Use Python and Matlab for all questions below (unless some derivation is required which you can do better by hand). If using Python, use Jupyter notebook to format properly, comment when necessary, and include all required and necessary outputs (Jupyter allows LATEX embedding which would be great if you could use it). Convert the notebook to pdf and upload to blackboard against the assignment. If using MATLAB, use MATLAB publish to convert to pdf. Solution documents should be well commented.

Q1. In this question, you need to create your own synthetic data and do curve fitting on it. First create an input vector x as [-20, -19,….19, 20].

(a) Create an output vector $y = a_0 + a_1 x + n$ where you choose a_0, a_1 as some constants such that $a_0 \neq 0$ and $0.5 \leq a_1 \leq 2.0$. n represents some random noise added to the data. At any data point x_i, you can add some random noise n_i where n_i is a random number between -2 and 2. Here is how you can create a random number between 0 and +1:
https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.random.rand.html#numpy.random.rand
To create a random number between -2 and +2, you can use 4*(np.random.rand()-0.5) [Think about why this creates a random number between -2 and +2]. Make sure that the random numbers n_i are all different at different i. Now plot x, y making sure that they are plotted as a scatter (not connected through line segments). Do linear regression on this data to create a model $y = \bar{a}_0 + \bar{a}_1 x$ where $\bar{a}_0, \bar{a}_1$ are the regression estimates of a_0, a_1. How do $\bar{a}_0, \bar{a}_1$ compare with a_0, a_1? Plot the line $y = \bar{a}_0 + \bar{a}_1 x$ on the same plot as the data showing the curve fit model in comparison with the original data.

(b) Create an output vector $y = a_0 + a_1 x + a_2 x^2 + n$ where you choose a_0, a_1, a_2 as some constants such that $a_0 \neq 0$ and $0.5 \leq a_1 \leq 2.0$ and $0.1 \leq a_2 \leq 0.5$. At any data point x_i, you can add some random noise n_i where n_i is a random number between -10 and 10. Plot x, y making sure that they are plotted as a scatter (not connected through line segments). Do quadratic regression on this data to create a model $y = \bar{a}_0 + \bar{a}_1 x + \bar{a}_2 x^2$ where $\bar{a}_0, \bar{a}_1, \bar{a}_2$ are the regression estimates of a_0, a_1, a_2. How do $\bar{a}_0, \bar{a}_1, \bar{a}_2$ compare with a_0, a_1, a_2? Plot the curve $y = \bar{a}_0 + \bar{a}_1 x + \bar{a}_2 x^2$ on the same plot as the data showing the curve fit model in comparison with the original data.

(c) Load data.mat. In Python you can use loadmat function to load the data file:
https://scipy-cookbook.readthedocs.io/items/Reading_mat_files.html
In Matlab, you can simply do load('data.mat'). There are two variables in data.mat: x,y. Extract these variables as shown in the above link and fit a polynomial of appropriate degree to this data $y = f(x)$. What degree works best for a good fit? Plot the fitted curve on the same plot as the data.

Q2. Download the data from the stressstrain.csv data file on Blackboard. .csv is an excel format which can be imported in Matlab and Python

In MATLAB you can read the data using:
T = readtable('stressstrain.csv');

In Python you can use:

from numpy import genfromtxt

T = genfromtxt('stressstrain.csv', delimiter=',')

The data comes from an actual tensile test experiment conducted in MMAE419 on stainless steel 316 bar. The data is in the nonlinear regime. The first column is plastic strain values (e) and the second column is corresponding true stress (s) values. In the nonlinear regime, the relation between e and s is:

$$s = Ke^{\eta}$$

Where K, η are material constants which need to be determined using curve fitting. See this for more info:

http://audi.nchu.edu.tw/~wenjea/mechanical103/Chapter_5.pdf

(a) Plot $e\ vs.\ s$ on a scatter plot
(b) Find the constants K, η using curve fit.

Q3. If you have only two data points ($x1$,$y1$) and ($x2$,$y2$), show that the least square regression line which is the best fit for these two data points is the same as the line which directly connects these two points.