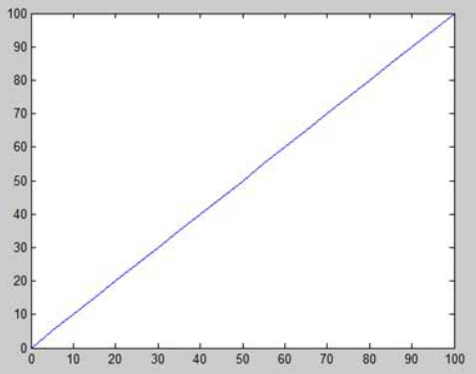
Colin Glory

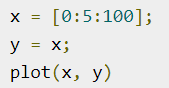
Mar 23, 2023

Probability and Applied Statistics

Professor Hoy

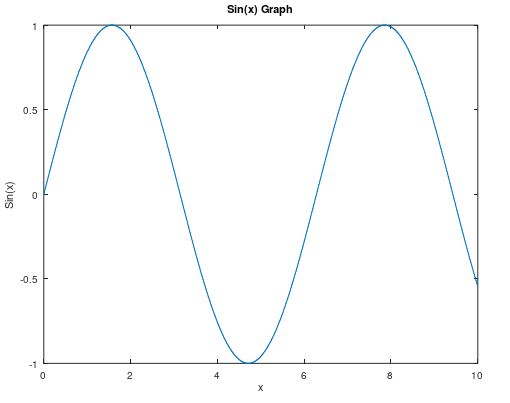
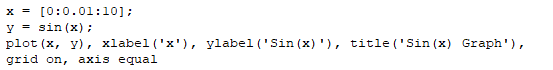
Matlab/Octave Report

What is MatLab? According to *Tutorials Point,* “MATLAB is a programming language developed by MathWorks. It started out as a matrix programming language where linear algebra programming was simple. It can be run both under interactive sessions and as a batch job.” The first point of Matlab that will be covered is plotting. Plotting in Matlab is a much easier task then programming functions and methods to plot graphs in java and python. It is as simple as specifying an x range of values, as well as a y range of values, then plotting them. The syntax for a basic graph is shown below:



→

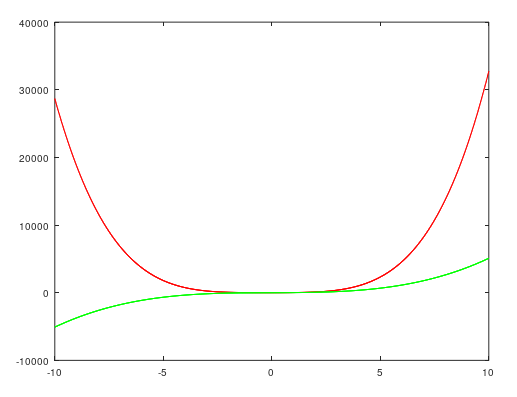
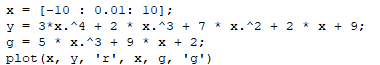
In the code above, there is an array assigned to the variable x. The first value is 0, which initiates the graph’s starting point for x values. The 0 is followed by a 5, which sets the increment value for each x point (0,5,10,15….). The final value that is included is 100, which is associated with the ending value for x values. Setting the y values is essentially assigning a function/formula for the x values to iterate through. In this instance, y = x, which will generate a straight line.

Diving a bit deeper into plotting, MATLAB will allow users to add axis titles, a graph title, and even grid lines. This allows for the graphs to look more professional. Here is a sample of sprucing up a graph to look more professional:

⇒

The code above is self-explanatory, due to it being easily identifiable for what each part stands for.

Matlab lets users get even crazier by having the ability to graph multiple graphs on the same plot. All a user needs to do is assign another variable to a function, then add it to the plot statement. A sample plot statement could be, “plot(x,y,x,g,”-”).” There can be even more customization for graphs, in which a user can adjust the color of their plotted graphs. In Matlab, there are eight basic colors: white, black, blue, red, cyan, green, magenta, and yellow. Here is a sample plot with multiple functions, as well as color changes:

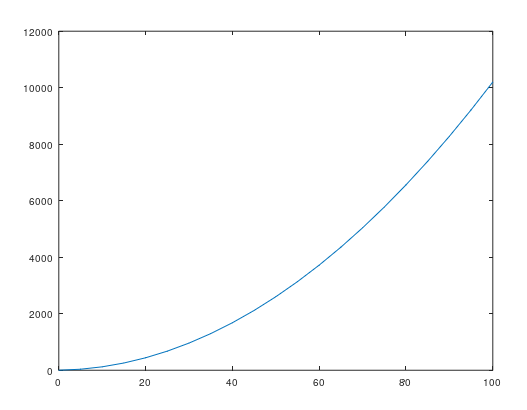


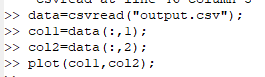
⇒

In Matlab, a user can load in a csv file that is

in their directory. Let’s start out with a normal csv.

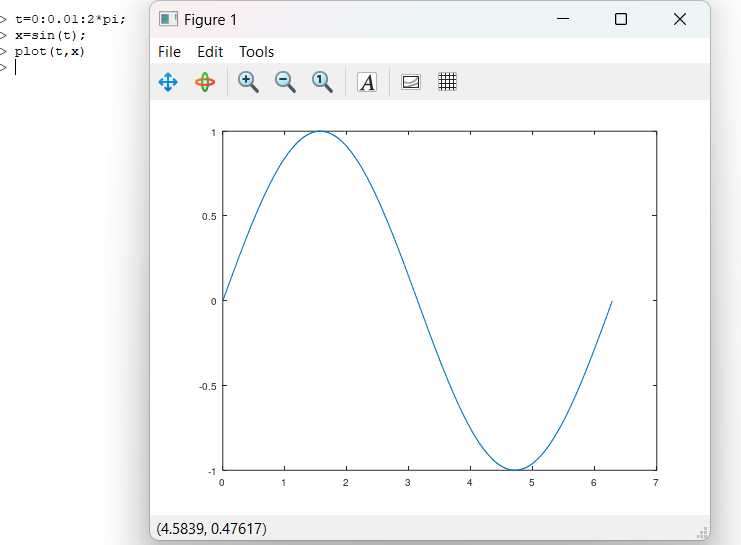
Loading in a csv is straightforward. There is a

command called *csvread.* A variable can be assigned to the command *csvread(“csv name”)*. If a user plots the variable assigned to the command, the graph will be incorrect. The columns of the csv must be assigned to an “x” and a “y.” To access the columns and assign them, use the following command: *col1=(:,column number)*. Here is an example of loading a csv file: 

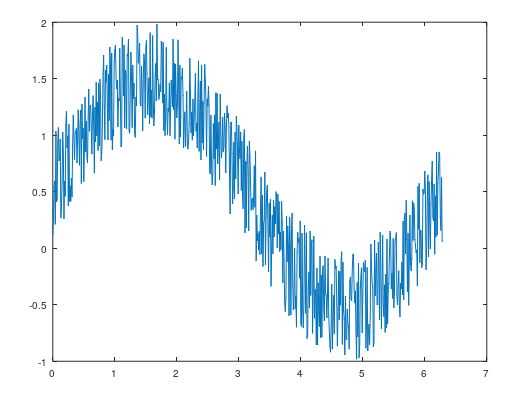


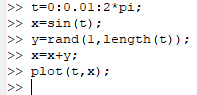
⇒

Let’s move on to salted and smooth data. A user can either load csv files of salted/smooth data or can manipulate Matlab/Octave itself. Let’s first start with salting data. I am going to salt the function of sin(t). Here is the code and graph of sin:

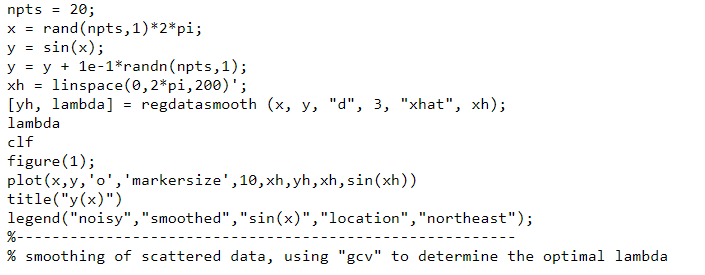


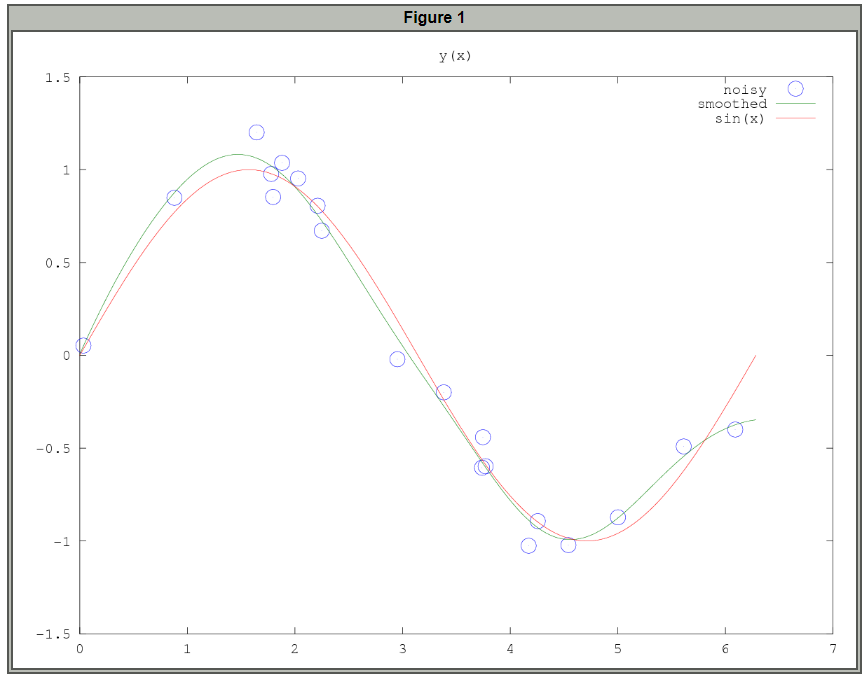
Now, let’s salt this chart. To do so you can follow these commands:



Now you can see what some salted data looks like. Let’s move onto salting data. 

⇒

If you are using *Octave*, then a user must download the *data-smoothing* package from their website. Once it is fully loaded, then we are ready to go. As seen on the left, there is the *regdatasmooth* function, which smoothes the y values. You will find below what the output will look like. As you can see, there is the normal line function, noisy data points, as well as the smoothed data line itself.



**Sources**

<https://www.tutorialspoint.com/matlab/index.htm>

<https://gnu-octave.github.io/packages/data-smoothing/>

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

https://www.youtube.com/watch?v=SP3zTandUjA