Instructions

For each question below, create a .m file (a script file) with the title "YourNameQuestion#". Write all of the code you need to answer the question in this script file. Your code should run from start to finish and produce the answers to the questions. For any parts of the question that require you to respond, please write your responses in the script and comment them out by placing a % symbol before the answer. This will allow me to read your answers while allowing Matlab to skip over those lines of code when executing. You should make every attempt to keep your code as concise as possible. This means that if there is an opportunity to make an operation generalizable (i.e. make it work for more than one set of conditions), you should do so. These questions are intended to get you thinking about the many different real world ways you might need to use the things you have learned about MATLAB so far, and to get you thinking creatively about how to solve problems with programming. If you need help on any of these questions, please work with classmates who have more experience. You will not be graded on these questions, but it is important that you understand the concepts, so please, do not just copy your classmates answers, make an attempt to understand the logic behind each piece of code.

1 Practice Problems

1.1 Fibonacci

A. Using a For Loop (and conditional statements if necessary), write a script that generates the first 10 numbers of the Fibonacci sequence (0,1,2,3,5,8,13,21,34).

B. Repeat this, generating the first 100 numbers of the Fibonacci sequence (plot these numbers to see how quickly this sequence grows).

1.2 Plotting in matlab.

A. Create a matlab figure window using figure. Plot vertical and horizontal axis lines, extending from -2 to +2 (plot the first one using the plot function, then execute hold on, then plot the second one). Plot these as black lines.

B. Now generate a random 2-dimensional vector \( v \) using randn, and plot it as a red line coming from the origin. Execute axis equal to tell matlab to make the units of the two axes the same, and then hold off to tell matlab you’re done adding things to this figure. Make sure the vector looks like what you expect! Re-execute your code several times (generating a new random vector each time), and verify that it’s working.
C. Create a vector containing 31 equi-spaced angles from 0 to 2\(\pi\) (hint: create a vector of integers from 0 to 30, and then re-scale it). Now, replot your vertical and horizontal axis as you did in section A, and use this new vector of angles to plot a unit circle in green, on top of vertical and horizontal axes. To do this, you will need to convert each angle in this vector into a x value and a y value to feed into the plotting function. (If you can’t remember how to do this, think back to your high school trigonometry). What happens if you forget to execute axis equal?

D. Plot a random 2-vector on top of this (in red as above), and verify visually and numerically whether its length is greater or less than 1. Do this six times and plot the output of each random vector draw as a separate subplot in a single figure (choose the arrangement of subplots that allows you to view the data as well as possible on your screen).

1.3 Polynomial regression.

Load the file `regress1.mat` into your MATLAB environment. Use the scatter function to plot variable Y as a function of X.

In many cases when analyzing data, we will want to find the best fitting curve for a set of data that explains the relationship between the independent variable \(X\) and the dependent variable \(Y\). Linear regression allows us to do this with some assumptions and constraints on the relationship between the two variables. The type of regression we will explore in this exercise is polynomial regression. Polynomial regression is a form of linear regression in which the relationship between the independent variable \(x\) and the dependent variable \(y\) is modeled as an \(N\)th degree polynomial.

As a reminder, an \(N\)th degree polynomial is defined by the following equation:

\[
p(x) = p_1X^n + p_2X^{n-1} + \ldots + p_nX + p_{n+1}
\]

In the following exercise, we will fit the coefficients (the p’s in the equation above) of polynomials of different degrees \(N\) to the data in `regress1.mat` (the vector \(x\) is the independent variable, and the vector \(y\) is the dependent variable).

To start, explore the matlab functions "polyfit" and "polyval" by typing "help polyfit" and "help polyval" into your matlab window. You will utilize these functions in the following exercise.

Read the instructions below carefully before completing this question. You will be asked to repeat the instructions below for several values of \(N\) (\(N=0,1,2,3,4,5\)).

Find a least-squares fit of the data with polynomials of order \(N\) using polyfit. Once you have the coefficient values that polyfit returns, evaluate a polynomial of degree \(N\) using polyval. Plot values of vector \(x\) versus the polynomial you just evaluated as a green curve.

Do this for each value of \(N\) (\(N=0,1,2,3,4,5\)), and plot each polynomial (as a curve) in its own subplot, plotted overtop of a scatter plot of the original data from `regress1`. Which fit do you think is "best"? Explain why you think this. Remember to include titles for each subplot (describing the degree of the polynomial plotted below).
1.4 The Psychometric Function.

I’ve written a simulation program \( C = \text{simpsych}(\mu, \sigma, I, T) \) that simulates a psychophysical experiment used to measure a psychometric function. \( I \) is a vector of intensities and \( T \) is a vector of the same length whose corresponding entry is the number of trials at the corresponding intensity. Assume that the observer’s task is 2 Alternative Forced Choice. The observer sees two stimulus arrays on every trial and must say which one contains the target they have been told to identify. One and only one contains the target. The observers’ probability of being correct on a trial is

\[
P = \frac{1}{2} + \frac{1}{2} \phi(I, \mu, \sigma^2) \tag{2}
\]

where \( \phi(x, \mu, \sigma^2) \) is the cumulative distribution function of the Gaussian distribution with mean \( \mu \) and variance \( \sigma^2 \) evaluated at \( x \). This curve (Eq. 2) is the psychometric function.

Take a look at the construction of the function \text{simpsych}. I’ve constructed the function as a lambda (or anonymous) function as a demonstration. Lambda functions are a useful tool for building functions within a script (they behave like normal functions, but they can be defined within a larger script and do not have to be saved as a separate file). I’ve inserted comments to help explain the parts of this particular lambda function.

A. For the first experiment, illustrate the use of \text{simpsych} with \( T = \text{ones}(1,20)*10000 \) (which creates a vector of length 20 filled with 10000’s), and \( I = 1:20 \) for \( \mu = 10 \) and \( \sigma = 1 \). Plot \( C ./ T \) versus \( I \).

B. For the second and third experiments, use the values of \( \mu = 5 \) and \( \sigma = 1 \), and then \( \mu = 10 \) and \( \sigma = 5 \) respectively. Plot the same curve for these experiments as you did in the experiment above. Describe the difference between the three experiments. If you increase \( \mu \) how does the curve change? If you increase \( \sigma \) how does the curve change? If you are not sure, make more plots with different values of mean \( \mu \) and standard deviation \( \sigma \). What is the range of \( p \)?

C. Now, repeat the experiments with \( T = \text{ones}(1,20)*100 \) and plot the results (including the psychometric function). Contrast the plots from the first set of experiments with the results from the second set of experiments. Comment on the difference.