Dustin McDonnell  
Octave Tutorial

Installed GNU Octave

Followed this tutorial <https://www.youtube.com/watch?v=bmE6SWE6c_A&list=PL1A2CSdiySGJ6oZe6XB-TTCFuHc5Fs1PO&index=2>

***Tutorial***

Learned basic arithmetic commands.

A math equations and numbers

Description automatically generated with medium confidence

Learned how to set and call Variables.

A white background with black text

Description automatically generated

Using a semi-colon at the end of setting a variable allows you to set the value in the background.



Initialized row and column vectors.

A white background with black text

Description automatically generated

Initialized Matrices

A white background with black numbers and letters

Description automatically generated

Basic Arithmetic can also be applied to matrices.

A screenshot of a computer

Description automatically generated

Octave has additional commands for special Matrices.

ones(x,y)

A white background with black text

Description automatically generated

zeros(x,y)

A number and numbers on a white background

Description automatically generated with medium confidence

Identity Matrix eye(x)

A screenshot of a computer

Description automatically generated

Generates n x m matrix of uniformly random elements: rand(x,y)

A number and numbers on a white background

Description automatically generated

Generates a n x m matrix of normally distributed random elements: randn(x,y)

A black text on a white background

Description automatically generated

Vectors can also be defined in a range using colons.

A number on a white background

Description automatically generated

Additionally, you can define how much a number increases with colons.

A number on a white background

Description automatically generated  
Accessing the contents of a matrix is done with parentheses.

A math equations on a white background

Description automatically generated

Octave uses a working directory like a terminal with similar commands for navigation.

Pwd for current pathway.



Cd to change directory, in this case I am navigating to the directory where the part 1 csvs are saved.

A close up of a text

Description automatically generated

Octave has built in load and save functions.

I thought you could potentially use this to load csvs, but this is not the case.



Instead, you would save the csv file into a matrix with a name of your choosing.



This also automatically prints the entire output which I will not be copying here as the CSV is 10000 lines long and my Octave IDE won’t let me scroll that far up.

This is how you would save data in Octave; I show this by creating a test matrix V and saving it to the same folder as the csvs.

A screenshot of a computer

Description automatically generated

You can also merge vectors like so.

A white background with black text

Description automatically generated

A math equations on a white background

Description automatically generated

There are functions for determining the size of a matrix.

A black text on a white background

Description automatically generated

In the example above I have the Cube Root csv saved in the variable m and octave returns that it is

20002 rows long and 2 columns wide.

You can also use length to determine the length of the longest side of the matrix.



This is when I discovered the need to print to terminal for testing purposes which is accomplished this way.



This was my first attempt to graph one of my cube root csvs.



A white rectangular object with a blue dot

Description automatically generated

It’s obvious that this is not the correct output. But why?

This introduced me to the “hold on” command, without it Octave automatically overwrites old plotted data, despite it being contained in a for loop.

Here is the follow up attempt.

A black and white text

Description automatically generated

A line graph with numbers

Description automatically generated

Which is exactly what we want.

We can then save it to the current directory with the following.



Boolean Operators in Octave are similar to other programming languages.

== Equal to

!=, ~= Not Euqal to

> Greater Than

< Less than

>= Greater than or equal to

<= Less than or equal to

If statements are like the For loop I used earlier. The language seems similar to Python in terms of syntax but I’m not super familiar with Python.

A white background with black text

Description automatically generated

Else statements work in a similar way.

A white background with black text

Description automatically generated

Here is an example while loop.

A math problem with numbers

Description automatically generated with medium confidence

From what the tutorial said each function in Octave is written and saved to it’s own individual .m file so you do not have multiple functions to one file.

***Part 2***

To begin I created a smaller cube root graph using my part 1 programs with the range -200 to 200 as Octave seems to struggle with larger graphs. Then I used variable m to contain the data.



plotGraph(input1)

A computer screen shot of a person

Description automatically generated

I wrote the plotGraph function because I got tired of writing the longform of the built-in plot function to fit the csv format. The output is below, you will notice it’s the same as the graph used in the tutorial section.

A line graph with numbers

Description automatically generated

saltGraph(input1, input2)

A screenshot of a computer program

Description automatically generated

Octave seems to only allow a random integer between 0 and 1 so to get around this I just multiplied whatever double rand(1) calls by input 2, which is the range the random number is supposed to be selectable from. This should be a close approximation of choosing a random number from 0 to input2.

Then I simulated a coin flip to determine whether the random number is added or subtracted from the initial value. The graph below shows the output.





A white background with many colored dots

Description automatically generated

smoothGraph(input1,input2)

A screenshot of a computer program

Description automatically generated

smoothGraph functions here in much the same way as it does in Part 1 of my project but with significantly less code about formatting because Octave makes it easier to pull information from a csv. The nested for loops check for out of bounds values and ignores them while averaging data points in a range of input2.





A graph showing a number of dots

Description automatically generated

It’s clearly smoother, but not as smooth as the Part 1 smoothed graph, most likely because it doesn’t have 20000 data points to average from.

<https://docs.octave.org/v4.2.0/One_002ddimensional-Interpolation.html>

This was also potentially an option but I found the documentation confusing, so I decided to write my own function.