**Octave Mini-Report**

By: Reece Cohen

For: CSCI 3327, Probability and Statistics

Date: December 11, 2024

**Section 1: Setup**

To get started with my research I had to decide between using MATLAB or Octave for all of my plotting and graphing. After a quick look at both options I went with Octave, mainly because it was free. The installation was simple and just used homebrew, a package installer for Mac operations systems. After, all that was left was to just put in the terminal “brew install octave” and then “octave –gui” to run it.

**Section 2: Learning Octave**

Octave is a language based around assisting in numerical situations. Most of my learned knowledge comes from Mr. STEM EDU TV, a channel on YouTube with just over 50 tutorials showcasing various ways you can use Octave.

**Section 3: Graphing**

Graphing gave me some trouble initially due to how I was trying to figure out how to get the y axis to equal x^2/10 but I kept getting the error

“*error: for x^y, only square matrix arguments are permitted and one argument must be scalar. Use .^ for elementwise power*.”

What I did not realize about this error was that it was trying to get me to use “.^” instead of just “^” for my squaring of x. After that problem was fixed the graphing was simple and here is the result



*Graph for the function (EX2.1)*

A screenshot of a computer

Description automatically generated

*Code for the graph of EX2.1 (EX2.2)*

**Section 4: Salting**

Figuring out how to salt the graph was both complicated and simple at the same time. The problem with the methods that octave provides is that it is not clear as to what parameters do and I feel they are not well documented enough. After messing with the methods to learn how they function the code itself to salt the graph was simple.

*>> x = -100:100;*

*>> y = x.^2/10*

*>> sd = std(y)*

*>> salt = sd \* randn(size(y))*

*>> salty = y + salt*

*Code used to compile the graph*

**

*Salted graph of EX 2.1 (2.3)*

**Section 5: Smoothing**

I had a similar situation with smoothing like I had with salting. The method for filter had a clearer explanation and the execution for smoothing was simple as well. For the filter method I had decided to use a similar window to the one I used in my own programmed plotter which was 3 units to the right and 3 units to the left of the current point.

*>> x = -100:100;*

*>> y = x.^2/10*

*>> sd = std(y)*

*>> salt = sd \* randn(size(y))*

*>> salty = y + salt*

window = 7

>> smoothy = filter(ones(1, window) / window, 1, salty);

>> smoothy = filter(ones(1, window) / window, 1, smoothy);

*Code to graph a smoothed salted graph*

The last line of code is repeated however many times you would want the graph to be smoothed, for my situation I had smoothed the graph about 5 times before it started to get flattened out. The graph itself is not as smoothed as I would like it to be.

****

*Smoothed graph of EX2.3 after 5 iterations (2.4)*

The graph still holds the general shape of EX2.1 but it’s still too rough for my liking. The problem here is that smoothing the graph out too many starts to flatten it out and the graph loses its original shape.

**Section 6: Conclusion**

I am not favorable to using this method of graphing, I enjoy how quick you can throw together a graph, but you do not have as much control as much as I would like to have, it may be that I do not know the language well enough but I am more comfortable using languages I already know and making the graphs in excel over this.

*Sources*

[Homebrew installation website](https://brew.sh/)

[Install Octave for Mac OS](https://wiki.octave.org/Octave_for_macOS)

[Mr. STEM EDU TV Octave tutorials](https://www.youtube.com/watch?v=GjvxquBlPYA&list=PLuR45MKVZJHiQDOdSzGcl7_S0jSOHn8o_)

Sites used to learn specific methods

* [Plotting and graphing (octave.org)](https://docs.octave.org/v4.0.0/Two_002dDimensional-Plots.html)
* [Standard deviation](https://octave.sourceforge.io/octave/function/std.html) (sourceforge.io)
* [Random number generator](https://docs.octave.org/v6.4.0/Random-Number-Generation.html) (octave.org)
* [Filter](https://octave.sourceforge.io/communications/function/filter.html) (sourceforge.io)