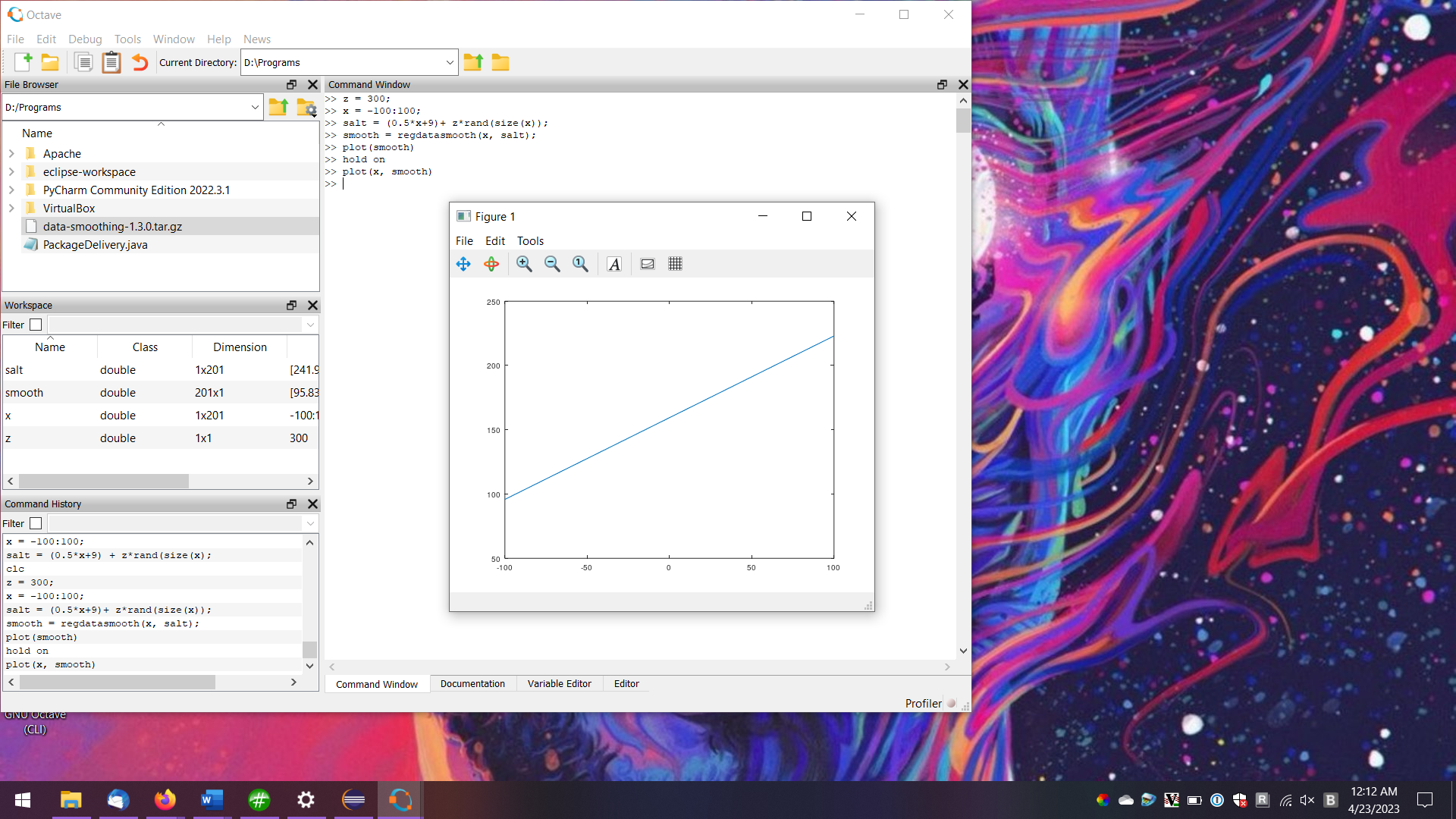
Z = 300;

x = -100:100;

Salt = 0.5\*x+9+z\*rand(size(x));

Smooth = regdatasmooth(x, salt);

Plot(smooth)



Tutorials: <https://www.tutorialspoint.com/matlab/matlab_vectors.htm>

<https://octave.sourceforge.io/data-smoothing/function/regdatasmooth.html>